ADVANCES IN INTELLIGENT AND SOFT COMPUTING 130

Kusum Deep
Atulya Nagar
Millie Pant
Jagdish Chand Bansal (Eds.)

Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20–22, 2011

Springer

Advances in Intelligent and Soft Computing 130

Editor-in-Chief: J. Kacprzyk

Advances in Intelligent and Soft Computing

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 116. Yanwen Wu (Ed.)
Advanced Technology in Teaching - Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009), 2012
ISBN 978-3-642-11275-1

Vol. 117. Yanwen Wu (Ed.)
Advanced Technology in Teaching - Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009), 2012
ISBN 978-3-642-25436-9

Vol. 118. A. Kapczynski, E. Tkacz, and M. Rostanski (Eds.)
Internet - Technical Developments and Applications 2, 2011
ISBN 978-3-642-25354-6

Vol. 119. Tianbiao Zhang (Ed.)
Future Computer, Communication, Control and Automation, 2011
ISBN 978-3-642-25537-3

Vol. 120. Nicolas Loménie, Daniel Racoceanu, and Alexandre Gouaillard (Eds.)
Advances in Bio-Imaging: From Physics to Signal Understanding Issues, 2011
ISBN 978-3-642-25546-5

Vol. 121. Tomasz Traczyk and Mariusz Kaleta (Eds.)
Modeling Multi-commodity Trade: Information Exchange Methods, 2011
ISBN 978-3-642-25648-6

Vol. 122. Yinglin Wang and Tianrui Li (Eds.)
Foundations of Intelligent Systems, 2011
ISBN 978-3-642-25663-9

Vol. 123. Yinglin Wang and Tianrui Li (Eds.)
Knowledge Engineering and Management, 2011
ISBN 978-3-642-25660-8

Vol. 124. Yinglin Wang and Tianrui Li (Eds.)
Practical Applications of Intelligent Systems, 2011
ISBN 978-3-642-25657-8

Vol. 125. Tianbiao Zhang (Ed.)
Mechanical Engineering and Technology, 2011
ISBN 978-3-642-27328-5

Vol. 126. Khine Soe Thaung (Ed.)
Advanced Information Technology in Education, 2011
ISBN 978-3-642-25907-4

Vol. 127. Tianbiao Zhang (Ed.)
Instrumentation, Measurement, Circuits and Systems, 2012
ISBN 978-3-642-27333-9

Vol. 128. David Jin and Sally Lin (Eds.)
Advances in Multimedia, Software Engineering and Computing Vol.1, 2011
ISBN 978-3-642-25988-3

Vol. 129. David Jin and Sally Lin (Eds.)
Advances in Multimedia, Software Engineering and Computing Vol.2, 2011
ISBN 978-3-642-25985-2

Vol. 130. Kusum Deep, Atulya Nagar, Millie Pant, and Jagdish Chand Bansal (Eds.)
Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20–22, 2011, 2012
ISBN 978-81-322-0486-2

Kusum Deep, Atulya Nagar, Millie Pant,
and Jagdish Chand Bansal (Eds.)

Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20–22, 2011

Volume 1

Springer

Editors
Dr. Kusum Deep
Department of Mathematics
Indian Institute of Technology Roorkee
Uttarakhand, India

Dr. Millie Pant
Department of Paper Technology
Indian Institute of Technology Roorkee
Uttarakhand, India

Prof. Atulya Nagar
Department of Computer Science
Liverpool Hope University
Liverpool, UK

Dr. Jagdish Chand Bansal
ABV-Indian Institute of Information
Technology and Management
Gwalior, India

ISSN 1867-5662 e-ISSN 1867-5670
ISBN 978-81-322-0486-2 e-ISBN 978-81-322-0487-9
DOI 10.1007/978-81-322-0487-9
Springer New Delhi Heidelberg New York Dordrecht London

Library of Congress Control Number: 2011944177

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Today, there are many real world complex problems that cannot be easily dealt with traditional mathematical methods. If the user is not very conscious about the exact solution of the problem at hand, then soft computing techniques come into picture and provide affordable solutions. Soft Computing represents a collection of computational techniques inheriting inspiration from Evolutionary Algorithms, Nature Inspired Algorithms, Bio-Inspired Algorithms, Neural Networks and Fuzzy Logic.

Soft Computing techniques are gaining much popularity in recent years due the fact that real world problems have become increasingly large, complex and dynamic. The size and complexity of the problems nowadays require the development of methods which can give the solution within a reasonable amount of time rather than an ability to guarantee the exact solution.

The theme of the International Conference on Soft Computing for Problem Solving (SocProS 2011) lies in Soft Computing and its applications to solve various real world problems. SocProS 2011 turned out to be a unique forum of researchers and practitioners to present advances in this ever growing field.

SocProS 2011 attracts a wide spectrum of thought-provoking research papers on various aspects of Soft Computing with umpteen applications, theories, and techniques. A total 194 research papers are selected for publication in the Proceedings, which is in Volume 1 and Volume 2.

The editors would like to express their sincere gratitude to the Plenary Speakers, Invited Speakers, Reviewers, Programme Committee Members, International Advisory Committee, Local Organizing Committee, without whose support the quality and standards of the Conference as well as this Proceedings would not have seen the light of the day.

On the Institutional side, we would like to express our gratitude to The Institution of Engineers (India), Roorkee Local Centre, Indian Institute of Technology Roorkee Campus, Roorkee, India to provide us a platform to host this Conference. Thanks are also due to the various sponsors of SocProS 2011.

We hope that the papers contained in this proceeding will prove helpful toward improving the understanding of Soft Computing at teaching as well as research level and will inspire more and more researchers to work in the field of soft computing.

Kusum Deep, IIT Roorkee, India
Atulya Nagar, LHU, Liverpool, UK
Milie Pant, IIT Roorkee, India
Jagdish Chand Bansal, ABV-IIITM Gwalior, India

Contents

Evolutionary Technique Based Compensator for Z – Shaped Pantograph System

Ashish Kumar Jain[1], Lini Mathew[2], and Shiv Kumar Tomar[3]

[1] Department of Electronics & Instrumentation Engineering
I.E.T., M.J.P.Rohilkhand University, Bareilly (U.P.)
akjain00@gmail.com
[2] Department of Electrical Engineering,
NITTTR, Chandigarh
lenimathew@yahool.com
[3] Department of Electronics & Communication Engineering,
I.E.T., M.J.P.Rohilkhand University, Bareilly (U.P.)
shivktomar@gmail.com

Abstract. This paper shows the importance of the model order reduction using Particle Swarm Optimization (PSO) technique by designing a compensator to obtain the desired specifications for a higher order Pantograph system. The PSO is an evolutionary technique of model order reduction in which an algorithm is developed to choose best solution among many possible groups of solutions, by minimizing the fitness function. In this paper the fitness function chosen is Integral Square Error which is a measure of Performance Index of the compensator of pantograph system.

Keywords: Compensator, Particle Swarm Optimization, Model Order Reduction, Integral Square Error, Pantograph System.

1 High-Speed Rail Pantograph Control System

A pantograph is a device that collects electric current from overhead lines for electric trains or trams. The most common type of pantograph today is the so called half-pantograph (Also known as 'Z'-shaped pantograph). This device is evolved to provide a more compact and responsive single arm design at high speeds when the trains get faster in speed [1]. The pantograph is spring loaded device which pushes a contact shoe up against the contact wire to collect the electric current needed for running the train. As the train moves, the contact shoe starts sliding over the wire and can set up acoustical standing waves in the wires which break the contact and degrade current collection. Therefore, the force applied by the pantograph to the catenary is regulated to avoid loss of contact due to excessive transient motion.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1–10.
springerlink.com

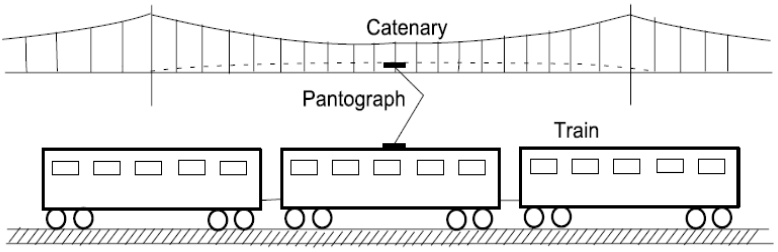


Fig. 1. High-Speed Rail System Showing Pantograph and Catenary

The electric transmission system for modern electric rail systems consists of an upper load carrying wire (known as a catenary) from which a contact wire is suspended as shown in Fig.1. The pantograph is connected between the contact wire and the electric contact.

2 Particle Swarm Optimization

In PSO, the 'swarm' is initialized with a population of random solutions. Each particle in the swarm is a different possible set of the unknown parameters to be optimized. Representing a point in the solution space, each particle adjusts its flying toward a potential area according to its own flying experience and shares social information among particles. The goal is to efficiently search the solution space by swarming the particles toward the best fitting solution encountered in previous iterations with the intent of encountering better solutions through the course of the process and eventually converging on a single minimum error solution. The position corresponding to the best fitness is known as $pbest$ and the overall best out of all the particles in the population is called $gbest$. The velocity update in a PSO consists of three parts; namely momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm. The parameters c_1 and c_2 determine the relative pull of $pbest$ and $gbest$ and the parameters r_1 and r_2 help in stochastically varying these pulls. The modified velocity and position of each particle can be calculated using the current velocity and the distances from the $pbest_{j,g}$ to $gbest_g$ as shown in the following formulae [2, 3].

$$v_{j,g}^{(t+1)} = w * v_{j,g}^{(t)} + c_1 * r_1(\ ) * (pbest_{j,g} - x_{j,g}^{(t)}) + c_2 * r_2(\ ) * (gbest_g - x_{j,g}^{(t)}) \quad (1)$$

$$x_{j,g}^{(t+1)} = x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \quad (2)$$

where w is inertia factor which balances the global wide-rang exploitation and the local nearby exploration abilities of the swarm.

The computational flowchart of the PSO process [4] is given in Fig.2.

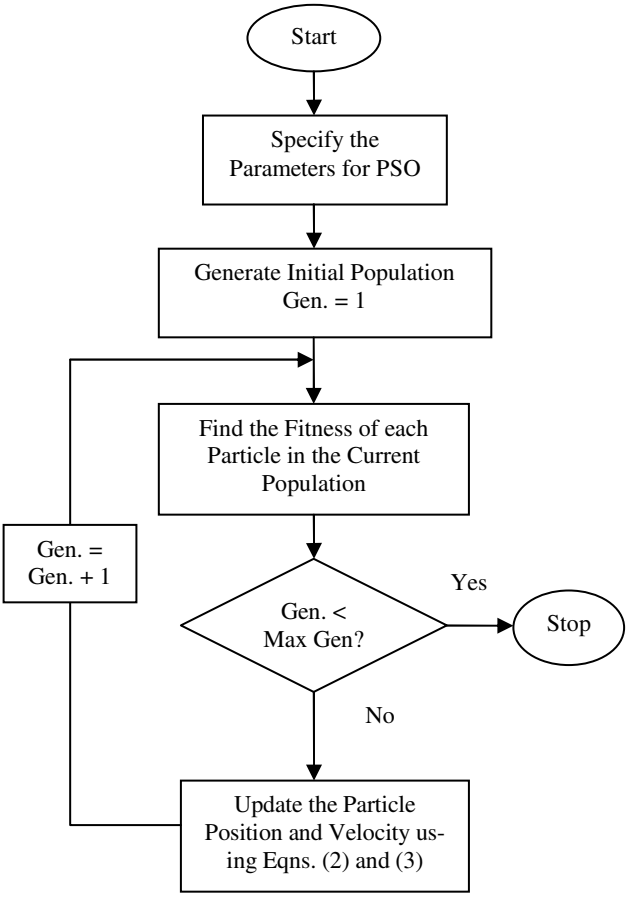


Fig. 2. Flow Chart of PSO Algorithm

3 Pantograph Dynamic System

The proposed method to regulate the force between the pantograph shoe and the electric contact uses a closed-loop feedback system, whereby a force is applied to the bottom of the pantograph, resulting in an output force applied to the catenary at the top. Fig.3 shows the pantograph and the catenary coupling. Contact between the head of the pantograph and the catenary is represented by a spring. Output force is proportional to the displacement of the spring, which is the difference between the catenary and pantograph head vertical positions.

Equations of motion are:

$$F_{up} - f_{vf}\dot{y}_f - K_h(y_f - y_h) - f_{vh}(\dot{y}_f - \dot{y}_h) = M_f\ddot{y}_f \tag{3}$$

$$K_h(y_f - y_h) + f_{vh}(\dot{y}_f - \dot{y}_h) - K_s(y_h - y_{cat}) = M_h\ddot{y}_h \tag{4}$$

$$K_s y_h - (K_s + K_{ave}) y_{cat} = 0 \tag{5}$$

Transfer function of Pantograph Dynamics is obtained by putting the values of all the variables e.g. $K_{ave} = 1.535 \times 10^6\, N/m$, $K_s = 82.3 \times 10^3\, N/m$, $K_h = 7 \times 10^3\, N/m$, $M_f = 17.2\, kg$, $M_h = 9.1 kg$, $f_{vh} = 130\, Ns/m$ and $f_{vf} = 30\, Ns/m$ in equations (3), (4) and (5). By solving these equations the transfer function is determined [5] as:

$$G(s) = \frac{y_h(s) - y_{cat}(s)}{F_{up}(s)} = \frac{0.7883(s + 53.85)}{s^4 + 23.59\, s^3 + 9785\, s^2 + 81190\, s + 3.493 \times 10^6} \tag{6}$$

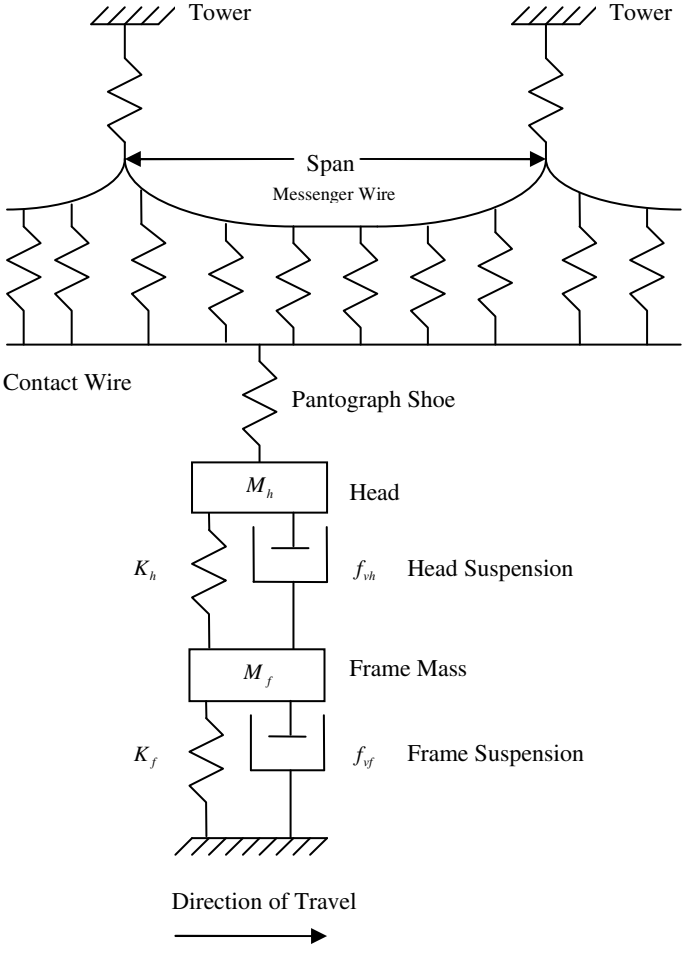


Fig. 3. Coupling of Pantograph and Catenary

A functional block diagram shown in Fig.4 depicts the various signals i.e. the desired output force as the input, the force F_{up} applied to the bottom of the pantograph dynamic, the difference in displacement between the catenary and the pantograph head and the output contact force. It also shows blocks representing the input transducer, controller, actuator generating F_{up}, pantograph dynamics, spring and the output sensor.

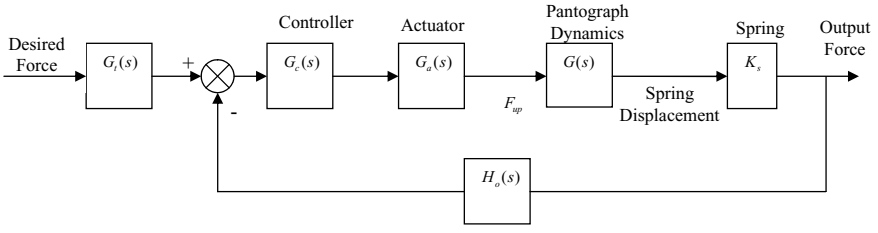


Fig. 4. Functional Block Diagram of an Active Pantograph

Using input transducer ($G_t(s)$ = 1/100), controller ($G_c(s) = K$), actuator ($G_a(s)$ =1/1000), pantograph spring (K_s = 82.3 $\times 10^3$ N/m), and sensor ($H_o(s)$ = 1/100), the closed-loop transfer function [6] comes out to be:

$$G_{cl}(s) = \frac{KG(s)(0.823)}{1 + KG(s)(0.823)} \tag{7}$$

$$G_{cl}(s) = \frac{G_p(s)}{1 + G_p(s)} \tag{8}$$

where,

$$G_p(s) = KG(s)(0.823) \tag{9}$$

Equation (8) represents a closed-loop system, part of which, the $G_p(s)$ is open-loop system shown in Fig.5. By applying Routh's stability criterion, limit of K is determined. For the system to be stable:

$$K < 188145.26 \tag{10}$$

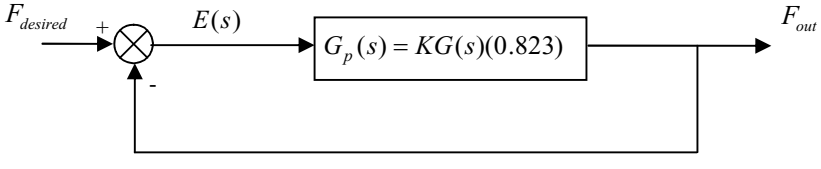


Fig. 5. Equivalent Block Diagram of an Active Pantograph Control System

The steady state error e_{ss} of the system is given by [7]:

$$e_{ss} = \lim_{s \to 0} \; s \times E(s) \tag{11}$$

$$e_{ss} = \lim_{s \to 0} \; \frac{1}{1 + KG(s)(0.823)} \tag{12}$$

For steady state error to be minimum, K should be maximum using equation (12) and satisfying the condition of stability equation (10) derived above, let's assume:

$$K = 188000$$

With this choice of K and by using equations (9), we get:

$$G_p(s) = 188000 \times \frac{0.823 \times 0.7883(s + 53.85)}{s^4 + 23.59\, s^3 + 9785\, s^2 + 81190\, s + 3.493 \times 10^6}$$

$$G_p(s) = \frac{121968.9(s + 53.85)}{s^4 + 23.59\, s^3 + 9785\, s^2 + 81190\, s + 3.493 \times 10^6} \tag{13}$$

Using equation (8), the closed-loop transfer function of the system comes out to be:

$$G_{cl}(s) = \frac{121968.93s + 6.568 \times 10^6}{s^4 + 23.589s^3 + 9784.9s^2 + 203158.97s + 1.006 \times 10^7} \tag{14}$$

The unit step response of this system is plotted using MATLAB's step command and is shown in Fig.6. Looking at the response, the settling-time seems to be around T_s = 400 sec. and the damping-ratio of the system is very small resulting in so many repulses before steady-state.

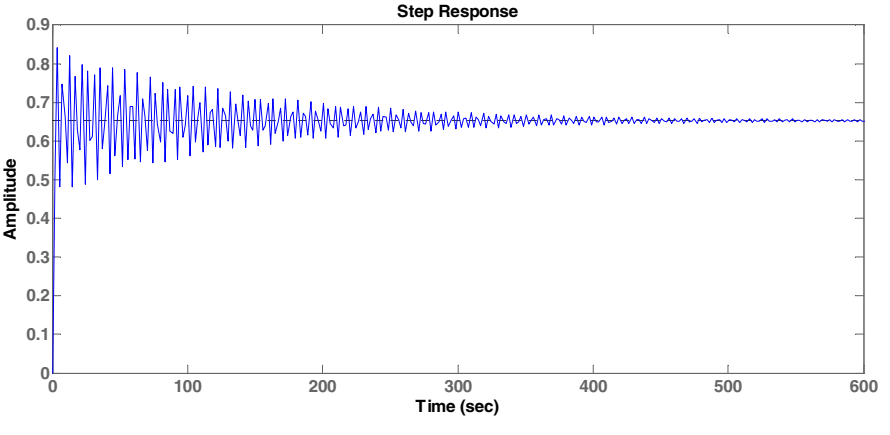


Fig. 6. Unit Step Response of an Active Pantograph Control System

One of the ways to eliminate the high frequency oscillations is to cascade a notch filter with the plant. Transfer function of the notch filter is given by equation (15) and Fig.7 shows the block diagram of a Notch filtered Pantograph control system [5].

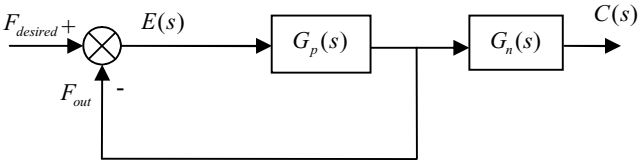


Fig. 7. Block diagram of a Notch Filtered Closed-Loop System

$$G_n(s) = \frac{s^2 + 16s + 9200}{(s+60)^2} \tag{15}$$

Step response of the notch filtered closed-loop system is plotted in Fig.8.

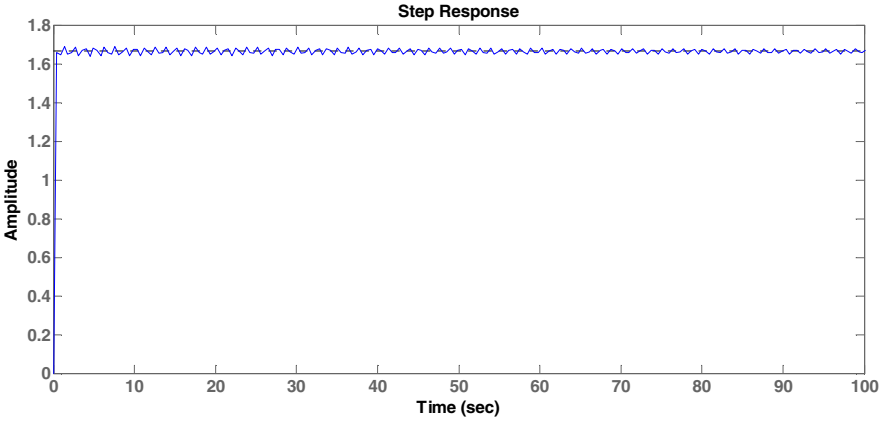


Fig. 8. Unit Step Response of Notch Filtered Active Pantograph Control System

Problem statement is to design a Proportional and Derivative (PD) controller as $G_c(s)$ shown in Fig.9 for the above active pantograph system to yield a settling-time of approximately 0.3 sec with no more than 20% overshoot.

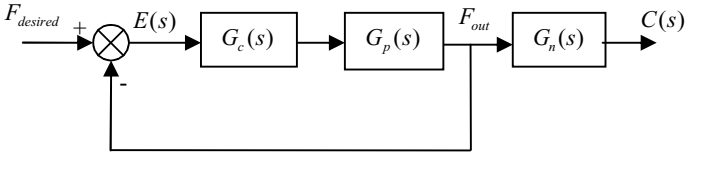


Fig. 9. Block diagram of Notch Filtered Active Pantograph Control System

$$\text{Settling time} = t_s = \frac{4}{\zeta \omega_n} = 0.3 \tag{16}$$

$$\text{Overshoot} = e^{-\zeta\pi/\sqrt{1-\zeta^2}} = 0.2 \tag{17}$$

By solving equations (16) and (17), we obtain the value of damping ratio ζ and natural frequency ω_n as 0.456 and 29.2 respectively.

The resultant reference model is:

$$R_m(s) = \frac{C(s)}{F_{desired}(s)} = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} = \frac{854.8}{s^2 + 26.67s + 854.8} \tag{18}$$

The open-loop transfer function $G_p(s)$ from equation (13) is reduced from 4th to 2nd order using proposed algorithm.

$$R_p(s) = \frac{19.34s + 830.7}{s^2 + 9.082s + 441.8} \tag{19}$$

where $R_p(s)$ is the reduced order transfer function of the original 4th order plant transfer function $G_p(s)$.

The fitness function J in the proposed algorithm is taken as an integral squared error of difference between the responses given by the expression:

$$J = \int_0^{t_\infty} [y(t) - y_r(t)]^2 \, dt \tag{20}$$

Where $y(t)$ and $y_r(t)$ are the unit step responses of original and reduced order systems.

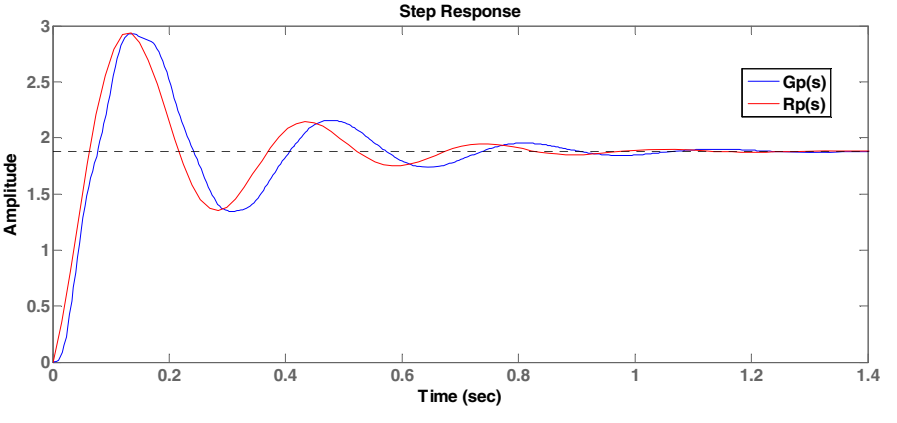


Fig. 10. Comparison of Step Responses of the Original and the Reduced Order Transfer Functions

The values of parameters e.g, swarm size, number of iterations, w, c_1 and c_2 used for implementation of this algorithm are 25, 10, 0.4, 1.5 and 1.5 respectively. The step responses of both the original $G_p(s)$ and the reduced order plant $R_p(s)$ are compared in Fig.10.

From Fig.9, we get:

$$\frac{C(s)}{F_{desired}(s)} = \frac{G_c(s)R_p(s)}{1+G_c(s)R_p(s)} \times G_n(s) = R_m(s) \tag{21}$$

$$G_c(s)R_p(s) = \frac{R_m(s)/G_n(s)}{1-R_m(s)/G_n(s)} \tag{22}$$

By putting the values of transfer functions $G_n(s)$ and $R_m(s)$ from equations (15) and (18) respectively into the equation (22), we get:

$$G_c(s)R_p(s) = \frac{854.8s^2+102576s+3077280}{s^4+42.67s^3+9626.7s^2+156464.8s+4786880} \tag{23}$$

By substituting value of transfer function of $R_p(s)$ from equation (19) into equation (23), we get:

$$G_c(s) = \frac{854.8s^4+1.103\times10^5 s^3+4.38\times10^6 s^2+7.32\times10^7 s+1.36\times10^9}{19.34s^5+1656s^4+2.21\times10^5 s^3+1.102\times10^7 s^2+2.2\times10^8 s+3.97\times10^9} \tag{24}$$

$$G_c(s) = 0.46\times\frac{s^2+12.78s+407.05}{s^2+16.088s+547.82} \tag{25}$$

It is approximated as 2nd order compensator by applying indirect method of order reduction [8] in equation (25).

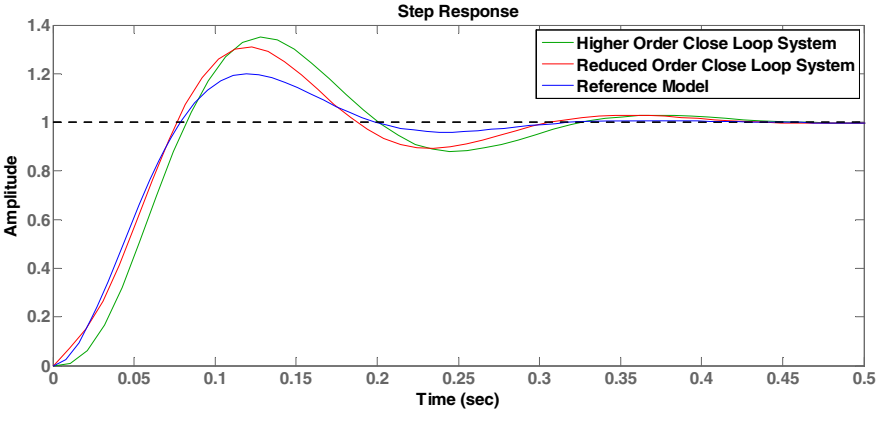


Fig. 11. Comparison of Step Responses of the Original and the Reduced Order Closed-Loop Systems

Fig.11 shows the comparison of step responses of the reference model, closed loop transfer function of the original plant with implementation of designed compensator and that of the reduced model implemented with same compensator. It is seen that all the three responses are approximately matching both in steady state and transient stable regions.

Table 1. Comparison of Different Time Specifications of Step Responses

Model	Rise-Time (Sec)	Settling-Time (Sec)	% Overshoot	Peak-Time (Sec)
Reference Model	0.053	0.28	19.96	0.1192
Higher Order Model	0.050	0.40	35.2	0.1277
Reduced Order Model	0.055	0.38	30.93	0.1224

Table1 shows the comparison of time durations of step responses of the reference model, closed-loop transfer function of the original plant and that of the reduced model. It is obvious from the results that the various specifications are quiet comparable to each other.

The authors have used other evolutionary techniques such as GA, ACO etc. in the other related work which will be published in near future.

References

[1] O' Connor, D.N., Eppinger, S.D., Seering, W.P., Wormly, D.N.: Active Control of a High Speed Pantograph. Journal of Dynamics System, Measurement and Control 119, 1–4 (1997)

[2] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1945 (1995)

[3] Parmar, G., Prasad, R., Mukherjee, S.: Reduced Order Modeling of Linear MIMO Systems using Genetic Algorithm. International Journal of Modeling and Simulation 6(3), 173–184 (2007)

[4] Tomar, S.K., Jain, A.K.: Particle Swarm Optimization Based Reduced Order Models for Controller Design. In: XVII International Conference on System Science, Poland, p. 43 (2010)

[5] Nise, N.S.: Control System Engineering, pp. 256–258. John Wiley & Sons, United States of America (2000)

[6] Bandi, P.: High-Speed Rail Pantograph Control System Design, pp. 1–7 (2009)

[7] Ogata, K.: Modern Control Engineering, pp. 608–610. Prentice - Hall of India Private Limited (1996)

[8] Tomar, S.K., Prasad, R.: A Indirect approach of model order reduction of linear Time-invariant systems using truncation method. In: XXXII National Systems Conference, NSC 2008 (2008)

Study on Ductility of Ti Aluminides Using Mamdani Fuzzy Inference System

R.K. Gupta[1], Bhanu Pant[1], P.P. Sinha[1], Rama Mehta[2], and Vijaya Agarwala[3]

[1] Materials and Mechanical Entity, VSSC, Trivandrum-22, India
rohitkumar_gupta@vssc.gov.in
[2] National Institute of Hydrology, Roorkee-247 667, India
rama@nih.ernet.in
[3] Dept. of Met. & Mat. Engg., Indian IIT, Roorkee-247 667, India
vijayfmt@iitr.ernet.in

Abstract. Application of Ti aluminide is limited due to its poor ductility at room temperature. Over the last two decades several studies have been conducted and large amount of data have been generated. These data have been utilized in the present work to obtain optimized set of parameters through Mamdani fuzzy inference system. Ductility database were prepared and three parameters viz. alloy type, grain size and heat treatment cycle were selected for modeling. Additionally, ductility data were generated from literature for training and validation of models on the basis of linearity and considering the primary effect of these three parameters. All the data have been used to frame the fuzzy rules with its membership values. Performance of the models was evaluated, which shows model has better agreement with the data generated from observed data. Possibility of improving ductility more than 5% is observed for multicomponent alloy with grain size of 10-50μm using a multistep heat treatment cycle.

Keywords: Ti aluminide, ductility, Mamdani, Fuzzy.

1 Introduction

Ti aluminide has been an important aerospace material due to its high temperature properties and lower density as compared to superalloys. The ordered structure of aluminide, which makes them useful for high temperature applications, renders them brittle at ambient temperature [1-3]. Therefore, inspite of having good properties, the usefulness of these alloys has been limited to some specific applications only. Room temperature tensile ductility is maximum (~1.5%) at around Ti-48Al (at%) aluminum. Hence, development of Ti aluminides has been centered around Ti-48Al (at%) composition, which belongs to the γ (TiAl) plus α_2 (Ti_3Al) region of the phase diagram [4-6]. Ductility of binary alloy is insufficient for further processing and applications. Therefore, various methods like alloy

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 11–22.
springerlink.com

addition, controlled processing, heat treatment, etc. are applied to get optimum combination of strength and ductility [7-16]. Several studies have been conducted with limited success in improving ductility of the alloy. However, these studies are expensive due to the use of high purity alloying elements and processing under controlled atmosphere. Therefore, there is a need of theoretical model for optimization of process parameters. Experimentation with such optimized parameters shall minimize the number of observed attempts and could lead to achieve desired ductility.

During the last decade, there has been an increased interest in applying new emerging theoretical techniques such as Fuzzy Inference System (FIS) and Artificial Neural Network (ANN) for optimization related problems [17-19]. These are the most common data driven models. These models intend to describe the non-linear relationship between input (antecedent) and output (consequence) to the real system.

Fuzzy inference systems are the most important modelling tool based on fuzzy set theory. Conventional fuzzy inference systems are typically built by domain experts and have been used in automatic control, decision analysis and expert systems. Optimization and adaptive techniques expand the applications of fuzzy inference systems to fields such as adaptive control, adaptive signal processing, non- linear regression, and pattern recognition. The key idea about fuzzy logic theory is that it allows for something to be partly this and partly that, rather than having to be either all this or all that. In fuzzy logic, variables are "fuzzified" through the use of membership function (MF) that defines the membership degree to fuzzy sets and fuzzy objectives and constraints. It can be of different forms and in categories. Methods of constructing MF from statistical data are described elsewhere [20-24]. Fuzzy rule base system can be used as suitable representation of simple and complex physical systems. The fuzzy rule based model operates on an "IF-THEN" principle, where the "IF" is a vector of fuzzy explanatory variables or premises and "THEN" is fuzzy consequence. Two types of fuzzy inference systems are mainly used: Mamdani type[17] and Takagi Sugeno type [25]. Mamdani fuzzy inference system (MFIS) is the most commonly used fuzzy methodology. This is the first control system built method using fuzzy set theory. Within the inference, fuzzy sets from the consequent of each rule are combined through the aggregation operator to get the resulting fuzzy set. After the aggregation process there is a fuzzy set for each output variable. The crisp value of output variable is obtained through defuzzification method. A typical model processing is explained through flow diagram in Fig. 1. A large number of inputs and multi-output can be dealt with in modeling and as the number of inputs in a fuzzy rule-base system increases, the number of rules quickly becomes too large, unidentifiable, and unmanageable. But in present study three inputs and single output are used. One of the major concerns in modeling is the identification of the appropriate membership functions [21-27] for each input that represent the transformation of membership degrees into output. To obtain a simple, transparent, accurate and reliable model, the shape and type of MFs have been chosen according to the variation and nature of input data. Also the fuzzy rules have been framed with proper weights according to the fuzzy rule base matrix for all input data together.

In the present work, MFIS models for prediction of ductility of Ti aluminide intermetallics are developed. The models were validated with the literature database using three important indices i.e. root mean square error (RMSE), coefficient of determination (R2) and the model efficiency criterion of Nash and Sutcliffe [28].

2 Ductility Parameters and Data for Model

Ductility values were consolidated from literature [5-11, 29-34] and three important parameters viz. alloy chemistry, grain size and heat treatment cycles were identified, which have major influence on the ductility of the alloy. Accordingly, collected data were categorized into 2 types of alloys (2 component and multi-component alloy), 4 types of grain sizes and 7 types of heat treatment cycles with certain assumptions (Table 1). Ductility values collected from literature with various categories of parameters are presented in Table 2, which were further interpolated according to trends in literature values and following the linearity basis. In this way 56 ductility data combinations were generated, in the same order as given in Table 2, for each alloy type and for each grain size in different heat treatment conditions. Total number of observed values is 56, which is used for testing. The ranges of input data have been divided in different membership categories to frame the fuzzy rules for Mamdani fuzzy Inference method.

Table 1. Assumptions for categorizing literature data [18]

Sl. no.	*Assumptions*
1.	Effect of alloying elements like Cr, V and Mn is same on ductility.
2.	Effect of alloying elements on various alloys Ti 44-52Al is same as on Ti48Al
3.	Grain size means diameter of grain for equiaxed grains and interlamellar spacing for lamellar grain.
4.	Value of grain size for different composition Follow the same trend as it follow for a specific composition and the trend is extrapolated.
5.	Ductility data referred from literature is for mean Grain diameter.
6.	Ductility data for alloy type and grain size is the Maximum ductility for the alloy in desirable heat treatment condition.

3 Model Development

Three variables composition, grain size and heat treatment cycles were considered as antecedent and ductility as consequence during the modeling.

3.1 Formulation of MFIS Model

Models were developed by Mamdani Fuzzy Inference System (MFIS), which identified suitable numbers of fuzzy if-then rules and associated membership

function. Calculation of the degree of fulfillment (DOF) was carried out for each rule. There are five parts of MFIS: fuzzification of the input variables, application of the fuzzy operator, implication and aggregation of the consequents across the rules, and defuzzification.

In fuzzification of input data, fuzzy operators like fuzzy intersection or conjunction (AND), fuzzy union or disjunction (OR), and fuzzy complement (NOT) are used. These functions are known as the classical operators as AND = min, OR = max, and NOT = additive complement. The fuzzy logic operators such as the AND or OR operators obey the classical two valued logic. The AND methods are supported for min (minimum) and Prod (Product) while the OR methods are supported for max (maximum) and the Probor (Probabilistic Or method). The Probabilistic Or method is calculated according to the equation

$$\text{Probor}(a, b) = a + b - ab \quad (1)$$

Table 2. Parameters and ductility values [18]

Parameters	*Detail Characteristics*	*Ductility (%e)*
Alloy type	1. Binary alloy (Ti-48Al at%) [5, 6, 30]	2.1
	2. Multicomponent alloy (Ti-48Al + 0-4M*) [9, 11]	3.0
Grain size	1. 10 μm [31, 33]	2.0
	2. 50 μm [31, 33]	1.8
	3. 100 μm [34]	1.4
	4. 250 μm [34]	1.1
Heat treatment cycles	1. Above Tα (2 hrs) then FC to RT [34]	0.5
	2. Just below Tα (20 min.) then AC to RT [11]	1.5
	3. Below Tα (100^0C) soaking (2 hrs) then AC to RT [11]	3.6
	4. Above Tα (1 hr) then CC1 to below Te (soaking for 6hrs) then AC to RT [8]	0.8
	5. Just above Tα (30 min) then CC2 to RT + heating below Te and soaking for 6 hrs. then AC to RT [8]	2.8
	6. Just below Tα (3 hrs.) then CC1 to below Te (soaking for 4 hrs.) then AC to RT [8]	3.8
	7. Soaking just below Tα for 4 hrs. then FC to (Tα-50^0C) and soaking for 4 hrs. then FC to RT + heating and soaking below Te (24 hrs) then AC to RT [34]	5.0

M is alloying element (Cr/ Mn/ V at the desired content ranging from 0-4 at%, leading to maximum ductility). Tα (α transus temperature for specific alloy), RT (room temperature), FC (furnace cooling), AC (air cooling), CC1 (controlled cooling 80^0C/min), CC2 (controlled cooling 100^0C/min), Te (eutectoid temperature of alloy).*

3.1.1 Formulation of Fuzzy Rule-Base

A fuzzy rule based model was made, where fuzzy rule system R, is defined as the set of rules which consist of sets of input variables or premises $A_{I,k}$ in the form of fuzzy sets with membership functions $\mu_{Ai,k}$ and a set of consequences B, also in the form of a fuzzy set.

If a_1 is $A_{I,1}$ & a_2 is $A_{I,2}$ & ---------- a_k is $A_{I,k}$ then B_I is the set of consequences.

Problems are encoded into rules described in the following form:

The fuzzy rule base was formed based on the data trend, intuition and logic. From the data a matrix of rule base was prepared based on the range of input parameters.

The common fuzzy categories commonly used are EL = Extremely low, VL = Very low, L = Low, M = Medium, H = High, VH = Very High, & EH = Extremely High

The rule based algorithm is as follows

If the Input 1 is *EL* and Input 2 is *VH* and the Input 3 is *M,* then the Output is *L.*

It is claimed that such rules more closely resemble the way we think than do mathematical rules. Fuzzy rule-based modeling is centred on the definition and verification of rule system. Rule system response depend on the choice of both operator and defuzzification method [24].

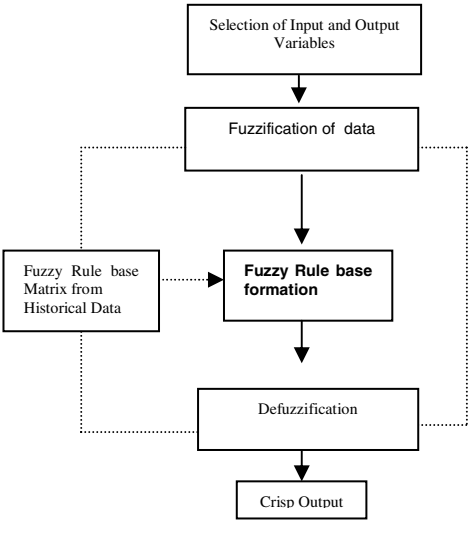


Fig. 1. Flow diagram of Mamdani Fuzzy Inference System

3.1.2 Implication and Aggregation

The fuzzy operator operating on the input fuzzy sets provides a single value corresponding to the inputs in the premise. This result is applied on the output membership function through implication method to obtain a fuzzy set for fuzzy

rule. Each input in each rule goes under the defined operator and then it gives a single output fuzzy set. After the implication of the fuzzy operations, the same number of output is obtained as the number of rules framed i.e. each rule will have its own single output. Rule's weight is applied before applying the implication method. The users can change the weight (between 0 and 1) of the rule according to the rule depending upon how much it is close to the reality.

Then aggregation occurs for each output variable as a final step, where all the fuzzy sets are combined into a single fuzzy set. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. A typical algorithm of MFIS is presented in appendix-A.

3.1.3 Defuzzification

The input for the defuzzification is a fuzzy set (the aggregated output fuzzy set) and the output is a single crisp number. The most popular defuzzification method is the Centroid method, which returns the center of area under the curve. Centroid method is used here because it gives results using center of curve obtained by aggregation method. This is used when number of rules are large.

3.2 MFIS Model Calibration

The rule system responses depend on the selection of parameter combination and defuzzification methods. During the present modeling, Fuzzy rules are defined as: IF (Composition is A) AND (Grain size is B) AND (Heat treatment cycle is C) THEN (Ductility is D). In all, 55 rules have been framed with Mamdani fuzzy rule matrix. Typical examples of fuzzy rules are:

Rule : IF composition is binary (Ti48Al) alloy, AND grain size is 10 μm AND heat treatment cycle is 1st one (of Table 2) THEN ductility is 1%.

Rule : IF composition is binary (Ti48Al) alloy, AND grain size is 10 μm AND heat treatment cycle is 3rd one (of Table 2) THEN ductility is 4%.

All three inputs are fuzzified with membership functions in Fig. 2 of output 'Ductility'. MFIS structure with MFs of inputs and output is presented in Fig. 3. In present formulation, all data have been used together to frame the rules with its degree of membership. All 56 values have been used to frame the fuzzy rule matrix.

The number of rules in the fuzzy rule base is $\prod_{i=1}^{n} c_i$, where c_i is the number of categories in the ith input and n is the total number of inputs. In this problem, inputs are alloy type, grain size and heat treatment cycles with 2, 4 and 7 categories of MFs respectively. Guassian type MF has been used for alloy type while Triangular MFs have been used for other two inputs. For every input vector, a membership degree to each fuzzy set greater than zero is computed from the membership function. Therefore, all the rules apply simultaneously for each combination of inputs and a crisp output value using centroid defuzzyfication method is obtained for the given data.

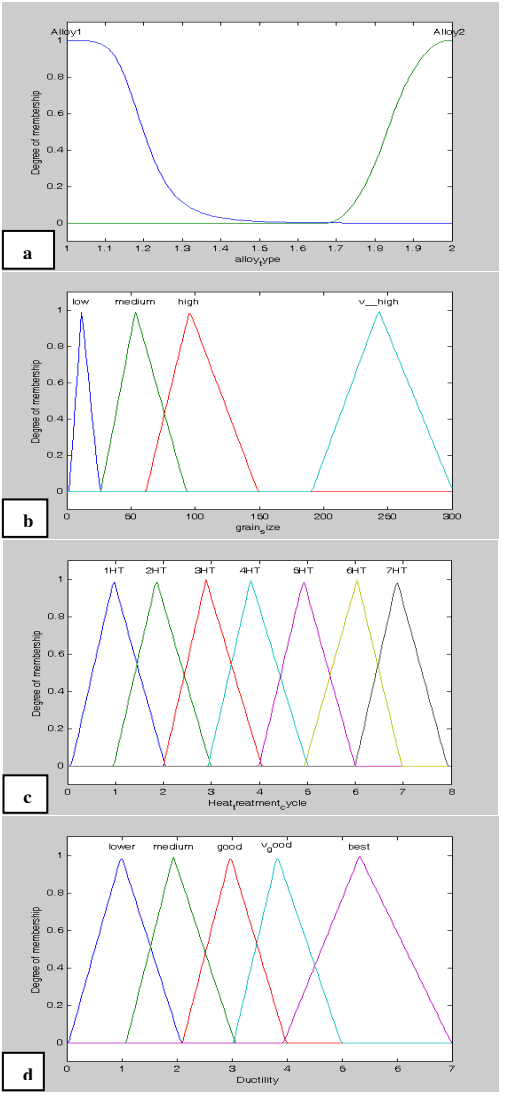


Fig. 2. Membership function for MFIS (a) Input 1 - two alloy types, (b) Input 2 - four grain size (c) Input 3 - seven heat treatment cycles, (d) output-ductility

4 Performance Indices

All models developed with different sets of data were tested and computed ductility was compared with literature-based ductility by means of RMSE (root mean square error) and R^2 (regression coefficient) statistics. The models performances were also evaluated by using Nash-Sutcliffe criterion [28] of percent variance (VAREX) explained as given below:

$$VAREX = \left[1 - \frac{\sum_{t=1}^{N} (O_t - P_t)^2}{\sum_{t=1}^{N} (O_t - \bar{O})^2} \right] x100 \quad (2)$$

where N is number of observations; O_t is observed value at time t (m^3/s); $\bar{O}$ is mean of the observed values (m^3/s); and P_t is predicted ductility at time t (m^3/s). The value of VAREX ranges from 0 (lowest performance) to 100 (highest performance).

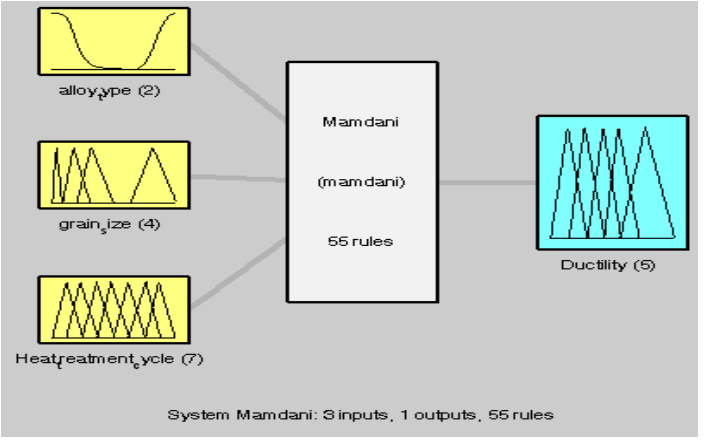


Fig. 3. MFIS structure showing inputs and output with membership functions

5 Results and Discussion

Artificial Intelligence serves as a smooth interface between the qualitative variables and the numerical domains of the inputs and outputs of the model. As seen from the literature database, ductility of Ti aluminide is a function of number of variables, which are qualitative in nature. However, it can be presented mathematically where certain variables have primary effect on ductility of intermetallics and the same has been considered in the present work. The rule-base nature of the model allows the use of qualitative metallurgical relations expressed as general effect of different variables, which makes the model transparent to interpretation and analysis. Among the emerging modeling techniques, ANN and Mamdani fuzzy model have been considered to be useful. Also, models are developed with the data generated from literature database with certain realistic assumptions and using different sets of data for training and validation. This makes the model more reliable.

In Mamdani Fuzzy method, all the data is used to frame the fuzzy database matrix because it needs to develop rules manually and better rules can be made with full length of data. Results are very good within 98% variance because here the model is trained with data at all conditions. Results of Mamdani model is presented through Figs 4-6. Predicted ductility values are very close to observed values. Here achieved ductility value is nearly 5.55% in six specific combinations

of parameters (as shown in figure 4). This level of ductility is obtained for multicomponent alloy with small grain size (10-50 μm) following heat treatment no 3, 6 and 7 of Table 2. It means alloying elements have important role, which improve ductility with finer grains and heat treatments carried out mostly below the Tα.

Table 3. Performance indices

MODEL	*RMSE*	*Regression Coefficient*	*VAREX*
Fuzzy Mamdani	0.0528	0.9383	93.49

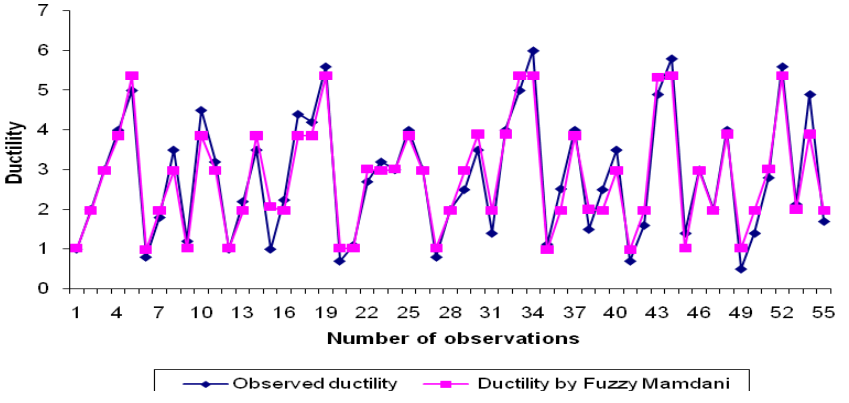


Fig. 4. Comparative results of observed and modeled ductility

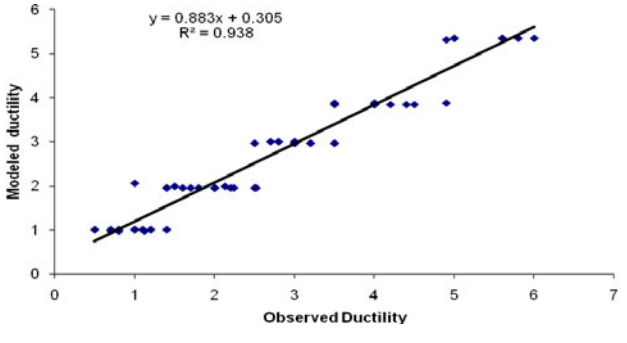


Fig. 5. Linear relationship of observed and modeled ductility

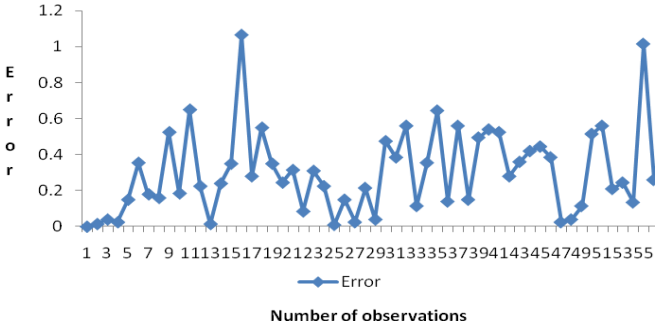


Fig. 6. Error graph

The maximum model error is found to be 1.065 otherwise model error is ranging from 0 to 0.62 (Fig.6). The performance of the model is shown in Table 3. It is clearly seen that RMSE is 0.0528, regression coefficient and VAREX are 93.83% and 93.49% respectively. It indicates that all the three performance indicators are close to ideal indices.

Linear relationship of ductility obtained by MFIS model developed for whole dataset is having good correlation with observed (literature data) output confirming the reliable prediction characteristics of the developed model. Scattering of data values are more for ANN with dataset 1 (Fig. 3a) and least for ANN with dataset 3 (Fig. 3c) and Mamdani with dataset 4 (Fig. 10). It also shows predicted ductility is close to reported value and it follows a definite trend. Therefore, use of different combinations of data sets for training and validation gives better prediction of property in ANN and MFIS models. It means both this model would be helpful in deriving optimum combination of parameters to obtain maximum ductility in Ti aluminide intermetallics.

6 Conclusions

1. Models developed through MFIS are giving results close to the observed ductility value specially with a good length of data.
2. Mamdani is found to be more efficient as it framed rules using all data environment. It is user friendly.
3. Error (difference between observed and modeled values) for MFIS model is ranging between 0 to 1. Highest error 1.0 is observed at two places only otherwise it ranges from 0 to 0.62.
4. Regression coefficients are 0.938 in MFIS modeling.
5. Achievable ductility value is 5.55% in specific combinations of parameters (6 cases). It is obtained for multicomponent alloy with small grain size (10-50 μm) following heat treatment no 3, 6 and 7 of table 2. It means alloying elements have important role, which improves ductility with finer grains and heat treatments carried out mostly below the Tα.

Acknowledgements. Authors extend their thanks to Director, VSSC for permitting to publish this work.

References

1. Lipsitt, H.A.: High Temperature Ordered Intermetallic Alloys. In: Koch, C.C., Liu, C.T., Stolloff, N.S. (eds.) MRS, vol. 39, p. 351 (1985)
2. Taub, A.I., Fleischer, R.L.: Science 243, 616 (1989)
3. Huang, S.C., Cheshut, J.C.: Gamma Ti Al and its alloys-Intermetallic compounds. In: Westbrook, J.H., Fleischer, R.L. (eds.), vol. 2, p. 73. John Wiley & Sons Ltd. (1994)
4. Kattener, U.R., Lin, J.C., Chang, Y.A.: Metall. Trans. 23A, 2081 (1992)
5. Kim, Y.W., Froes, F.H.: High temperature aluminides and Intermetallics. In: Whang, S.H., Liu, C.T., Pope, D.P., Stiegler, J.O. (eds.), p. 485. The Minerals, Metals & Materials Society (1990)
6. Kim, Y.W., Froes, F.H.: High temperature aluminides and Intermetallics. In: Whang, S.H., Liu, C.T., Pope, D.P., Stiegler, J.O. (eds.), p. 465. The Minerals, Metals & Materials Society (1990)
7. Huang, S.C., Hall, E.L., Shih, D.S.: ISIJ Int. 10(31), 1100 (1991)
8. Kim, Y.W.: Acta Metall. Mater. 40(6), 1121 (1992)
9. Huang, S.C., Hall, E.L.: Acta Metall. Mater. 39(6), 1053 (1991)
10. Wang, J.N., Xie, K.: Intermetallics 8, 545 (2000)
11. Seetharaman, V., Semiatin, S.L.: Mater. Sci. & Engg. A 299A, 195 (2001)
12. Beschliesser, M., Chatterjee, A., Lorich, A., Knabl, W., Kestler, H., Dehm, G., Clemens, H.: Mater. Sci. and Engg. A 329-331, 124 (2002)
13. Imayev, V., Imayev, R., Kuznetsov, A.: Scripta Mater. vol. 49, p. 1047 (2003)
14. Clemens, H., Bartels, A., Bystrzanowski, S., Chladil, H., Leitner, H., Dehm, G., Gerling, R., Schimansky, F.P.: Intermetallics 14, 1380 (2006)
15. Perez-Bravo, M., Madariaga, I., Estolaza, K., Tello, M.: Scripta Mater. 53, 1141 (2005)
16. Wang, J.N., Yang, J., Wang, Y.: Scripta Mater. 52, 329 (2005)
17. Mamdani, E.H., Assilian, S.: Int. J. Man Machine Stud. 7(1), 1 (1975)
18. Gupta, R.K., Rama, M., Vijaya, A., Bhanu, P., Sinha, P.P.: Trans. IIM 63(6), 833 (2010)
19. Kosko, B.: Neural Networks and Fuzzy Systems. Prentice Hall of India Pvt. Ltd., India (1996)
20. Ross, T.J.: Fuzzy logic with Engineering Applications. Electrical Engineering series. Mc Graw Hill International Editions (1997)
21. Civanlar, M.R., Trussel, H.J.: Constructing membership functions using statistical data. Fuzzy Sets and Systems 18, 1 (1986)
22. Devi, B.B., Sharma, V.V.S.: Information Sci., vol. 35 (1), p. 43 (1985)
23. MATLAB Fuzzy logic toolbox. The MATH WORKS, Inc. (1995)
24. Bardossy, A., Duckstein, L.: Fuzzy rule based modeling with applications to geophysical, biological and engineering systems. CRC press, Boca Raton (1995)
25. Takagi, T., Sugeno, M.: IEEE Trans. Syst. Man. Cyber. 15, 116 (1985)
26. Dombi, J.: Fuzzy Sets and Systems 35, 1 (1990)
27. Chiu, S.L.: J. Intelligent Fuzzy System 2, 267 (1994)
28. Nash, J.E., Sutcliffe, J.V.: Hydrology J. 10, 282 (1970)
29. Huang, S.C.: Metall. Trans. 23A, 375 (1992)

30. Salishchev, G.A., Imayev, R.M., Senkov, O.N., Froes, F.H.: J. of Metals 46 (2000)
31. Imayev, R., Imayev, V., Salishchev, G.: Scripta Metall. 29, 719 (1993)
32. Imayev, R., Imayev, V., Salishchev, G.: Scripta Metall. 29, 713 (1993)
33. Hu, D.: Intermetallics 10, 851 (2002)
34. Tang, J., Huang, B., Liu, W., He, Y., Zhou, K., Wu, A., Peng, K., Qin, W., Du, Y.: Materials Research Bulletin 38, 2019 (2003)

APPENDIX-A

A typical algorithm of Mamdani fuzzy inference system can be expressed as follows. It is assumed that there are R_r (r = 1,2,…….., n) rules in the above-mentioned form.

For each implication R_r, y_r is calculated by the function f_r in the consequence

$$y_r = f_r(x_{1,}\, x_{2,}..x_p) = b_r(1) + ….+ b_r(p) = \sum_{j=1}^{p} b_r(j) \quad \text{for all r = 1..p} \tag{A-1}$$

where $b_r(1)$…….$b_r(p)$ are the consequents of the input variables.

The weights for input variables are calculated by

$$r_r = (m_1^r \Lambda m_2^r \Lambda .....m_k^r) * P^r \tag{A-2}$$

where $m_1^r, m_2^r .....m_k^r$ denote the membership values through membership functions of the input values for the r^{th} rule . The occurrences probability is shown by P^r and Λ stands for Min operation. The final output y inferred from n implications is given as the average of all y_r with the weights r_r

$$y = \frac{\sum_{r=1}^{n} r_r \times y_r}{\sum_{r=1}^{n} r_r} \tag{A-3}$$

A New Disc Based Particle Swarm Optimization

Anupam Yadav and Kusum Deep

Department of Mathematics, Indian Institute of Technology Roorkee, Uttarakhand
anupuam@gmail.com,kusumdeep@iitr.ernet.in

Abstract. Particle swarm optimization (PSO) is preponderantly used to find solution for continuous optimization problems, has the advantage of being cheaper and quicker. This paper introduces a new Disc-based particle swarm optimization (DPSO) algorithm to solve the complex optimization problems. The availability of the introduced algorithms is validated statistically on several benchmark problems and also compared with the existing versions of PSO.

Keywords: Particle Swarm optimization, LXPSO CPSO QPSO.

1 Introduction

Since the conventional computing algorithms are not well capable to solve the real life problems because of having incomplete and noisy data and some multi dimensional problems. Soft computing prototypes mainly nature inspired computational techniques appears to be worthy replacement for the conventional methods. Based on simple behavior of natural species and events these methods are easy to implement and able to solve the complicated real life problems. In this article we are focused on Particle Swarm Optimization.

We have good number of sub branches and developed versions of Particle Swarm Optimization till date from its very first creation by Kennedy and Eberhart[5]. This article is inspired by set based PSO proposed by Neethling and Engelbrecht, 2006[7] and Veenhuis C.B., 2008[8]. The concept proposed in this paper redefines the definition of personal best and global best and the behavior of cluster (disc) of particles is observed instead of a single particle. This disc based PSO is successfully applied to six benchmark problems of the operations research. The results are compared with the existing versions of the PSO.

The paper is prepared as follows: Section 2 introduces the particle swarm optimization algorithm. The definition of the newly defined personal best and global best are proposed in the next section with working scheme of the disc based PSO, parameter selection, results, comparisons with other PSO and its statistical validations. Finally some conclusions are drawn with a future prospect.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 23–30.
springerlink.com

2 Particle Swarm Optimization

Particle swarm optimization is a nature inspired evolutionary search technique which is probabilistic in nature. It is inspired by the social behaviour of animals such as bird's flocking and fish schooling. It was jointly proposed by Kennedy and Eberhart, 1995[5]. It simply uses the learning, information sharing and position updating strategy and very simple to implement. For better understanding mathematically it can be defined as in the following para.

For a D-dimensional search space, the nth particle of the swarm at time step t is represented by D-dimensional vector $x_n^t = (x_{n_1}^t, x_{n_2}^t, ...x_{n_D}^t)^T$, the velocity of the particle at time step t is denoted by $V_n^t = (V_{n_1}^t, V_{n_2}^t, ...V_{n_D}^t)$, let the best visited position of the particle at time step t is $P_n^t = (P_{n_1}^t, P_{n_2}^t, ...P_{n_D}^t)$. Let $'g'$ is the index of the best particle in the swarm. The position of the particle and its velocity is being updated using the following equations:

$$V_{nd}^{t+1} = V_{nd}^t + c_1 r_1 (P_{nd}^t - x_{nd}^t) + c_2 r_2 (P_{gd}^t - x_{nd}^t) \tag{1}$$

$$x_{nd}^{t+1} = x_{nd}^t + V_{nd}^t \tag{2}$$

here$d = 1, 2, 3, ..., D$ represents the dimension and $n = 1, 2, 3..., S$ represents the particle index. c_1 and $c_2$vare the constants and and are random variables with unifromly distributed between 0 and 1. Equation (1) and (2) define the classical version of PSO. The basic PSO algorithm by Kenedy and Eberhart,1995 is given by:

Create and initialize a D-dimensional swarm S
For t =1 to the maximum iterations
For n =1 to S,
For d = 1 to D,
Apply the velocity update equation (1)
Update position using equation (2)
End for d ;
Compute fitness of updated position ;
If needed update the historical information aboutP_n and P_g
. End for n ;
Terminates if meets problem requirements;
End for t;

3 Disc Based PSO

In this section, the disc based PSO is described. This idea of PSO will espouse to solve the discrete as well as continuous problems. After the very first conception of PSO, it meets with many more exploitations and modifications till date. The initial idea of information sharing among the particles got more maturity in various ways. The algorithm proposed by Kennedy and Eberheart in 1995[5] has its own way to establish communication among the inhabitants. They gave a theme to decide personal best and global best of the population. In addition to this Shi and

Eberheart[10] incorporated the topology for picking the two bests. A part from this modification and developments are pushed in, the way of defining velocity and position update equations. Communication was stick to individual particles having the common theme of these editions of PSO; C.B. Veenhuis (2008)[8] interrupted this continued tradition of PSO and rendered a new idea of Set based PSO. To deal, set as a particle in search space is the basic cognition of set based PSO[9]. Inspiring from these movements of the PSO found in literature we have designed disc based PSO. The information sharing idea and the searching strategy of personal best and global best is redefined in a suitable manner. The description and definitions this PSO is intended in the following paragraph.

3.1 Theme of Disc Based PSO

The very first aim of the disc based PSO is to use discs of defined radius to search the point of interest instead of singleton particle. The wrapper theme of Disc PSO seems to concur with set based PSO but it has basic distinction between both at very fundamental level. The random search techniques are having dreadful behavior of being struck at local minima's. To avoid this or overcome the possibility of being struck at point of lesser interest, the algorithm is designed in a way to overcome these undesired possibilities. We need to define few terms essential for the next section.

3.2 Definitions

Definition 1. The fitness of the particle will be defined by the function $F : R^n \rightarrow R$. s.t. x(t+1) will have a better fitness value than x(t) if F(x(t+1))<F(x(t)) (In case of Minimization). n is the natural number showing the dimension of the search space.

Definition 2. Disc, here in after will be treated as a circular space with a centre having certain radius.

Definition 3. Personal best (pb), the best known particle in each disc.

Definition 4. Temporary global best (tgb), best among the personal bests.

Definition 5. Global Disc, disc centered at temporary global best.

Definition 6. Global best (gb), best known particle with in global disc.

3.3 Pseudo Code of Disc Based PSO

Generate initial population randomly.
At each point of the generated population regenerate a disc of radius r.
Choose N number of particles in each disc randomly.

Calculate the fitness value of each particle in each disc with fitness value of its center.
Choose the best particle in each disc and mark it as personal best (pb) of each disc.
Again choose the best particle among the and name it temporary global best (tgb).
Generate a disc of radius r centered at tgb and choose N number of particles in tgb.
Choose the best among the and name it global best.
Update the position of each particle using equations (3) and (4)
Iterate the algorithm unless it meets with stopping criteria.

$$V(t+1) = w * V(t) + c_1(x(t) - pb) + c_2(x(t) - tgb) + c_3(x(t) - gb) \tag{3}$$

$$x(t+1) = x(t) + V(t+1) \tag{4}$$

In each iteration, we will linearly decrease the value of r converging to zero. To avoid the chaotic behavior of velocity V and position x in each iteration, both are clinched by $[V_{max}, V_{min}]$ and $[x_{min}, x_{max}]$ actively to the problem specific range.

3.4 Selection of Parameters

The parameter selection has been done in a logical manner. The value of w is taken to be 0.9 the best found value in literature, the values of c_1, c_2 and c_3 are chosen smartly as c_1 represents the weight to the influence towards personal best (pb) as so on c_2 and c_3 represents the influence of tgb and gb. c_1 is decreased linearly from c_1 to zero while c_2 and c_3 are increased linearly from 0 to c_2 and c_3 respectively. The reason for choosing parameters in this fashion is to avoid striking at local minima. In initial iteration the value of c_1 will be greater than c_2 and c_3 this will influence the current particle to move towards local minima's forcibly, with the increase in iterations the influence of c_2 and c_3 term will dominate the influence of c_1 term. This will guide the current particle to move towards global minima. The reduction in the radius of discs at each iteration will reduce the number of function evaluations. The stopping criterion for the algorithm is dual in nature
1000 Iterations.
Error is less than .001.
Algorithm will stop whichever condition will meet first.

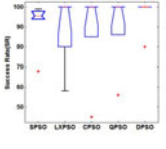


Fig. 1. Comparisons among existing versions of PSO with DPSO based on Success rate (SR)

4 Experiments

To test the availability of designed PSO experiments are done on the functions listed in Table 1. The used parameters are the standard ones $w = 0.9, c_1 = c_2 = c_3 = 0.5 + log(2)$ and N=30. Unit length is chosen for the radius of each disc and in each iteration the radius is decreased by 0.1 linearly. Validation of position and velocity of the particles are done according to $[x_{min}, x_{max}]$ interval. 100 independent runs and 1000 iterations are done with different population sizes by maintaining the same seed every time for each individual problem.

5 Performance and Evaluation Criteria

The efficiency of DPSO is tested against benchmark functions given in Table 1. The functions of Table 1 are considered here for 30 variables. To avoid attributing the optimization results to the choice particular initial population and to conduct fair comparisons, each test is performed 100 times. Starting from various randomly selected points in the hyper rectangular search domain. The five PSOs (SPSO[2], LXPSO[1], CPSO[3], QPSO[4] and DPSO) are implemented in MATLAB R2008a. A run during which the algorithm finds a solution satisfying $f_{min} - f_{opt} < 0.001$, when f_{min} the best solution found when an algorithm terminates and f_{opt} is known as a global minimum of the problem, is called a successful run. For each method and problem following are recorded:

Success Rate (SR) = ((No. of success of runs)/(total runs))*100

Average Error (AE) = Average error of $f_{min} - f_{opt}$ over 100 runs.

Standard Deviation (SD) = Standard deviations of error $f_{min} - f_{opt}$.

Success performance (SP) = AFE*(No. of successful runs)/(No. of total runs) [6]

Table 1. Benchmark Functions

Function	Range	Optimum
$f_1(x) = \sum_{i=0}^{n}(x_i^2 - 10cos(2\pi x_i) + 10)$	[2.56,5.12]	0
$f_2(x) = \frac{1}{4000}\sum_{i=0}^{n} x_i^2 - \prod cos(\frac{x_i}{\sqrt{(i+1)}})$	[300,600]	0
$f_3(x) = \sum_{i=0}^{n-1}(x_{i+1} - x_i^2)^2 + x_{i-1}^2$	[15,30]	0
$f_4(x) = -\sum_{i=1}^{n} x_i sin((\lvert x_i \rvert))$	[-500,500]	-418.9829*n
$f_5(x) = (\sum_{i=0}^{n-1}(i+1)x_i^4) + rand[0,1]$	[-1.28,1.28]	0
$f_6(x) = 20 + e - 20exp(-0.2\sqrt{((1/n))\sum_{i=1}^{n} x_i^2}) - exp((1/n))\sum_{i=1}^{n} cos(22\pi x_i))$	[-32,32]	0

6 Results and Discussion

The algorithm has been tested on six benchmark functions and also compared with the four existing algorithms LXPSO, CPSO, QPSO and SPSO. In order to study the performance of proposed DPSO we have obtained the results which are shown in Table 2 to Table 6. All the results are generated by programs of concern PSO coded on MATLAB. For the validation of comparison for each run we have generated the same pattern of random numbers as a initial population. The parameters used in the evaluation process are taken from the concern paper of each version of PSO. The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multilevel equations, it may be necessary.

Table 2. Success Rate

Function	SPSO	LXPSO	CPSO	QPSO	DPSO
f_1	95	100	100	100	100
f_2	96	100	100	100	100
f_3	68	58	45	56	80
f_4	94	80	85	86	100
f_5	99	100	100	100	100
f_6	98	100	100	100	100

Table 3. Average function evaluation of Successful runs(AFE)

Function	SPSO	LXPSO	CPSO	QPSO	DPSO
f_1	218	186	196	222	161
f_2	404	439	395	408	529
f_3	19403	18470	21254	17979	17902
f_4	919	815	900	711	591
f_5	617	587	507	605	551
f_6	922	992	925	1070	806

Table 4. Average Error

Function	SPSO	LXPSO	CPSO	QPSO	DPSO
f_1	0.0006420	0.000408	0.000338	0.000343	5.22E-05
f_2	0.0004135	0.000396	0.000487	0.000442	0.000115
f_3	0.0021674	0.000955	0.001072	0.000937	2.02E-05
f_5	0.0004128	0.000444	0.000466	0.000482	4.25E-05
f_6	0.0003901	0.000521	0.000546	0.000525	4.09E-05

Table 5. Standard Deviation

Function	SPSO	LXPSO	CPSO	QPSO	DPSO
f_1	0.001003	0.000309	0.000293	0.000302	4.03E-05
f_2	0.001016	0.000289	0.000308	0.000295	0.000205
f_3	0.002028	0.000103	0.000430	0.000126	0.000630
f_4	0.000405	0.000312	0.000456	0.000399	6.41E-05
f_5	0.000202	0.000274	0.000296	0.000289	4.10E-05
f_6	0.000311	0.000271	0.000299	0.000312	4.05E-05

Table 6. Success Performance

Function	SPSO	LXPSO	CPSO	QPSO	DPSO
f_1	218	286	196	222	161
f_2	404	439	395	408	529
f_3	19403	19170	21294	17979	18933
f_4	919	815	900	711	591
f_5	617	587	507	605	551
f_6	932	992	925	1095	806

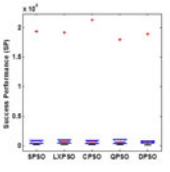


Fig. 2. Comparisons among existing versions of PSO with DPSO based on success performance (SP)

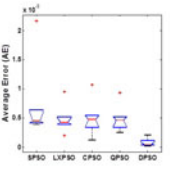


Fig. 3. Comparisons among existing versions of PSO with DPSO based on average error (AE)

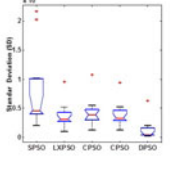


Fig. 4. Comparisons among existing versions of PSO with DPSO based on standard deviation (SD)

7 Conclusions

A disc based particle swarm optimization (DPSO) method has been proposed. Velocity update equation is defined in a new way. Based on proposed DPSO, different PSO variants, such as global version of the PSO as well as the local versions of the PSO can be extended to Disc versions. Since DPSO is proposed to solve complex multimodal problems and from the statistical validity of the algorithm, DPSO can be use to solve real life high complex problems. The most important the parameter selection for the DPSO has been done in a smarter manner that guaranty the desired accuracy of the algorithm. We have implemented experiments to compare the propose algorithm with existing versions of PSO. Experimental results have shown that the proposed algorithm is promising.

Acknowledgement. Author is thankful to the Council of Scientific and Industrial Research, Govt. of India for financial support also author is thankful to the Institute Computer Center, Indian Institute of Technology Roorkee for providing computational facilities.

References

1. Bansal, J.C., Veeramachaneni, K., Deep, K., Osadciw, L.: Information sharing strategy among particles in particle swarm optimization using laplacian operator. In: Swarm Intelligence Symposium (SIS), vol. 30, pp. 30–36. IEEE (2009)
2. Clerc, M., Kennedy, J.: Standard Particle Swarm Optimization (2006), http://www.particleswarm.info/StandardPSO2006.c (cited July 12, 2009)
3. Deep, K., Bansal, J.C.: A new chaotic particle swarm optimization algorithm. International Journal of Mathematical Modelling, Simulation and Applications (IJMMSA) 1(3), 249–263 (2009)
4. Deep, K., Bansal, J.C.: Hybridization of Particle Swarm Optimization with Quadratic Approximation. Opsearch 46(1), 3–24 (2009)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
6. Liang, J., Runarsson, T., Mezura-Montes, E., Clerc, M., Sugnathan, P., Coello, Deb, K.: Problems definitions and evaluation criterial for CEC 2006. Special Session on Constrained Real Parameter Optimization Technical Report (2006)
7. Neethling, M., Engelbrecht, A.P.: Determining RNA Secondary Structure using Set-based Particle Swarm Optimization. In: Evolutionary Computation, CEC 2006, pp. 1670–1677. IEEE (2006), doi:10.1109/CEC.2006.1688509
8. Veenhuis, C.B.: A Set-Based Particle Swarm Optimization Method. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 971–980. Springer, Heidelberg (2008)
9. Wei-Neng, C., Jun, Z., Chung, H.S.H., Wen-Liang, Z., Wei-Gang, W., Yu-Hui, S.: A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems. IEEE Transactions on Evolutionary Computation 14(2), 278–300 (2010), doi:10.1109/TEVC.2009.2030331
10. Yu-hui, S.: Evolutionary Computation. IEEE Transactions 14(2), 278–300 (2010), doi: 10.1109/TEVC.2009.2030331

Application of Globally Adaptive Inertia Weight PSO to Lennard-Jones Problem

Kusum Deep and Madhuri

Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee – 247667, Uttarakhand, India
kusumfma@iitr.ernet.in, madhuriiitr@gmail.com

Abstract. Particle Swarm optimization (PSO), one of the most popular natural computing paradigms, usually requires a large number of fitness evaluations to reach the global optimal solution when applied to some real world problems. Recently a new inertia weight strategy namely Globally Adaptive Inertia Weight (GAIW) has been introduced in PSO which deals with this problem to some extent. In this paper the efficacy of GAIW in PSO has been tested on a computationally challenging global optimization problem namely Lennard-Jones problem. The experiments have been performed for small clusters of 3 to 12 atoms. The computational results thus obtained have been compared with three previously existing inertia weight versions of PSO. The results demonstrate that the use of GAIW in PSO significantly reduces the required number of fitness evaluations and hence the computational time while giving higher success rate.

Keywords: inertia weight, particle swarm optimization, lennard-jones problem, atomic clusters, global optimization.

1 Introduction

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique [1, 2]. It has been applied to solve a wide variety of global optimization problems occurring in science, engineering and industry. It is very popular because of its simple mathematical operations, a small number of control parameters, quick convergence and ease of implementation.

Inertia weight is one of the few parameters of PSO that can be used to control its performance. PSO usually takes a large number of functional evaluations for arriving at an acceptable solution. To deal with this difficulty and to make PSO more effective an adaptive inertia weight strategy called Globally Adaptive Inertia Weight (GAIW) has been introduced recently [3]. The authors have tested its performance on six benchmark global optimization problems and have reported that it provides better convergence rate with less computational effort and high success rate when compared to some other inertia weight variants of PSO. In order to test the efficacy of this inertia weight strategy in PSO to solve a variety of problems, a

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 31–38.
springerlink.com

very highly computationally challenging problem namely Lennard-Jones problem that has been used as a benchmark global optimization problem in literature, is taken in the present paper. Ten instances of this problem for clusters of 3 to 10 atoms have been considered. On these problem instances performance of the GAIW PSO has been compared to three other inertia weight variants of PSO namely fixed inertia weight (FIW), linearly decreasing inertia weight (LDIW) and non-linearly decreasing inertia weight (NDIW) that have been taken from literature [4, 5].

The remainder of the paper is organized as follows: Section 2 gives an overview of the basic PSO and section 3 presents the four inertia weight strategies that have been compared in this paper. Section 4 gives a brief introduction of the Lennard-Jones problem. The experimental results and discussions are presented in section 5. The paper is concluded with a direction for future work in section 6.

2 Basic PSO

PSO applies the concept of social intelligence of unintelligent species for solving real world problems. In PSO terminology population is called swarm and the individuals in the population are called particles. Each particle represents a position in the search space and has a velocity vector associated with it. Each of them maintains a memory that helps it in keeping track of the best position it has achieved so far, called the personal best (or the pbest) position of the particle, and the best position achieved so far by any particle in the swarm which is called the global best position (or the gbest position) of the swarm. The velocity and position of each particle at each iteration is updated according the following velocity and position update equations:

$$v_{id} = \underbrace{wv_{id}}_{\text{Inertia component}} + \underbrace{c_1 r_1 (p_{id} - x_{id})}_{\text{Cognitive component}} + \underbrace{c_2 r_2 (p_{gd} - x_{id})}_{\text{Social component}} \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

This process is then iterated until some predefined stopping criterion is satisfied. Here w is the inertia weight, $X_i=(x_{i1},x_{i2},\ldots x_{id})$ represents the position of the i^{th} particle in a d-dimensional search space, $P_{besti}=(p_{i1},p_{i2},\ldots p_{id})$ is i^{th} particle's pbest position, $P_{gbest}=(p_{g1},p_{g2}\ldots p_{gd})$ is the gbest position and $V_i=(v_{i1},v_{i2},\ldots v_{id})$ is the velocity of i^{th} particle. The acceleration coefficients c_1 and c_2 control how far a particle can move in a single iteration. The coefficients r_1 and r_2 are the uniformly generated random numbers in the range [0, 1].

3 Globally Adaptive Inertia Weight PSO

Inertia weight controls the performance of PSO by balancing exploration exploitation trade-off. Exploration is the ability of an algorithm to explore different

regions of the search space in order to locate a good optimum. Exploitation, on the other hand, is the ability to concentrate the search around a promising area in order to refine a candidate solution as defined by [6]. Since the introduction of inertia weight in PSO, a lot of research has been devoted to find optimal or the standard value of inertia weight. But results show that its value is problem dependent and no standard value has yet been found.

Most of the existing methods use some deterministic approach for dynamic adjustment of inertia weight. K. Deep et al. [3] have proposed a non-deterministic strategy called globally adaptive inertia weight (GAIW). It is based on the improvement in the best fitness of the particles as the search process progresses. In this method at any iteration if the swarm finds a better global best position, the inertia weight is increased so as to increase exploration in the current fruitful directions otherwise it is made zero so as to avoid searching potentially unfavourable inertial directions. In this process inertia weight may take very large values that may result in the explosion of particle velocities, so the velocities are clamped in the range $[-V_{dmax}, V_{dmax}]$ i.e.,

$$\left.\begin{array}{ll} v_{id} = -V_{d\max} & if \quad v_{id} < -V_{d\max} \\ v_{id} = V_{d\max} & if \quad v_{id} > V_{d\max} \end{array}\right\} \tag{3}$$

Also it was observed experimentally that the inertia weight sometimes becomes zero and remains zero for many successive iterations which means that the swarm sometimes stagnates at suboptimal solution and the social and cognitive components are not capable of escaping it. So some inertia is provided to the swarm whenever it remains zero for M successive iterations. Hence the inertia weight update equation for GAIW PSO is given by

$$\left.\begin{array}{l} w(t+1) = 0.9 \qquad \text{if } t = 0 \\ \text{and for } t > 0 \text{ we have} \\ w(t+1) = f(t\text{-}1) - f(t) \\ w = w_{start} - (w_{start} - w_{end}) * t / t_{\max}; \quad \text{if } w = 0 \text{ for M} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{successive iterations} \end{array}\right\} \tag{4}$$

Where w_{start} is the initial value and w_{end} is the final value of the inertia weight, and t is the current iteration (generation) of the algorithm while the $t_{\max}$ is the maximum number of iterations (generations) specified by the user.

4 Lennard-Jones Problem

The Lennard-Jones problem (LJP) assumes that the potential energy of a cluster of atoms is given by the sum of pairwise interactions between atoms, with these interactions being Vander Waals forces given by the Lennard-Jones 6-12 potential. The problem consists of determining the positions $t_1, t_2, ..., t_n \in R^3$ of an n atom cluster in such a way that the Lennard-Jones potential

$$V = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(r_{ij}^{-12} - 2r_{ij}^{-6} \right) \tag{5}$$

generated by atomic interactions is minimized, where r_{ij} is the Euclidean distance between the points t_i and t_j. Now since each point corresponds to Cartesian coordinates in each of the *x,y* and *z* directions so the actual number of variables is three times the number of atoms in the given cluster. The problem is then to find the positions of each atom of the cluster that corresponds to the global minimum value of *V*, equation (5). From the viewpoint of mathematical optimization, it is a difficult global optimization problem which does not yield easily either to discrete or to continuous optimization methods [7, 8].

5 Experiments and Discussions

In this section the performance of GAIW PSO is compared with three other variants of PSO namely fixed inertia weight (FIW), linearly decreasing inertia weight (LDIW) and non-linearly decreasing inertia weight (NDIW) on 10 instances of LJP. All the PSOs are implemented in C and experiments are carried out on a Xeon, 3.4 GHz machine under LINUX operating system. GAIW is determined by equation (4). The equations (6) and (7) are used to determine LDIW [4] and NDIW [5] respectively.

$$w = w_{start} - (w_{start} - w_{end}) * t / t_{max} \tag{6}$$

$$w = (w_{start} - w_{end}) * tan\left(\frac{7}{8} * \left(1 - \left(\frac{t}{t_{\max}} \right)^{k} \right) \right) + w_{end} \tag{7}$$

Where w_{start}, w_{end}, $t_{\max}$ and t have the same meanings as in equations (6), *tan()* is the trigonometric tangent function, and k is the control variable which can control the smoothness of the curve that reflects the relationship between the w and t.

5.1 Parameter Settings and Performance Evaluation Criteria

Here we use c_1=c_2=2, t_{max}=20000. Some parameters are given in Table 1. Maximum allowable velocity in each dimension is taken to be

$$V_{dmax} = (x_{dmax} - x_{dmin}) * 0.25$$

Where x_{dmax} and x_{dmin} are the upper and lower bounds for particles' positions in d^{th} dimension of the search space. The simulation of an algorithm is stopped as soon as either of the following two conditions is satisfied: (i) reaching the maximum number of iterations, (ii) getting the minimum error ε (i.e., getting a solution within the tolerance limit ε=0.01). For FIW PSO, w=0.68 is set. For all other algorithms w_{start}=0.9 and w_{end}=0.4 are set. Also for NDIW PSO k=0.6 (as recommended in [5]) is taken. For GAIW M=25 is taken for all problem instances.

Each test is performed 100 times, starting from various randomly selected points in the search space. All the results have been averaged over the successful runs out of 100. A run is considered a success if the algorithm finds a solution satisfying $| f_{opt} - f_{min} | < \varepsilon$, where f_{min} is the best solution found when an algorithm terminates and f_{opt} is the known global optimum of the problem. For each algorithm and for each problem the following are recorded:

a) Average number of function evaluations of successful runs (AFE).

b) Success Rate= $\frac{(\#\text{ of successful runs})}{\text{total runs}} \times 100$

c) Average Execution Time (AET) of successful runs

d) Success Performance (SP) = $\frac{(\text{AFE})}{\#\text{ of successful runs}} \times (\#\text{ of total runs})$ [9].

5.2 Results and Discussions

The results are presented in Tables 2 to 5 and Figures 1 to 4. In the tables the boxes with a hyphen (-) correspond to no success in any of the 100 runs and also the results are given in the same order as in Table 1 i.e., 1st row corresponds to 3 atoms and so on. To carry out statistical analysis box plots have been used. For drawing boxplots the problems with no success are assigned maximum number of function evaluations, maximum time and the maximum success performance. In the box plots each box represents the performance of an algorithm with respect to one criterion for all the problem instances taken together.

The goal of the analysis is to observe if the proposed strategy shows an improvement over the existing ones or not. Average number of function evaluations (AFE) is a measure of computational cost of the algorithm. It is clear from Figure 1 and Table 2 that GAIW PSO significantly reduces the AFE and the order of performance of PSOs based on it is: GAIW > FIW > NDIW > LDIW.

Note: A > B implies that algorithm A performs better than algorithm B from a particular point of view.

The average execution time (AET) of the successful runs is presented in Table 3 and Figure 2. Clearly GAIW is again the best performer. The order of PSOs based on the AET is: GAIW > FIW > NDIW > LDIW. Also it can be seen that by using the proposed strategy the AET decreases significantly.

Table 4 and Figure 3 compares success rate (SR) of all versions of PSOs considered here. SR is directly proportional to the reliability of the method. Clearly the performance order of PSOs based on SR is: GAIW > FIW > LDIW > NDIW.

In order to observe the consolidated effect on SR and AFE performance, a comparison among all four versions is made on the basis of success performance (SP) also. Table 5 and Figure 4 present this information. From SP point of view the performance order is: GAIW > FIW > NDIW > LDIW.

An interesting point that can be observed from the tables is that even in the cases when other algorithms could not find a solution, GAIW performs well, except one case for 11 atoms. On the basis of above analysis, GAIW performs the best among all five versions of PSO considered here.

Table 1. Parameters

No. of atoms	Swarm Size	Search Space
3	20	$[-0.52, 0.45]^{9}$
4	25	$[-0.52, 0.62]^{12}$
5	30	$[-0.75, 0.75]^{15}$
6	40	$[-0.75, 0.75]^{18}$
7	45	$[-0.96, 0.87]^{21}$
8	50	$[-0.9, 1.022]^{24}$
9	55	$[-2, 2]^{27}$
10	60	$[-2, 2]^{30}$
11	70	$[-1.1,1.2]^{33}$
12	75	$[-1.1,1.2]^{36}$

Table 2. Average Function Evaluations

FIW	LDIW	NLIW	GAIW
323	815	994	298
2875	81669	39933	2488
17254	183380	90168	30830
25860	247040	_	38340
252915	342315	_	171461
410413	485683	338100	105131
335679	594880	400868	341924
338616	672600	_	303270
541240	_	_	_
_	_	_	980850

Table 3. Average Execution Time

FIW	LDIW	NLIW	GAIW
0.001	0.003	0.003	0.001
0.015	0.436	0.21	0.013
0.134	1.498	0.698	0.236
0.275	2.788	_	0.412
3.712	5.118	_	2.406
7.856	9.213	6.26	1.889
7.608	13.307	9.064	7.746
9.169	18.376	_	8.254
18.465	_	_	_
_	_	_	37.898

Table 4. Success Rate

FIW	LDIW	NLIW	GAIW
100	98	95	99
64	55	45	71
68	50	50	85
4	1	0	4
3	1	0	12
4	3	3	8
8	1	2	10
5	1	0	2
1	0	0	0
0	0	0	1

Table 5. Success Performance

FIW	LDIW	NLIW	GAIW
323	831.63	1046.32	301.01
4492.19	148489.1	88740	3504.23
25373.53	366760	180336	36270.59
646500	24704000	_	958500
8430500	34231500	_	1428842
10260325	16189433	11270000	1314138
4195988	59488000	20043400	3419240
6772320	67260000	_	15163500
54124000	_	_	_
_	_	_	98085000

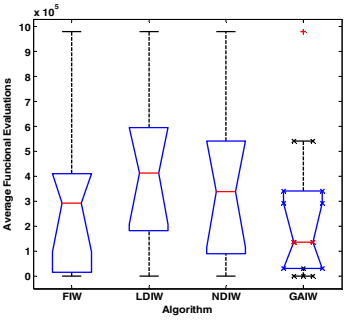


Fig. 1. Average Function Evaluations

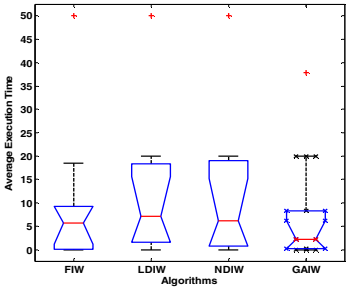


Fig. 2. Average Execution Time

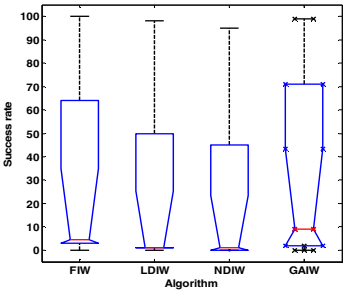


Fig. 3. Success Rate

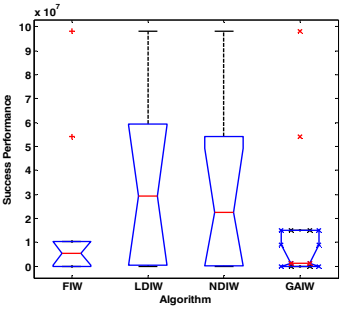


Fig. 4. Success Performance

6 Conclusion

In this paper a recently developed inertia weight variant of PSO, called globally adaptive inertia weight (GAIW) PSO has been applied to Lennard-Jones problem (LJP), in order to test its efficacy in solving hard problems. Its performance has been compared with fixed inertia weight, linearly decreasing inertia weight and non-linearly decreasing inertia weight variants of PSO. Ten instances of the LJP for clusters of 3 to 12 atoms have been considered for this purpose. The experimental results show that GAIW solves most of the problems studied here in less time with less computational effort and high success rate. Thus GAIW PSO performs better than all other variants of PSO considered here. The future work in this direction may be to apply GAIW PSO to various other optimization problems occurring in real life.

Acknowledgment. The second author, Madhuri, acknowledges Council of Scientific and Industrial Research, New Delhi, India, for providing financial support for this work.

References

[1] Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE Conf. on Neural Networks (Perth), pp. 1942–1948 (1995)
[2] Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publishers (2001)
[3] Deep, K., Arya, M., Bansal, J.C.: A non-deterministic adaptive inertia weight in PSO. In: Proc. Genetic and Evolutionary Computation Conference (GECCO 2011), pp. 1155–1162. ACM (2011)
[4] Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proc. IEEE Conf. on Evolutionary Computation., IEEE Press, Piscataway (1998)
[5] Li, L., Xue, B., Niu, B., Tan, L., Wang, J.: A Novel Particle Swarm Optimization with Non-Linear Inertia Weight Based on Tangent Function. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS (LNAI), vol. 5755, pp. 785–793. Springer, Heidelberg (2009)
[6] Engelbercht, A.P.: Fundamentals of computational swarm intelligence. John Wiley & Sons (2005)
[7] Bernard, R.B., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., Karplus, M.: CHARMM: A program for macromolecular energy, minimization, and dynamics calculation. Journal of Computational Chemistry 4, 187–271 (1983)
[8] Colvin, M., Judson, R., Meza, J.: A genetic algorithm approach to molecular structure determination. Paper Presented at International Conference on Industrial and Applied Mathematics, Washington, DC (1991)
[9] Liang, J., Runarsson, T., Mezura-Montes, E., et al.: Problem definitions and evaluation criteria for the CEC 2006. Special Session on Constrained Real-Parameter Optimization, Technical Report (2006)

Serial DPGA vs. Parallel Multithreaded DPGA: Threading Aspects

A.J. Umbarkar[1] and M.S. Joshi[2]

[1] Department of Information Technology, Walchand College of Engineering, Sangli, MS, India
anantumbarkar@rediffmail.com
[2] Department of Computer Engineering, Government College of Engineering & Research, Awasari (Khurd), Dist - Pune, MS, India
madhuris.joshi@gmail.com

Abstract. The multiple main populations, reserve populations and subpopulations concepts of a Genetic Algorithms (GAs) offers the advantage of diversity. However, as the population evolves, the GA loses its diversity. As the population converges, it begins to lose its diversity and cannot avoid the local optima problem. This problem is known as *Premature Convergence* for Parallel GAs (PGA) too. The paper compares the Binary encoded Simple GA (SGA), Binary encoded Serial/Sequential Dual Population Genetic Algorithm (SDPGA) and Binary encoded Multithreaded Parallel DPGA (MPDPGA) performances for function optimization on multicore system. The Dual Population Genetic Algorithm (DPGA) is an evolutionary algorithm that uses an extra population called the reserve population to provide additional diversity to the main population through crossbreeding. The experimental results on unimodal and multimodal classes of test problem shows the MPDPGA outperforms over SGA and SDPGA. The performance of MPDPGA with DPGA[1] is better in terms of accuracy, number of generations and execution time on multicore system. The performance of MPDPGA with DPGA-ED[1] is better for Rosenbrock and Schwefel whereas worse for Ackley and Griewangk.

Keywords: Serial/Sequential Dual-Population GA, Parallel Dual-Population GA, Function Optimization, Premature Convergence, Diversity.

1 Introduction

The GAs are one of the important soft computing tools which are used in engineering and optimization. GAs are efficient search or optimization algorithms based on principles of natural selection and population genetics.

[1] DPGA and DPGA-ED are work of T. Park [12].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 39–50.
springerlink.com

GAs are robust, which are generally able to find optimal solutions in reasonable amount of time, but as they are applied to problems which are NP hard and having big dimensions, then there is an increase in the searching time to find optimal solutions. In order to minimize the searching time multiple efforts have been made to make GAs faster and one of the most promising choices is to make efficient parallel design of GAs.

Parallel GAs (PGAs) have proved to be useful in number of industrial and commercial applications. Many real-life problems may need days or weeks of computing time to solve on serial machines. Although the intrinsic time complexity of a problem cannot be lowered by using a finite amount of computing resources in parallel, parallelism often allows reducing time to reasonable levels. This can be very important in an industrial or commercial setting where the time to find the solution is critical for decision making. Furthermore, the new models offered by structured populations often allow better exploration of alternative solutions of the search space.

The multicore processor technology is getting cheaper and common; one cannot ignore its importance anymore. To enhance GAs performance, GAs can take advantage of parallel computing environments based on their well paralleled nature [1]. PGAs provide abilities of GAs and measurable speedup in computing against conventional GAs for a slight increase in their complexity.

Nowadays there is a need of performance improvement in each application so as to minimize the time required for execution. The parallel computing is a broad area under which highly computational problems are solved. The problems are divided in to subparts and then processed in parallel fashion. Though the GAs algorithmic development is at extreme level, but there is full scope for making it parallel on multicore computer system with parallel programming languages [2].

The gains from running GA in the parallel are as follows- run time saving, speedup of finding solutions, search of greater problem search space, following various diversified search paths, maximal utilization of computing machinery, finding alternative solutions to the same problem, higher efficiency than sequential GAs, in many cases, easy cooperation with other (EA and traditional) search procedures, speedup due to the use of multiple CPUs (faster search) and lastly efficient use of Multicore architecture for NP hard problems/Optimization problems.

The rest of the paper is organized as follows. Section 2 explains the literature survey of DPGA. Section 3 describes the fitness function and SDPGA and MPDPGA algorithms. Section 4 reports the experimental results for unimodal and multimodal optimization problems and compares SDPGA and MPDPGA with other genetic algorithms. Finally, section V provides conclusions.

2 Related Works

One of the very important facts given by Cant´u-Paz (2000) [3] proposes many configuration options of topologies, migration rates, number and size of demes.

Many researchers proposed PGAs from year 2000 to 2003, which are based on distributed systems. Sefrioui and P´eriaux (2000) [4] proposed Hierarchical Genetic Algorithms (HGAs) with multi-layered hierarchical topology and multiple models for optimization problems. Rivera (2001) [5] investigated how to implement PGAs with obtaining quality of solutions efficiently. Rivera reviewed the state-of-the-art in PGAs, parallelization strategies, emerging implementations and relevant results. Alba and Troya (2001) [6] proposed a common framework for studying PGAs. Alba and Troya (2002) [7] brought some uniformity in comparison, and knowledge exchange among the traditionally opposite kind of serial and parallel GAs. They analyzed the properties of steady-state, generational and cellular GAs and extended the idea to distributed model ring of the GA islands. Giacobini (2003) [8] proposed a theoretical study of the selection pressure in asynchronous cellular (fine-grained) evolutionary algorithms (cEAs). Xiao and Amstrong (2003) [9] proposed a model of parallel evolutionary algorithms (EAs) called a specialized island model (SIM). Gagn´e (2003) [10] says that the classic master slave distribution model was superior to the island-model when exploiting Beowulfs and networks of heterogeneous workstations. They identified the key features of a good computing system for evolutionary computation- transparency, robustness and adaptivity.

Till then, the PGAs work was concentrated on multiple demes on distributed system- Clusters**,** Grids and MPPs with common fitness function. After year 2003, multiple demes concept with one more objectives for diversity in search space. The research based on diversity in GAs, as a second objective are as follows - Yang (2003) proposed Primal-Dual Genetic Algorithm (PDGA) which adopted complimentary and dominance mechanism for diversity. PDGA solved complex dynamic problems efficiently than traditional SGA. [11], [12]. Park (2006) proposed DPGA improved version binary DPGA which shows that the performance of PGA can be improved by having the reserved population with different objective function than of main population. This DPGA algorithm is also inspired by complimentary and dominance mechanism for diversity. The DPGA algorithm of T. Park outperforms on stationary optimization problems [13]. Park (2007) has proposed DPGA-ED improved version of binary representation based DPGA. The reserve population DPGA-ED is different from DPGA by evolving itself. The DPGA-ED algorithm outperforms on nonstationary optimization Problems. [14]. Park (2008) has proposed DPGA2 for Nonstationary optimization Problems. The improvement in DPGA2 was two reserve populations which provide controlled inflow information to main population with survival selection. DPGA2 shows the performance without relying on prior knowledge of nonstationary optimization problems. [15]. Junhua (2008) proposed DSGA called Dual Species GA. In this algorithm one subpopulation works on local exploitation and other subpopulation work for global exploration with migration between them for optimal solution. DSGA shows better performance than SGA and 2PMGA [16]. The concept of reserve population with objectives as diversity in GAs on distributed systems was researched in previous works. A pretty positive attempt to implement a similar concept of DPGA on multicore technology with threading has been tried in this research paper.

3 Algorithms

DPGA has two distinct populations called *main population* M_p and *reserve population* R_p. Main population is having same role as that of the population of an ordinary GA. Main population aim to evolve to find a good solution with a high fitness value. The reserve population provides additional diversity to the main population. The reserve population is employed as a reservoir for keeping chromosomes which are different from those of the main population. DPGA uses migration as a means of information exchange. DPGA uses *crossbreeding* as a means of information exchange. The two different fitness functions for M_p and R_p of DPGA are:

Fitness Functions for Main Population [17]: The fitness function for main population is unimodal or multimodal test functions to check the large scale global optimization. The unimodal test functions- Rosenbrock (F1), Ackley (F2) and multimodal test functions- Griewangk (F3) and Schwefel (F4) are used in experimental work.

Rosenbrock Test Function $f_M(x)$: Rosenbrock's valley is a classic optimization problem, also known as *banana function*. The global optimum lies inside a long, narrow, parabolic shaped flat valley. Equation (1) is Rosenbrock Test Function.

$$f_M(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \tag{1}$$

Test area -2.048<= x_i <= 2.048; where i=1,…, N; where N: Dimension.

Its global minimum equal $f(x) = 0$ is obtainable for x_i, where $i = 1,\ldots,N$.

Fitness Function for Reserve Population $f_R(x)$: [18]

$$f_R(x) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{l} \left| m_{i,k} - x_k \right| \tag{2}$$

x_k : k^{th} gene of chromosome x from R_p.
$m_{i,k}$: k^{th} gene of chromosome m_i from M_p.
n: Population Size of M_p and R_p
l : Chromosome length.

The equation (2) compare the k^{th} gene chromosome from Mp with k^{th} gene of chromosome x from R_p and assigns the fitness for reserve populating.

In DPGA, M_p and R_p are randomly generated populations. Each individual of M_p and R_p are evaluated by their fitness functions. DPGA generates offspring by both *inbreeding* and *crossbreeding*. The M_p and R_p evolves by inbreeding process separately. The crossbreeding process migrate information from R_p to M_p. The Inbred Offsprings process is identical to the standard GA for generating offspring. The procedure for generating Crossbred Offspring selects the parents from different populations, one from the M_p and the other from the R_p.

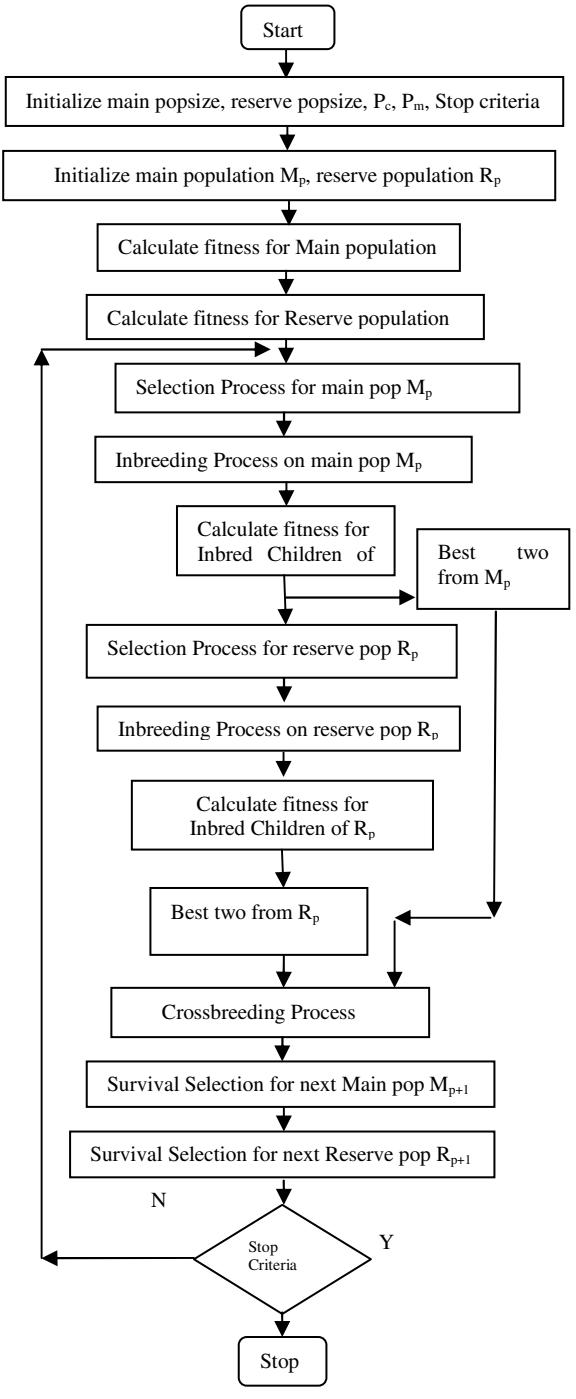


Fig. 1. Schematic diagram of SDPGA

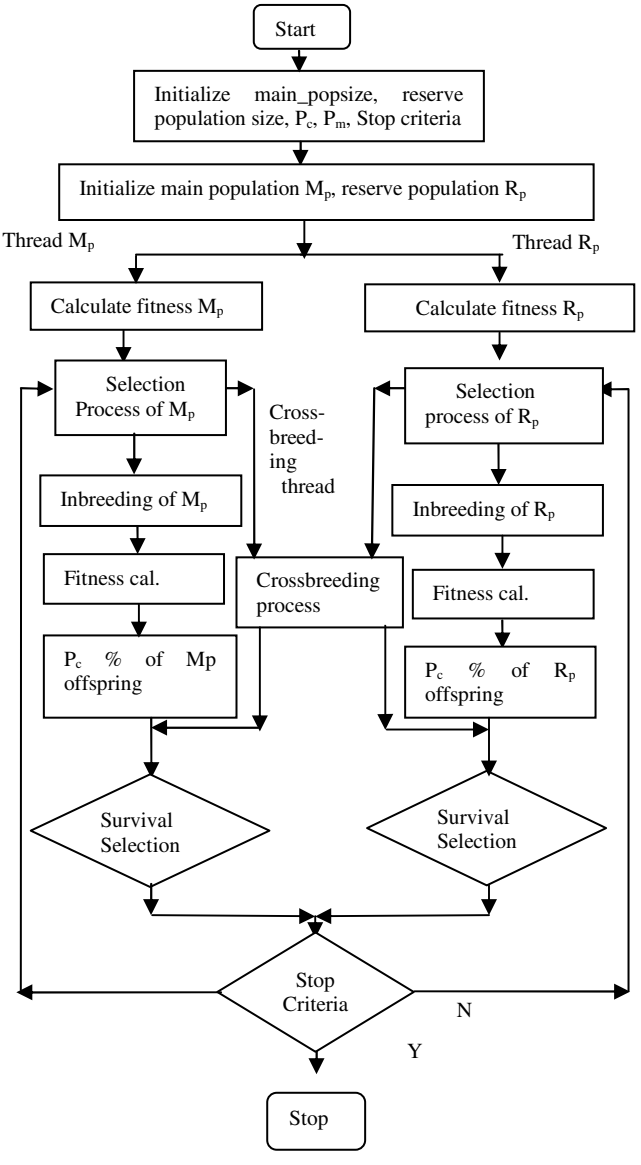


Fig. 2. Schematic diagram of MPDPGA

The implementation of DPGA is done with two kinds namely sequential/serial and parallel called as SDPGA and MPDPGA respectively. Fig. 1 shows the Schematic diagram of the SDPGA and Fig. 2 shows the Schematic diagram of the MPDPGA.

3.1 SDPGA

In SDPGA the main population and reserve population along with migration are evolving sequentially. SDPGA is single threaded DPGA, so the exploration, exploitation, migration *etc*. are sequential.

3.2 MPDPGA

The concept of thread increases the responsiveness, resource shearing, economy in utilization of multiprocessor architecture. The multithreaded parallel DPGA is implemented using three threads. In MPDPGA the main population, reserve population and migration have separate threads for evolving in parallel. Third thread is responsible for crossbreeding between M_p and R_p.

4 Experimental Results

The SGA, SDPGA and MPDPGA algorithms are implemented in Java language on a Personal Computer-Intel Dual Core CPU 2.80 GHz with 1 GB of memory and 150 GB hard disk. The profiling of java programs is done using Java Visual VM. It gives the analysis of CPU usage, Garbage Collection (GC) activity, heap and permanent generation memory, number of loaded classes and running threads etc.

Thread level profiling gives the clear impact threaded implantation of algorithms on CPU usage, heap use, loaded class and threads running for algorithms. The tracking of thread activity and uncovered inefficient patterns, like blocked event dispatch threads or unused worker threads will help to optimize algorithm. The memory allocation, objectswise, thread dump also gives a very clear view about the memory consumption for algorithm implementation. The Fig. 3 and 4 shows that, the SDPGA uses more CPU (~90 %) with single thread of execution, the MPDPGA uses CPU (~55 %) with multithreaded execution. This reduction in CPU usage is because of threaded implementation of DPGA by 35%, hence the utilization of CPU is optimal in case of MPDPGA over SDPGA. It is observed that, the usage of heap for SPDPGA is 2.28 Gigabytes and for MPDPGA is 2.93 Gigabytes out of maximum heap size of 64 Gigabytes. The MPDPGA usage of heap in percentage is 78% more than the heap utilization of SPDPGA. The MPDPGA usage is 0.65 Gigabytes more heap than SPDPGA, which means 4.57 % of maximum heap size.

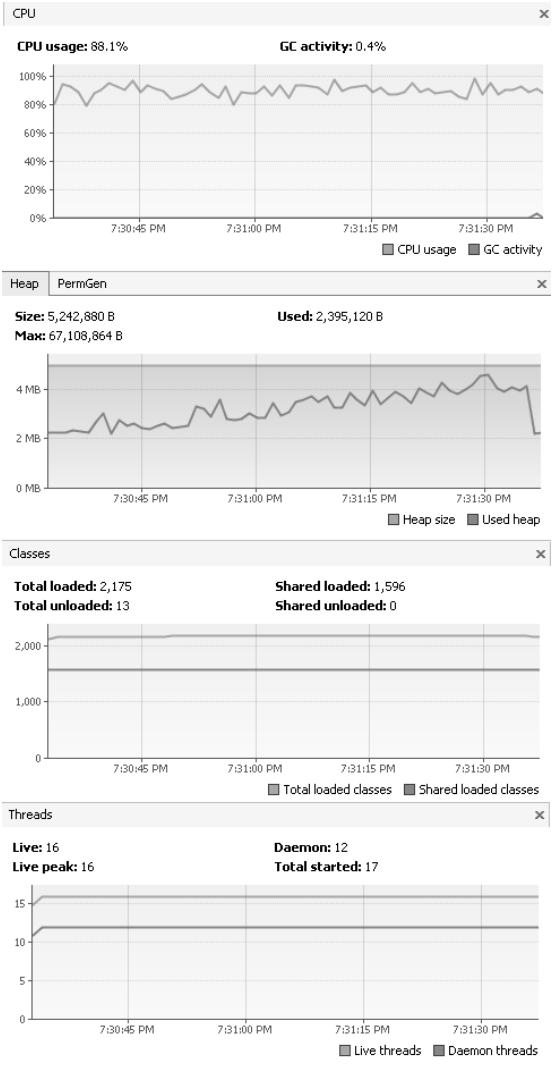


Fig. 3. CPU usage, Heap usage, Classes loaeded and live threads of SDPGA

It is observed that, MPDPGA uses heap more symmetric and systematic in comparision with SDPGA. Whereas the classes loaded in Java for SDPGA and MPDPGA are same. The thread plot shows that MPDPGA uses two more extra threads than SDPGA, which are for crossbreeding and reserve population.

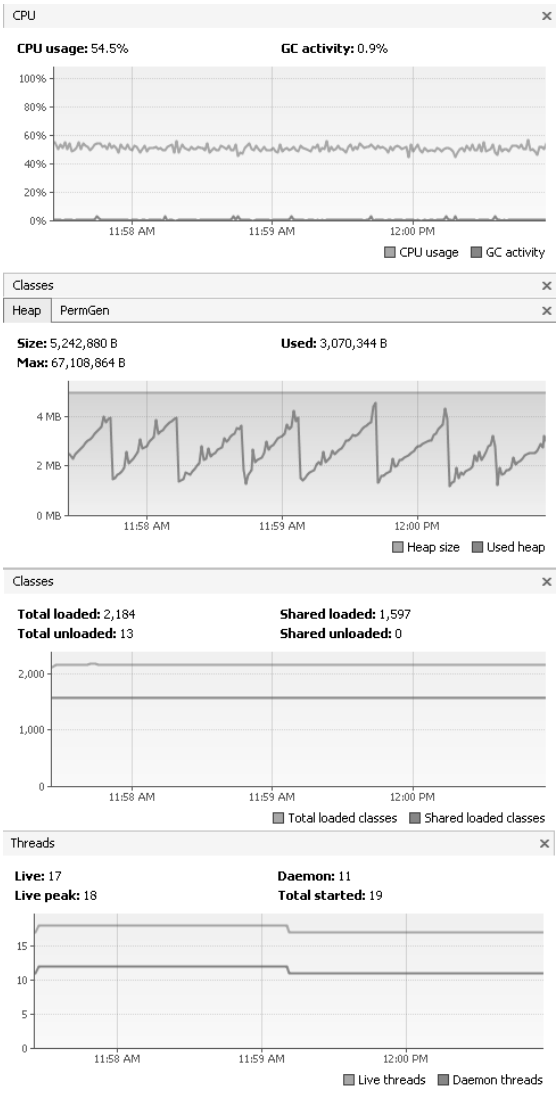


Fig. 4. CPU usage, Heap usage, Classes loaeded and live threads of MPDPGA

Table 1 gives the Parameter setting for SGA, SDPGA and MPDPGA. Table 2 gives comparison of MPDPGA, SDSGA, SGA and DPGA-ED[1] of T. Park. The results show that the MPDPGA is performing better than SDPGA, SGA and DPGA-ED1. It is seen that MPDPGA requires less number of generation and computational time on experimented hardware. The MPDPGA algorithm is implemented using Java threading, the utilization of multicore system to solve optimization problems increases.

Table 1. Parameter setting for SGA, SDPGA and MPDPGA

Initial Parameters	SGA	SDPGA	MPDPGA
Main Population size (M_p)	100	100	100
Reserve Population size (R_p)	-	100	100
Crossover Method	K-point	K-point	K-point
Crossover point K	5	5	5
Mutation Method	Flip-bit	Flip-bit	Flip-bit
Crossover Probability (Pc)	0.7	0.7	0.7
Mutation Probability (Pm)	0.03	0.03	0.03
Dimension (D)	5	5	5

Table 2. Comparison of Best Solution Found of SGA, SDPGA and MPDPGA with DPGA-ED[1] of T Park [18]

Test function (F)	Algorithm	Best Solution Found	No. of Generations
F1	MPDPGA	0.00612658	1002
	SDPGA	0.48340664	33,139
	SGA	0.94597337	1,54,462
	DPGA-ED[1]	0.9921	500000
F2	MPDPGA	0.49586544	175
	SDPGA	0.49528813	22,035
	SGA	0.49930387	60,676
	DPGA-ED[1]	0	16081
F3	MPDPGA	0.15830217	150
	SDPGA	0.12492430	3,96,121
	SGA	0.32442724	4,01,512
	DPGA-ED[1]	0.0680	483255
F4	MPDPGA	0.004589554	2053
	SDPGA	0.00232797	1,23,549
	SGA	-0.07381499	1,57,294
	DPGA-ED[1]	0.0014	89007

Table 3 is the comparison of MPDPGA, SDPGA and SGA with DPGA[1], DPGA-ED[1] of T. Park. Table 2 and 3 are comparative performance measures with algorithms of [18]. The differences MPDPGA, SDPGA, SGA with DPGA[1], DPGA-ED[1] are given with better, worse or not significant.

- A plus sign (+): the performance of algorithm is better than the one of the corresponding algorithm.
- A minus sign (-): the performance of algorithm is worse than the one of the corresponding algorithm.
- An approximate sign (≈): not statistically significant differences.

Table 3. Comparison of Best Solution Found of SGA, SDPGA and MPDPGA with DPGA-ED[1] of T Park [18]

Test function (F)	Algorithm	MPDPGA Versus	DPGA[1] Versus	DPGA-ED[1] Versus
F1	MPDPGA	---	+	+
	SDPGA	-	+	+
	SGA	-	+	•
F2	MPDPGA	--	•	-
	SDPGA	•	•	-
	SGA	•	-	-
F3	MPDPGA	---		-
	SDPGA	•	-	-
	SGA	+	-	-
F4	MPDPGA	---	+	•
	SDPGA	•	+	+
	SGA	+	+	-

5 Conclusion

In this research paper, the Binary encoded MPDPGA is implemented using two populations – Reserve and Main with different objective functions. The algorithm provides additional diversity to the search space while optimizing a test functions. The thread level parallelism between reserve and main populations gives better results in terms of less execution time and less number of generations. While comparing the performance of MPDPGA with DPGA[1], it is better in terms of accuracy, number of generations and execution time on multicore system. Comparing the performance of MPDPGA with DPGA-ED[1] it is found that, MPDPGA is better than DPGA-ED[1] for F1and F4, where as worse for F2and F3. From the profiling results of SGA, SDPGA and MPDPGA algorithms, it is observed that minimum three threads should be use in MPDPGA to speed up the process of solving optimization problems. It is also observed that the CPU utilization is optimized because of threading.

References

[1] August, A.D., Chiou, K.P.D., Sendag, R., Yi, J.J.: Programming Multicores: Do Application Programmers Need to Write Explicitly Parallel Programs? Computer Architecture Debates in IEEE MICRO, 19–32 (2010)

[2] Konfrst, Z.: Parallel Genetic Algorithm: Advances, Computing Trends, application and Perspective. In: Proceeding of 18th International Parallel and Distributed Processing Symposium (IPDPS 2004). IEEE Computer Society (2004)

[3] Cantú-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers (2000)

[4] Sefrioui, M., Périaux, J.: A Hierarchical Genetic Algorithm Using Multiple Models for Optimization. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 879–888. Springer, Heidelberg (2000)

[5] Rivera, W.: Scalable Parallel Genetic Algorithms. Artificial Intelligence Review 16(2), 153–168 (2001)

[6] Alba, E., Troya, J.M.: Analyzing synchronous and asynchronous parallel distributed genetic algorithms. Generation Computer Systems 17(4), 451–465 (2001)

[7] Alba, E., Troya, J.M.: Improving flexibility and efficiency by adding parallelism to genetic algorithms. Statistics and Computing 12(2), 91–114 (2002)

[8] Giacobini, M., Alba, E., Tomassini, M.: Selection Intensity in Asynchronous Celluar Evolutionary Algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, IL, vol. 2, pp. 955–966 (2003)

[9] Xiao, N., Amstrong, M.: A Specialized Island Model and Its Application in Multiobjective Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, IL, vol. 2, pp. 1530–1540 (2003)

[10] Gagné, C., Parizeau, M., Dubreuil, M.: The Master-Slave Architecture for Evolutionary Computations Revisited. In: Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, IL, vol. 2, pp. 1578–1579 (2003)

[11] Yang, S.: PDGA: the Primal Dual Genetic Algorithm. Design and Application of Hybrid Intelligent System, 214–223 (2003)

[12] Yang, S.: Nonstationary problem optimization using the primal-dual genetic algorithm. In: IEEE Congress on Evolutionary Computation (CEC 2003), pp. 2246–2253 (2003)

[13] Park, T., Ryu, K.R.: A dual population genetic algorithm with evolving diversity. In: IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3516–3522 (2007)

[14] Park, T., Ryu, K.R.: A dual population genetic algorithm with evolving diversity. In: Proc. IEEE Congr. Evol. Comput., pp. 3516–3522 (2007)

[15] Park, T., Choe, R., Ryu, K.R.: Dual population genetic algorithm for Nonstationary Optimization. In: Proc. Genetic Evol. Comput. Conf. (GECCO 2008), pp. 1025–1032 (2008)

[16] Junhua, L., Ming, L.: Genetic Algorithm with Dual Species. In: Proceeding of International Conference on Automation and Logistic, pp. 2572–2575 (2008)

[17] Molga, M., Smutnicki, C.: Test functions for optimization needs- (2005) (unpublished)

[18] Park, T., Ryu, K.R.: A dual population genetic algorithm with evolving diversity. In: Proc. IEEE Congr. Evol. Comput., pp. 3516–3522 (2007)

Dynamic Call Transfer through Wi-Fi Networks Using Asterisk

Mohammed Abdul Qadeer

Department of Computer Engineering, Aligarh Muslim University, Aligarh, India
maqadeer@gmail.com

Abstract. In the forthcoming time where the coverage area of the Wi-Fi networks would increase drastically, the transfer of call between two users present in a Wi-Fi network can be achieved thus reducing the effective cost of call rates significantly. The following article presents a view of how people in any Wi-Fi network can call each other via the network. Once the user is registered as a genuine user using any authentication methodology, the ASTERISK SERVER which has DHCP server configured provides an IP to any user requesting having a Wi-Fi enabled cell phone. Now, this user can talk to any other user in a Wi-Fi network by sending and receiving speech signals as data packets over the network. Since the dependence over the service provider is removed, overall cost is drastically reduced.

Keywords: Asterisk, VoIP, SIP, IAX, cellphone, call.

1 Introduction

Asterisk is a Private Branch Exchange (PBX). A PBX is a private phone switchboard, connecting to one or more telephones on one side to one or more telephone lines on the other. This is usually more cost effective than leasing a telephone line for each telephone needed in a business. Asterisk is an Open Source LINUX based server that can function as a PBX.

Asterisk is a framework that allows customization of modules, allowing us to create a phone system as per our own choice. Since Asterisk has a flexible architecture, it can be modified according to the requirements of a user or any organization. The call is achieved using Voice over Internet Protocol (VoIP) which costs far less than using the traditional telephony networks.

Till now, the use of asterisk has been restricted to Computers or special hardware devices manufactured by DIGIUM Inc. The aim of the paper is to elaborate a technique to develop an application for any Wi-Fi enabled cell phone to communicate to another such device using services of an Asterisk server through Wi-Fi "Fig. 1" rather than traditional telephony "Fig. 2".

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 51–61.
springerlink.com

In this technique, a user in a Wi-Fi network who has a Wi-Fi enabled cell phone first tries to locate an asterisk server in the network to make a call. If the server is found and destined user is also reachable through the server, then the call is put through via the asterisk server using VoIP else the call is forwarded through GSM cloud.

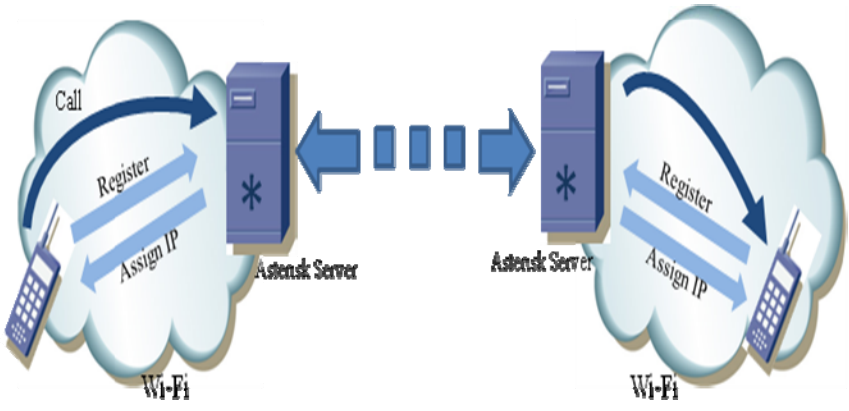


Fig. 1. Two cell phones communicating through asterisk server

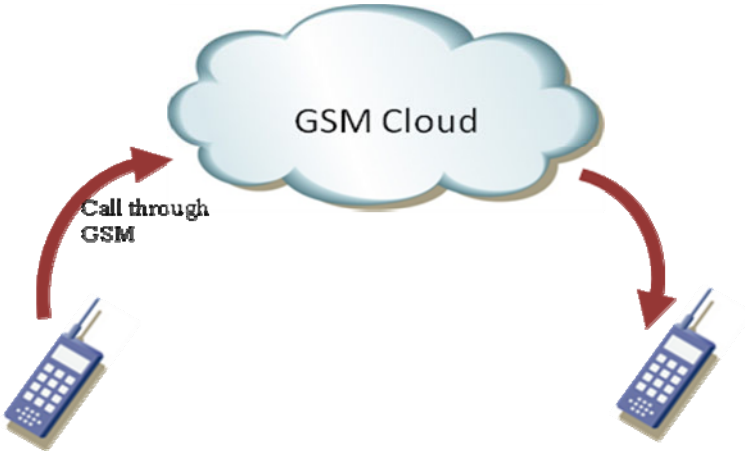


Fig. 2. Two phones communicating through traditional GSM

2 Functions of Asterisk

Asterisk has a whole range of features. Mentioned below are some of these features [1].

2.1 Station to Station Calls

Asterisk offers station-to-station calls. This means that users can dial from one phone to another phone, whether it is a hardphone or a softphone like Zoiper. Using Asterisk, we can call any particular phone registered with it.

2.2 Telephony

Asterisk supports all the basic features that a telephone company can provide like dialing, call waiting, call forwarding, caller ID.

2.3 Call Records

Asterisk can also receive a phone call and can make decisions based on the attributes of the call. Decisions like how to route calls, sending to multiple extensions, recording etc. can be made.

2.4 Call Detail Records

Asterisk keeps the record of all calls that go through it. We can store and use this information in our database for statistical and purposes.

2.5 Interactive Voice Record System

Asterisk has another powerful use that it can be used as an IVR through which we can take user responses for certain conditions and then we can issue commands on the basis of user input.

3 Asterisk and VoIP

Voice over Internet Protocol (VoIP) is a telephony network which allows telephone conversations to take place over a data packet switched network like the Internet. VoIP brings together both telecommunication and data services that are not only cheaper but also more versatile and can provide improved voice quality as compared to traditional telephony. This requires guaranteed bandwidth so as to enable data communication also.

In VoIP, the voice is digitized using an analog to digital converter (ADC) and the signal is sent through a data network which is reassembled to form the original analog signal using a digital to analog converter (DAC)[3].

Asterisk server depends on VoIP for transfer of data packets. It receives the packets from the respective users, analyses it and sends it to respective user or relays it to another asterisk server which may then transfer it further until the destined user gets the packet. All this is done using VoIP. There are various protocols for VOIP like SIP, IAX, H.323, MGCP etc [3]. Out of these, two prominent ones are SIP and IAX.

3.1 Need for VOIP Protocols

Since the basic idea of VOIP is the packetization of audio streams which have to be transported to long distances over IP based networks, there are certain challenges. Not only the packets are required to arrive in the same order but also this has to be achieved in a very short interval of time like less than 150 milliseconds. Out of order packets or lost packets will mean degradation in the quality of user level communication experience.

The original design of the internet protocols was not meant for real time media streaming. Endpoints were supposed to make up for the missing packets by requesting for them again and wait for all the packets to form a single coherent data. But we cannot do this for our audio signals where lost packets and reordering will have no significance at all. Also, the IP based protocols just divide the bits into desired packets size and send them but this is not always right for voice based packets as we require a sequential continuous data flow for sound quality. Hence, VOIP protocols were born.

3.2 VOIP Protocols

The basic mechanism of VOIP protocols is to have a set of transactions between the endpoints and thus resulting in twp persistent data streams that carry the conversation. This is achieved using a VOIP protocol some of them are discussed below [1].

3.2.1 IAX

The IAX protocol was developed by Digium designed for communicating with other Asterisk servers. IAX is a transport protocol that uses a single UDP port (4569) for both the channel signaling and media streams. Use of IAX can be large bandwidth advantages as it can trunk multiple sessions into one dataflow when sending a lot of simultaneous channels to a remote box. A single datagram can thus represent multiple media multiple media streams which can lower the overhead associated with individual channels. This helps to lower latency and reduce the processing power and bandwidth required, allowing the protocol to scale much more easily with a large number of active channels between endpoints. Hence, if large quantities of IP calls have to be passed between two endpoints one can go for IAX. As IAX is an open protocol, it can carry any future media streams that may be desired at that moment by bringing changes in the IAX as the need may be. Plain text, MD5 hashing, and RSA key exchange are the three authenticating ways of IAX. Media path is not encrypted between endpoints. To achieve this, we can switch for several existing solutions including using a Virtual Private Network (VPN) appliance or software to encrypt the stream in another layer of technology. But with dynamic key exchange at call setup IAX is now also able to encrypt the streams between endpoints.

3.2.2 SIP

The technique used in SIP is that each end of a connection is a peer and the protocol negotiates capabilities between them. SIP has grown in popularity as it is a relatively simple protocol, with syntax similar to that of other familiar protocols such as HTTP and SMTP. SIP is supported in Asterisk with the chan_sip.so module. SIP is an application-layer signaling protocol and uses the port 5060 for communications. SIP can be transported with either the UDP or TCP transport-layer protocols. Asterisk does not currently have a TCP implementation for transporting SIP messages, but it is possible that future versions may support it. SIP is used to "establish, modify, and terminate multimedia sessions such as Internet telephony calls". Since SIP does not transport media between endpoints, RTP is used to transmit media "Fig. 3" between endpoints. RTP uses high-numbered, unprivileged ports in Asterisk (10,000 through 20,000, by default). [1]

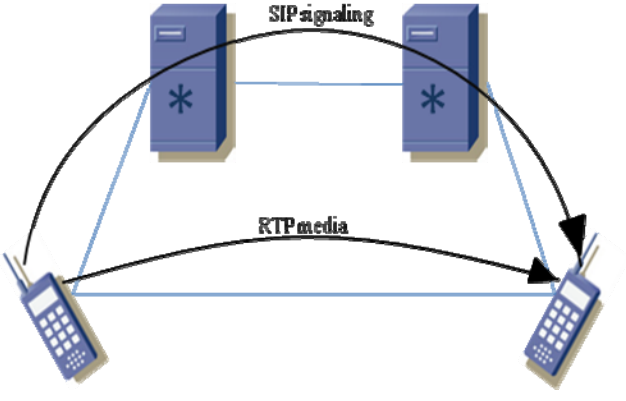


Fig. 3. The SIP trapezoid [1]

SIP trapezoid as a common topology to illustrate SIP and RTF. When user A wants to call another user B, A's phone contacts the B's proxy server and the proxy tries to find B (often connecting through his proxy). Once the phones have started the call, they communicate directly with each other (if possible), so that the data doesn't have to tie up the resources of the proxy.

The advantages of the SIP protocol lie in its wide acceptance and architectural flexibility. SIP is widely expected to deliver far more than VoIP capabilities, including the ability to transmit video, music, and any type of real-time multimedia. SIP is considered to deliver the majority of new voice applications but IAX appears to perform better in comparison with the established SIP protocol [5].

A challenge/response system is used to authenticate users. An initial INVITE is sent to the proxy whom the end device wishes to communicate. The proxy then sends back a 407 Proxy Authorization Request message, which contains a random set of characters referred to as a *nonce*. This once is used along with the password to generate an MD5 hash, which is then sent back in the subsequent INVITE.

Assuming the MD5 hash matches the one that the proxy generated, the client is then authenticated. Denial of service (DoS) attacks are probably the most common type of attack on VoIP communications. A DoS attack can occur when a large number of invalid INVITE requests are sent to a proxy server in an attempt to overwhelm the system. These attacks are relatively simple to implement, and their effects on the users of the system are immediate. SIP has several methods of minimizing the effects of DoS attacks, but ultimately they are impossible to prevent. SIP implements a scheme to guarantee that a secure, encrypted transport mechanism is used to establish communication between the caller and the domain of the callee. Beyond that, the request is sent securely to the end device, based upon the local security policies of the network.

4 Functions of the Mobile Application

The application will be installed on a Wi-Fi enabled cell phone. When a user wants to make a call to some other phone in the organization, the application will force the call to be through a Wi-Fi network via an Asterisk server, if available.

The application will perform following actions:

4.1 Register

When the user will enter the asterisk zone of an organization, i.e. a Wi-Fi network with an asterisk server, the application will try to register the cell phone along with the user profile on the phone. The caller-id number of the user will be same as the user's phone number. If the phone gets registered successfully, it will be provided with an IP from the DHCP server. The user's caller-id number and its IP address will be added in the Asterisk server's database.

4.2 Make a Call through Asterisk Server

When the user wants to make a call to another phone and dials the phone number, the application will request the asterisk server to find the dialed number in its database or to search for some other asterisk server in the organization having that number in its database. If the number is found then the corresponding IP address will be used to make the call through the asterisk server over VoIP. However if the requested number is not found, that means that phone is currently not registered to any asterisk server and/or is not in any asterisk zone, and the call will be through the GSM cloud "Fig. 3".

4.3 Receive a Call

The phone registered on an asterisk server can receive calls either through VoIP or the GSM cloud, depending upon the registration status of the caller, whether it has been registered on any asterisk server or not.

4.4 Unregister

When the phone leaves the area of the asterisk server, it will get unregistered. Its phone number and the IP assigned to it are deleted from the database of the asterisk server. This can also happen if the user deliberately severs the connection with the server.

5 Problems

There are some problems to be faced when making a call such as:

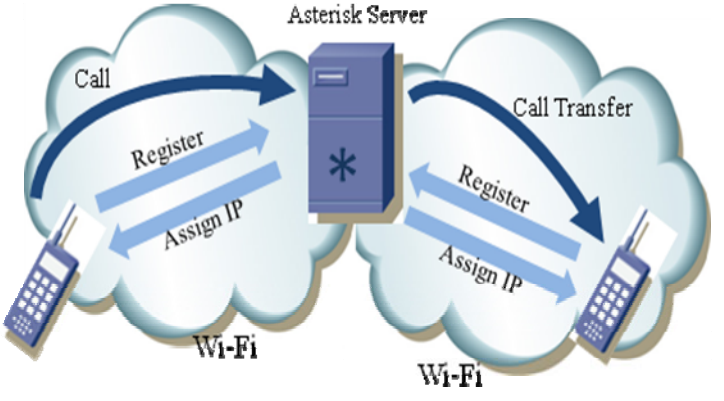


Fig. 4. Two cell phones on the same Wi-Fi network

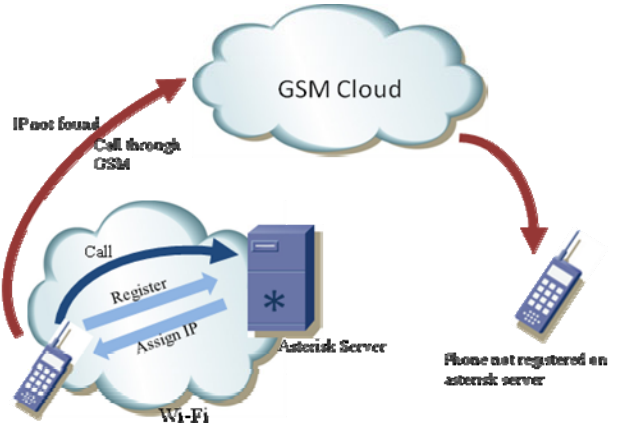


Fig. 5. Two cell phones communicating when one user is not registered on asterisk

5.1 Finding the IP of the Dialed Number

If both the phones between which communication is sought are in the same Asterisk zone then there is no problem with finding the final IP. The problem arises when both the phones are in different asterisk zones. IP's of both the phones will be added in databases of different asterisk servers. The caller's asterisk server will have to find the IP of the callee by requesting a large number of other asterisk servers to find the IP until the IP is found or it is found that the dialed number is not registered on any asterisk sever.

One solution may be, when a phone registers on an asterisk server, that server will broadcast that information to all other asterisk servers to update the database of all the asterisk servers. This method is feasible only when the number of asterisk servers is low, number of phones registered is low, and the frequency of registering and unregistering of phones is low. But in the case of large number of phones registered under a large number of different asterisk servers, this method becomes cumbersome and space & time demanding.

6 More Services

Asterisk has a well-through-out architecture which can be configured as our need, to give the maximum flexibility. Apart from the functions that the application will perform as discussed above, there are more services that the application can perform after creating custom modules.

6.1 Short Message Service

We can create a custom module so that the asterisk server may also be used for short message services as well. The user can send and receive messages just like any other service.

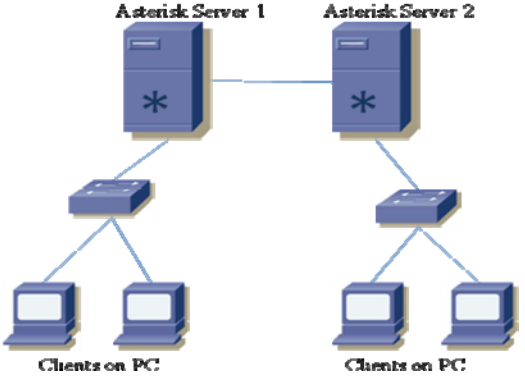


Fig. 6. PC to PC call on a LAN through Asterisk servers

6.2 Social Networking

Another future work can be social networking where several users can create and view profiles and exchange information.

7 Our Testbeds

There are three kinds of testbeds on which Asterisk server can be used to send a call through. These are explained below.

7.1 PC to PC Call on a LAN

We can make a call between two or more PCs through Asterisk Server on a LAN using a Softphone "Fig 7". The two PCs may be registered on the same server or different Asterisk Servers as long as the servers are connected through LAN. The configurations of various asterisk users who are supposed to interact have to be entered on the servers. When a user wishes to make a call to another user who is also registered on any connected server, the user can simply dial the other user's number and the call gets completed when the other user accepts the call. The various status messages on the server are shown in the "Fig. 8".

Fig. 7. PC to PC call on a LAN through Asterisk servers

7.2 Call between Wireless Computing Devices

Asterisk can also be used to make a call between several wireless users who may be registered on same or different Asterisk servers and are connected through Wireless LANs. This service may help the users to remain connected as they move around in an office or an institution "Fig 9".

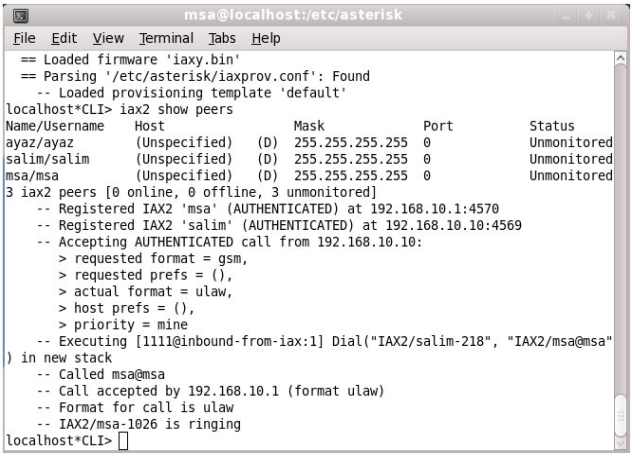


Fig. 8. Console showing call setup status

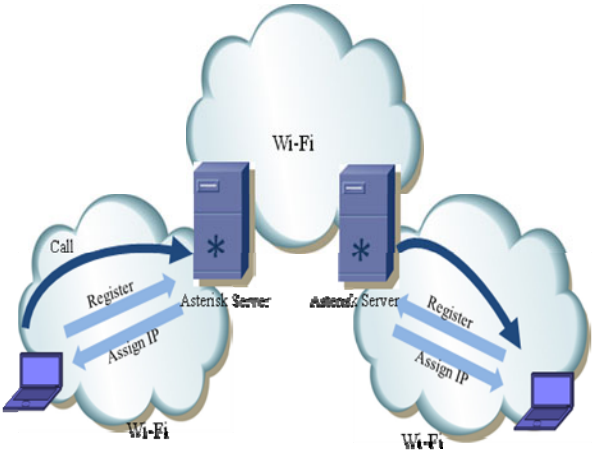


Fig. 9. Call between Laptops on Wi-Fi through Asterisk servers

7.3 Call between Wi-Fi Cell Phones

The call rates between several users through their cell phones can be drastically reduced if the users are able to call through VoIP using Asterisk. As soon as the users enter an Asterisk enabled Wi-Fi area, they get connected to the Asterisk server. When a user registered on the Server calls another user, then first the call is forwarded to the local Asterisk server. If the user is found, then the call is sent through the server, else the call is sent through the GSM network. This will make calls much cheaper as compared to normal GSM telephony.

7.4 A Typical Office Scenario

In this scenario, we have different kinds of devices like PCs, Laptops and Cell Phones. The above mentioned techniques of call transfer can be implemented in an institution or office as depicted in the "Fig 10".

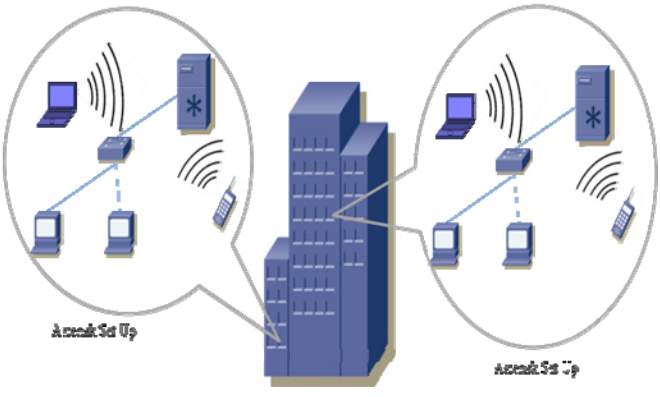


Fig. 10. Typical Office Scenario

References

[1] Van Meggelen, J., Madsen, L., Smith, J.: Asterisk: The future of telephony, 2nd edn. O'Reilly Media, Inc.

[2] Gomillion, D., Dempster, B.: Building telephony systems with asterisk. Packt Publishing Ltd. (2005)

[3] Mao, G.F., Talevskit, A., Chang, E.: Voice over internet protocol on mobile devices. In: 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS (2007)

[4] Rosenberg, J., Schuluinne, H., Camarilla, G., Johnston, A., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. The Internet Engineering Task Force (June 2002)

[5] Abbasi, T., Prasad, S., Seddigh, N., Lambadaris, I.: A comparative study of the sip and iax voip protocols. In: 2005 IEEE CCECE/CCGEI, Saskatoon (2005)

[6] IAX protocol RFC: `http://www.rfc-editor.org/authors/rfc5456.txt`

Differential Evolution Strategies for Multi-objective Optimization

Ashish M. Gujarathi[1] and B.V. Babu[2]

[1] Chemical Engineering Department,
Birla Institute of Technology and Science (BITS), Pilani – 33031, India
ashishg@bits-pilani.ac.in, ashishgujrathi@gmail.com
[2] Institute of Engineering and Technology, JKLU, Jaipur – 302026, India
profbvbabu@gmail.com

Abstract. Multi-objective optimization (MOO) using evolutionary algorithms has gained popularity in the recent past due to its ability of producing number of solutions in a single run and handling multiple objectives simultaneously. In this effort, several MOO algorithms are developed. In this manuscript several strategies of multi-objective differential evolution algorithm (namely, MODE-I, MODE-III, elitist MODE and hybrid MODE) are briefly discussed. Three important unconstrained test problems are considered for validating the performance (in terms of Pareto front and convergence & diversity metrics) of strategies of MODE algorithm with other popular algorithms from literature. It is observed that the strategies of MODE algorithm are in general able to produce Pareto front with good convergence to the true Pareto front.

Keywords: Differential Evolution, Multi-objective Differential Evolution (MODE), Evolutionary Algorithms (EAs), Pareto front, Multi-objective optimization (MOO).

1 Introduction

An optimization problem involving more than one objective to be optimized is referred as multi-objective optimization (MOO) problem. The optimum solution corresponding to a single objective optimization problem refers to the optimal basic feasible solution (satisfying bounds of variables and the constraints). However, in case of multi-objective optimization, the optimum solution refers to a compromised (not necessarily the optimum with respect to any objective) set of multiple feasible solutions. In case of MOO problems, the decision maker is always interested in obtaining a solution suitable to his/her design requirements, i.e., a single solution. But due to the associated trade-off and multiple objectives nature of problem, the desired solution may vary as per the decision makers need and the choice. Evolutionary Algorithms (EAs) often results in set of solutions.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 63–71.
springerlink.com

The specialty of such solutions is that as we move from one solution to the other we gain in terms of one objective at the cost of loss in another objective involved in the study. Such a set of solutions are referred as the Pareto optimal set and the solutions in this set are non-dominated with respect to each other.

The multi-objective optimization algorithms can be broadly classified into classical or traditional methods (deterministic) and nontraditional or population based search algorithms. Since last two decades, evolutionary population based search algorithms have captured more attention due to their ability of giving number of population points in a single simulation run. Classical methods also have their own place in the research of multi-objective optimization due to their simple transition rules as compared to evolutionary algorithms. The population based search algorithms have specific characteristics that they start with multiple population points and all points usually converge towards the Pareto optimal front after required number of simulation runs. The detailed survey on some of the evolutionary algorithms (population based search algorithms) is given in the literature [1 – 9]. In the following section several strategies of MODE algorithm are briefly discussed.

1.1 Strategies of Differential Evolution for MOO

Differential evolution (DE) [10 – 13] is an improved version of Genetic Algorithms (GA) [14]. Multi-objective differential evolution (MODE) [15 – 16] is an extension of differential evolution for multi-objective optimization study. MODE algorithm has been successfully used for multi-objective optimization real world industrial problems and many complex test problems.

MODE Algorithm. The simplified flowchart of the original MODE algorithm (MODE-1) is presented in Fig. 1a. The pseudo-code for MODE algorithm can be found in literature [15 - 16]. In MODE-1 algorithm, in each generation, the dominated solutions are removed from the list and only the non-dominated solutions are allowed to undergo DE operations. The scaling factor is generated from a random number generator between 0 and 1. The off springs are placed into population if they dominate the main parent.

MODE-III Algorithm. The recombination operation in DE is proved to be a powerful technique [10, 17]. In recombination, a competition is made between trial (child) and target (Parent) vectors. The non-dominated vector among the trial and target is sent to the next generations (using survival of the fittest principle). The domination check alone can give the Pareto optimal front. Therefore the non-dominated sorting before the DE loop is removed from original MODE algorithm. At the end of the maximum generations a non-dominated sorting check is kept to remove the individuals which are not on the Pareto optimal front. The simplified flowchart of the MODE-III algorithm is presented in Fig. 1b. Detailed working principle of MODE-III algorithm is given in our earlier publication [18].

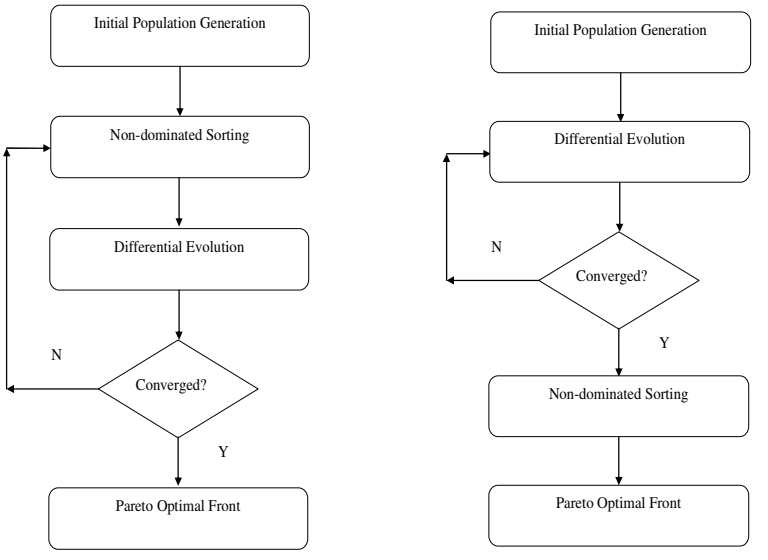


Fig. 1. Schematic diagrams of MODE (Fig. a) and MODE-III (Fig. b) algorithms

Elitist MODE Algorithm. There are three major steps in elitist MODE algorithm: (1) Processing of initial population of size *NP* using differential evolution, (2) Combining the solutions obtained from DE (*NP*) and those obtained from non-dominated sorting (*Q*) to get a total of (*NP*+*Q*) solutions ensuring elitism, and (3) Maintaining *NP* number of population points in the next generation by using a crowding distance sorting. The simplified flowchart of the elitist MODE algorithm is presented in Fig. 2. The detailed working principle and schematic diagram of elitist MODE is available in our publication [19]. Industrial applications of Elitist MODE algorithms and its performance with respect to other algorithms is available in the literature [20 - 21].

Hybrid MODE Algorithm. The deterministic sequential simplex method [22, 11] is used for local search, whereas one of the evolutionary multi-objective differential evolution strategy is used for global search. The deterministic method is used as an accelerator, which finds new superior points to converge to the Pareto front at a faster rate. The simplified flowchart of the hybrid MODE algorithm is presented in Fig. 3. Detailed working principle of hybrid multi-objective differential evolution algorithm along with its performance on industrial applications is available in our recent publications [18, 23]. The population set is initialized randomly within the specified bounds of the variables and the corresponding costs (objective function value) are evaluated. Entire population points are preserved for recombination operation. Three vectors from the initial population are selected at random in order to create a noisy random vector. Then the trial (child) vector is generated by cross over between the target and noisy random vectors (parents).

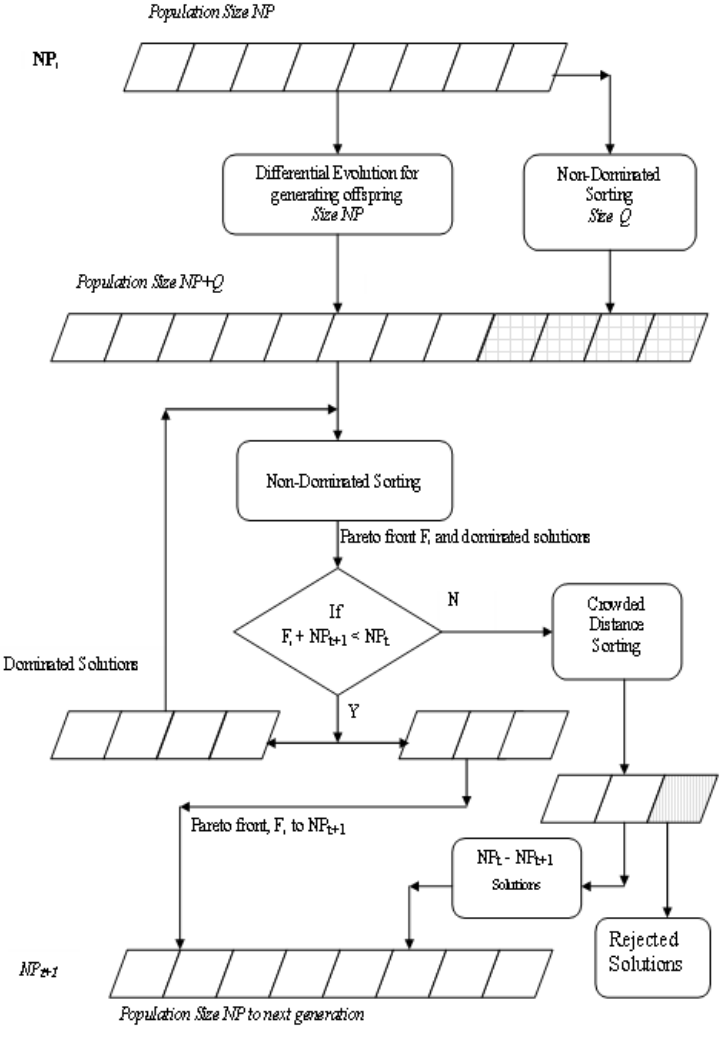


Fig. 2. Schematic diagram of elitist MODE algorithm [19 – 21]

The cost of the trial vector is compared with that of the target vector for dominance. The winner vector is then used for local search to obtain a better population point in the neighbourhood using sequential simplex algorithm. If the point obtained using sequential simplex method dominates any point in the current population, the dominated point gets replaced by the new point, and otherwise a new selection procedure takes place. As a local search method is applied on each point, not only convergence becomes faster, but diversity of solutions is also maintained.

2 Results and Discussion

Two important and widely used metrics are employed to calculate the convergence and divergence extents of algorithm to assess the quality of the Pareto front. Three different test problems are considered. The unconstrained test problems (1) SCH [24] has a convex search space and a convex Pareto front, (2) FON [25] has a non-convex search space and a non-convex Pareto front, (3) KUR [26] has a non-convex search pace and a non-convex and discontinuous (or disconnected) Pareto front. The values of performance metrics (convergence and diversity metrics) obtained using different algorithms are shown in Table 1. The average (γ for convergence and Δ for divergence) and the variance (σ^2_γ for convergence and σ^2_Δ for divergence) of 10 individual runs are used [as given in the literature [3]] to judge the performance of algorithms for both the metrics.

For the SCH test function, all the strategies of MODE and NSGA-II algorithms are able to converge to the true Pareto optimal solutions (Fig. 4a). However, a smooth and well diverse Pareto front is observed in case of Hybrid MODE, and

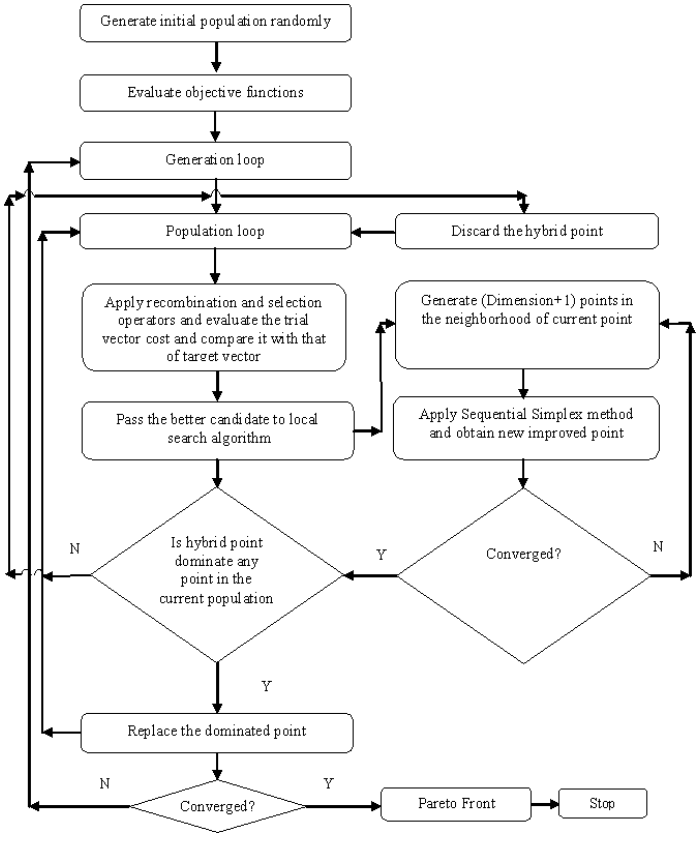


Fig. 3. Schematic diagram of hybrid MODE algorithm [18]

NSGA-II (both real- and binary coded versions). The number of non-dominated solutions obtained using hybrid MODE, and Real- coded NSGA-II is 100 against 40 obtained using MODE III algorithm. The lesser number of non-dominated solutions resulted in a scattered Pareto front using MODE and MODE III. Table 1 shows that hybrid MODE, elitist MODE, and PAES are equally good in terms of average of convergence metric. However, the variance for all the strategies of MODE algorithm is 0 as against 0.000003 obtained in case of PAES. The variance value of 0 also indicates that the algorithm is able to give consistent results. Real coded NSGA-II algorithm performed better in terms of diversity metric achieving the lowest value of 0.449265.

Table 1. Performance metrics (convergence and divergence) comparison of several algorithms on selected test problems

Algorithm	Metric	SCH	FON	KUR
Real coded	γ	0.003391	**0.001931**	0.028964
NSGA-II˜	σ^2_γ	**0**	**0**	0.000018
	Δ	*0.477899*	**0.378065**	**0.411477**
	σ^2_Δ	0.003471	**0.000639**	**0.000992**
NSGA-II	γ	0.002833	0.002571	0.028951
Binary˜	σ^2_γ	*0.000001*	**0**	0.000016
	Δ	**0.449265**	*0.395131*	*0.442195*
	σ^2_Δ	0.002062	0.001314	0.001498
SPEA˜	γ	0.003465	0.010611	0.049077
	σ^2_γ	**0**	*0.000005*	0.000081
	Δ	0.818346	0.804113	0.880424
	σ^2_Δ	0.004497	0.002961	0.009066
PAES˜	γ	**0.001313**	0.151263	0.057323
	σ^2_γ	0.000003	0.000905	0.011989
	Δ	1.063288	1.162528	1.079838
	σ^2_Δ	0.002868	0.008945	0.013772
MODE*	γ	0.0021	0.02554	0.03837
	σ^2_γ	**0**	0.00063	0.00057
	Δ	0.67099	0.70069	0.82097
	σ^2_Δ	0.01332	0.03397	0.0053
MODE III*	γ	0.002236	0.003381	*0.003028*
	σ^2_γ	**0**	**0**	**0**
	Δ	0.59953	0.620052	0.671036
	σ^2_Δ	0.00155	0.00095	0.00192
Hybrid	γ	*0.001967*	0.002807	0.003723
MODE*	σ^2_γ	**0**	**0**	**0**
	Δ	0.597286	0.538185	0.675896
	σ^2_Δ	*0.001238*	*0.000891*	0.002681
Elitist	γ	**0.001948**	*0.002119*	**0.002921**
MODE*	σ^2_γ	**0**	**0**	**0**
	Δ	0.571475	0.700227	0.714344
	σ^2_Δ	0.006496	0.018964	0.003301

* Results obtained in the present study; ˜Results reported in Ref [3]
First Best: **Bold** Font; Second Best: *Italic* font.

The Pareto front for the test problem FON obtained using several algorithms is shown in Fig. 4b. The Pareto optimal solutions correspond to $x_i^* = -1/\sqrt{3}$. MODE algorithm resulted in only 3 numbers of points on the Pareto front (out of initial population of 100) after a specified numbers of generations.

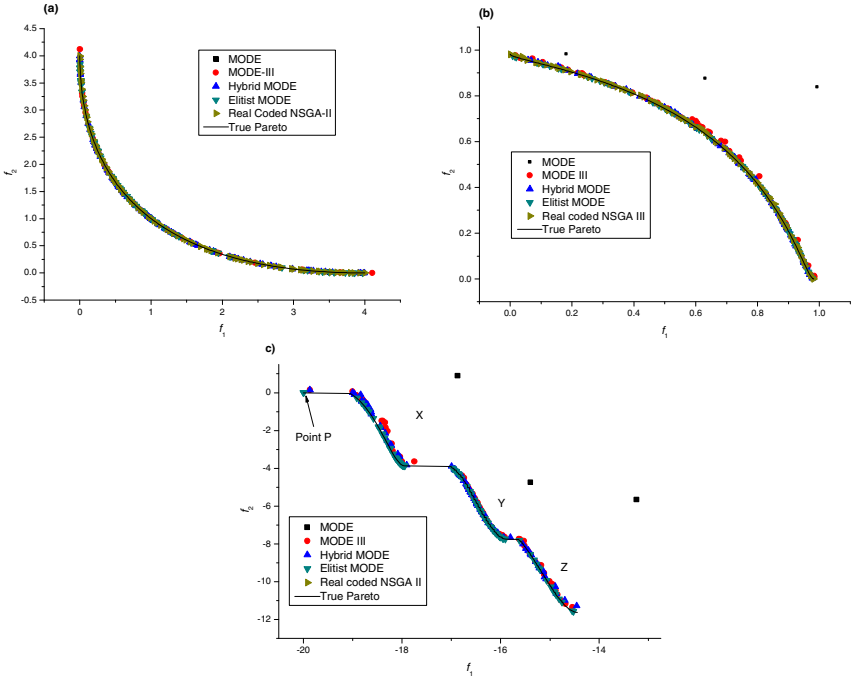


Fig. 4. Pareto fronts of SCH (Fig. a), FON (Fig b) and KUR (Fig. c) obtained using strategies of multi-objective differential algorithm

All the algorithms are able to cover the entire range of solutions on the Pareto front except MODE algorithm. Comparison of Pareto fronts show that the non-dominated solutions obtained with MODE III are slightly scattered and are away from the global Pareto front. However, the Pareto fronts obtained using other strategies of MODE and NSGA-II algorithms are converged to the true Pareto fronts. In terms of convergence, both the versions of NSGA (i.e., binary and real coded) and strategies of MODE (i.e., MODE III, Elitist MODE, and hybrid MODE) attained good accuracy as compared to other algorithms such as SPEA and PAES. However, diversity metric of NSGA-II (Binary) is better as compared to other algorithms. Elitist MODE algorithm resulted in first best values in terms of convergence metric (as shown in Table 1).

KUR test problem has multiple disconnected non-convex Pareto fronts as shown in Fig. 4c. The special feature of KUR test problem is to obtain the unique

point (such as point *P*) in Fig. 4c. Point *P* is a non-dominated solution with $x_i^* = 0$ for all the variables. Region *y* and region *z* (as shown in Fig 4c) have a good distribution of solutions as compared to region *x*. Region *x* and point *P* appear with a value of decision variable (DV) $x_1 = 0$, while region *x* and region *y* correspond to $x_1 < 0$. Hybrid MODE and MODE III algorithms approached the Pareto front to the closest accuracy as compared to other algorithms. However, diversity of solutions corresponding to real coded NSGA-II is better than that obtained with other studied algorithms.

3 Conclusions

Brief working principles of four different strategies of multi-objective differential evolution algorithm (namely, MODE-I, MODE-III, elitist MODE and hybrid MODE) are discussed. Three widely used unconstrained test problems are used (namely, SCH, POL and KUR). Important performance metrics (such as convergence and diversity metrics) are calculated for each test problems using the different strategies of MODE algorithm. The Pareto front obtained using each strategy of MODE and real coded NSGA-II algorithms are reported. The results obtained using strategies of MODE are compared with the results reported in the literature. It is observed that strategies of MODE algorithms are in general able to converge to the true Pareto front. However, there is a scope for improvement in the diversity issue.

References

[1] Back, T.: Evolutionary algorithms in theory and practice. Oxford University Press, New York (1996)
[2] Miettinen, K.M.: Nonlinear multiobjective optimization. Kluwer Academic Publishers, Boston (1999)
[3] Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley and Sons Limited, New York (2001)
[4] Onwubolu, G.C., Babu, B.V.: New optimization techniques in engineering. Springer, Heidelberg (2004)
[5] Eiben, A.E., Smith, J.E.: Introduction to evolutionary computing. Springer, Hiedelberg (2003)
[6] Coello, C.A.C., Lamont, G.B.: Applications of multi-objective evolutionary algorithm: Advances in natural computation, vol. 1. World Scientific Publishing Co. Pte. Ltd., Singapore (2004)
[7] Tan, K.C., Khor, E.F., Lee, T.H.: Multi-objective evolutionary algorithms and applications. Springer, London (2005)
[8] Biswas, A., Chakraborti, N., Sen, P.K.: A genetic algorithms based multi-objective optimization approach applied to a hydrometallurgical circuit for ocean nodules. Min. Processing. Ext. Metal. Rev. 30, 163–189 (2009)
[9] Biswas, A., Chakraborti, N., Sen, P.K.: Multiobjective optimization of manganese recovery from sea nodules using genetic algorithms. Mat. Manuf. Processes 24, 22–30 (2009)

[10] Price, K.V., Storn, R.: Differential evolution - a simple evolution strategy for fast optimization. Dr. Dobb's J. 22, 18–22 (1997)
[11] Babu, B.V.: Process plant simulation. Oxford Press, New York (2004)
[12] Babu, B.V., Angira, R.: Optimal design of an auto-thermal ammonia synthesis reactor. Compu. Che. Engng. 29, 1041–1045 (2005)
[13] Babu, B.V., Angira, R.: Modified differential evolution (MDE) for optimization of non-linear chemical processes. Compu. Che. Engng. 30, 989–1002 (2006)
[14] Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addission-Wesley, Reading (1989)
[15] Babu, B.V., Chakole, P.G., Mubeen, J.H.S.: Multi-objective differential evolution (MODE) for optimization of adiabatic styrene reactor. Che. Engng. Sci. 60, 4822–4837 (2005)
[16] Gujarathi, A.M., Babu, B.V.: Multi-objective Optimization of Industrial Styrene Reactor: Adiabatic and Pseudo-isothermal Operation. Che. Engng. Sci. 65, 2009–2026 (2010)
[17] Angira, R.: Evolutionary computation for optimization of selected nonlinear chemical processes. Ph. D. Thesis, Birla Institute of Technology and Science (BITS), Pilani, India (2005)
[18] Gujarathi, A.M., Babu, B.V.: Optimization of Adiabatic Styrene Reactor: A Hybrid Multi-Objective Differential Evolution (H–MODE) Approach. Ind. Engng. Che. Res. 48, 11115–11132 (2009)
[19] Babu, B.V., Gujarathi, A.M.: Elitist-Multi-Objective Differential Evolution (E-MODE) Algorithm for Multi-Objective Optimization. In: Proceedings of 3rd Indian International Conference on Artificial Intelligence (IICAI 2007), Pune, pp. 441–456 (2007)
[20] Gujarathi, A.M., Babu, B.V.: Improved Multi-Objective Differential Evolution (MODE) Approach for Purified Terephthalic Acid (PTA) Oxidation Process. Mat. Manuf. Processes 24, 303–319 (2009)
[21] Gujarathi, A.M., Babu, B.V.: Elitist Multi-Objective Differential Evolution Algorithm for Multi-Objective Optimization of Industrial Styrene Reactor. App. Comput. Intel. Soft. Comp. (accepted) (in Press)
[22] Nelder, J.M., Mead, R.: A Simplex method for function minimization. The Comput. J. 7, 308–313 (1965)
[23] Gujarathi, A.M., Babu, B.V.: Hybrid Multi-objective Differential Evolution (H-MODE) for optimization of Polyethylene Terephthalate (PET) reactor. Int. J. Bio-insp. Compu. 2, 213–221 (2010)
[24] Schaffer, J.D.: Some experiments in machine learning using vector evaluated genetic algorithms. Ph. D. Thesis, Vanderbilt University, Nashville, T.N (1985)
[25] Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evo. Comput. J. 3, 1–16 (1995)
[26] Kursawe, F.: A Variant of Evolution Strategies for Vector Optimization. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1990)

Dynamic Scaling Factor Based Differential Evolution Algorithm

Harish Sharma[1], Jagdish Chand Bansal[2], and K.V. Arya[3]

[1] ABV-Indian Institute of Information Technology and Management, Gwalior
harish0107@rediffmail.com
[2] ABV-Indian Institute of Information Technology and Management, Gwalior
jcbansal@gmail.com
[3] ABV-Indian Institute of Information Technology and Management, Gwalior
kvarya@gmail.com

Abstract. Differential evolution (DE) is a well known and simple population based probabilistic approach used to solve nonlinear and complex problems. It has reportedly outperformed a few evolutionary algorithms when tested over both benchmark and real world problems. DE, like other probabilistic optimization algorithms, has inherent drawback of premature convergence and stagnation. Therefore, in order to find a trade-off between exploration and exploitation capability of DE algorithm, scaling factor in mutation process is modified. In mutation process, trial vector is calculated by perturbing the target vector. In this paper, a dynamic scale factor is proposed which controls the perturbation rate in mutation process. The proposed strategy is named as Dynamic Scaling Factor based Differential Evolution Algorithm ($DSFDE$). To prove efficiency of $DSFDE$, it is tested over 10 benchmark problems.

Keywords: Optimization, Differential Evolution Algorithm, Scale Factor, Mutation Process, Meta-heuristics.

1 Introduction

Differential Evolution (DE) scheme is relatively a simple, fast and population based stochastic search technique, proposed by Storn and Price [15] . DE falls under the category of Evolutionary Algorithms (EAs). But in some sense it differs significantly from EAs, e.g. trial vector generation process (explained in section 2) uses the information of distance and direction from current population to generate a new trial vector. Furthermore,in EAs, crossover is applied first to generate a trial vector, which is then used within the mutation operation to produce one offspring while, in DE, Mutation is applied first and then crossover.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 73–85.
springerlink.com

Researchers are continuously working to improve the performance of DE. Some of the recently developed versions of DE with appropriate applications can be found in [2]. Experiments over several numerical benchmarks [19] show that DE performs better than the genetic algorithm (GA) [8] or the particle swarm optimization (PSO) [9]. DE has successfully been applied to various areas of science and technology, such as intrusion detection [14], chemical engineering [11], signal processing [4], mechanical engineering design [17], machine intelligence, and pattern recognition [13]. Recently, machine intelligence and cybernetics are most well-liked field in which DE algorithm has become a popular strategy.

There are two fundamental processes which drive the evolution of a DE population: the variation process, which enables exploring different areas of the search space, and the selection process, which ensures the exploitation of the previous experience. However, it has been shown that DE may occasionally stop proceeding toward the global optimum even though the population has not converged to a local optimum [10]. Therefore, to maintain the proper balance between exploration and exploitation behavior of DE, a concept of dynamic scaling factor is introduced in DE and DE with this concept is named as Dynamic Scaling Factor based Differential Evolution Algorithm ($DSFDE$). Furthermore, to show the efficiency of the proposed strategy, it is tested over 10 well-known benchmark optimization problems.

Rest of the paper is organized as follows: Section 2 describes brief overview of basic Differential Evolution Algorithm. Dynamic Scaling Factor based Differential Evolution ($DSFDE$) is proposed in Section 3. In section 4, experiments has been carried out. Finally, in section 5, paper is concluded.

2 Brief Overview of Differential Evolution Algorithm

DE has several strategies based on method of selecting the target vector, number of difference vectors used and the type of crossover [15]. In this paper $DE/rand/1/bin$ scheme is used where DE stands for differential evolution, 'rand' specifies that the target vector is selected randomly, '1' is for number of differential vectors and 'bin' notation is for *binomial* crossover. The popularity of Differential Evolution is due to its applicability to a wider class of problems and ease of implementation. Differential Evolution consists of the properties of both evolutionary algorithms and swarm intelligence. The detailed description of DE is as follows:

Like other population based search algorithms, in DE a population of potential solutions (individuals) searches the solution. In a D-dimensional search space, an individual is represented by a D-dimensional vector $(x_{i1}, x_{i2}, ..., x_{iD})$, $i = 1, 2, ..., NP$ where NP is the population size (number of individuals).

In DE, there are three operators: mutation, crossover and selection. Initially, a population is generated randomly with uniform distribution then the mutation, crossover and selection operators are applied to generate a new population. Trial vector generation is a crucial step in DE process. The two operators mutation and crossover are used to generate the trial vectors. The selection operator is used to select the best trial vector for the next generation. DE operators are explained briefly in following subsections.

2.1 Mutation

A trial vector is generated by the DE mutation operator for each individual of the current population. For generating the trial vector, a target vector is mutated with a weighted differential. An offspring is produced in the crossover operation using the newly generated trial vector. If G is the index for generation counter, the mutation operator for generating a trial vector $u_i(G)$ from the parent vector $x_i(G)$ is defined as follows:

- Select a target vector, $x_{i1}(G)$, from the population, such that $i \neq i1$.
- Again, randomly select two individuals, x_{i2} and x_{i3}, from the population such that $i \neq i_1 \neq i_2 \neq i_3$.
- Then the target vector is mutated for calculating the trial vector as follows:

$$u_i(G) = x_i(G) + F(x_{i2}(G) - x_{i3}(G)) \tag{1}$$

where $F \in (0, \infty)$ is the mutation scale factor which is used in controlling the amplication of the differential variation [6].

2.2 Crossover

Offspring $x'_i(G)$ is generated using the crossover of parent vector, $x_i(G)$ and the trial vector, $u_i(G)$ as follows:

$$x'_{ij}(G) = \begin{cases} u_{ij}(G), & \text{if } j \in J \\ x_{ij}(G), & \text{otherwise.} \end{cases}$$

J is the set of cross over points or the points that will go under perturbation, $x_{ij}(G)$ is the j^{th} element of the vector $x_i(G)$.

2.3 Selection

There are two functions of the selection operator: First it selects the individual for the mutation operation to generate the trial vector and second, it selects the best, between the parent and the offspring based on their fitness

value for the next generation. If fitness of parent is greater then the offspring than parent is selected otherwise offspring is selected:

$$x_i(G+1) = \begin{cases} x'_i(G), & \text{if } f(x'_i(G)) > f(x_i(G)). \\ x_i(G), & \text{otherwise.} \end{cases}$$

This ensures that the population's average fitness does not deteriorate.

The Pseudo-code for Differential Evolutionary strategy, is described as follows [6]:

Differential Evolution Algorithm:
Initialize the control parameters, F and CR;
Create and initialize the population, $P(0)$, of NP individuals;
while stopping condition(s) not true **do**
  for each individual, $x_i(G) \in P(G)$ **do**
    Evaluate the fitness, $f(x_i(G))$;
    Create the trial vector, $u_i(G)$ by applying the mutation operator;
    Create an offspring, $x'_i(G)$, by applying the crossover operator;
    if $f(x'_i(G)$ is better than $f(x_i(G))$ **then**
      Add $x'_i(G)$ to $P(G+1)$;
    else
      Add $x_i(G)$ to $P(G+1)$;
    end if
  end for
end while
Return the individual with the best fitness as the solution;
space

Here, F, scale factor and CR, crossover probability are the control parameters and influence the performance of the DE. P is the population vector.

3 Dynamic Scaling Factor Based Differential Evolution

3.1 A Few Drawbacks of DE

The inherent drawback with most of the population based stochastic algorithms is premature convergence. DE is not an exception. Any population based algorithm is regarded as an efficient algorithm if it is fast in convergence and able to explore the maximum area of the search space. In other words, if a population based algorithm is capable of balancing between exploration and exploitation of the search space, then the algorithm is regarded

an efficient algorithm. From this point of view, basic DE is not an efficient algorithm [12]. Also some studies proved that stagnation is another inherent drawback with DE i.e. DE sometimes stop proceeding toward the global optima even though the population has not converged to local optima or any other point [10]. Mezura-Montes et al. [12] compared the different variants of DE for global optimization and found that DE shows a poor performance and remains inefficient in exploring the search space, especially for multimodal functions. Price et al. [16] also drawn the same conclusions. The problems of premature convergence and stagnation is a matter of serious consideration for designing a comparatively efficient[1] DE algorithm.

3.2 Motivation for Modification in Scale Factor

This paper proposes a modification in scale factor to balance the perturbation rate in mutation process of Differential Evolution algorithm ($DSFDE$).

3.2.1 Dynamic Scale Factor

Exploration of the whole search space and exploitation of the near optimal solution region may be balanced by maintaining the diversity in early and later iterations of any random number based search algorithm. Mutation equation in DE which is shown as follows:

$$u_i(G) = x_i(G) + F \times (x_{i2}(G) - x_{i3}(G)) \tag{2}$$

i.e., the trial vector $u_i(G)$ is the weighted sum of target vector $x_i(G)$ and the difference $(x_{i2}(G) - x_{i3}(G))$ of two random vectors. Here, F is the weight to the difference of random vectors which controls the amplication of the differential variation. In basic DE, F is the scaling factor $F \in (0, \infty)$. Many studies have been carried out with varying scaling factor F [1] for better exploration and exploitation mechanism. Generally, the value of F is vary from (0, 1). The perturbation rate in ordinary DE remain uniform because scale factor which controls the amplication of the differential variation is suggested to be constant from $(0, \infty)$. However, the studies on Bio-Inspired algorithms suggest that these algorithms should explore more in early iterations and exploit more in latter iterations.

In this paper, we introduce the concept of Dynamic Scale Factor (DSF) which dynamically changes the step size with balanced expulsion and attraction of the differential variation for generating trial vector. For this purpose, initially we assign DSF, randomly a value in the range of [-0.8, 0.8] and then it decreases linearly till [-0.4, 0.4]. Now, It is clear from equation (2)

[1] As it is not possible to design a fully efficient population based stochastic algorithm

that for early iteration exploration capability of DE increases and later on exploitation capability increases. Hence, we can say that the use of DSF helps to maintain a proper balance of diversity and convergence in DE. DSF controls the production of neighbour potential solutions around x_i and represents the comparison of positions of two solutions. As can be seen from equation (2), as the difference between the parameters of the x_{i2} and x_{i3} decreases, the perturbation on the position x_i gets decreased, too. Thus, as the search approaches the optimum solution in the search space, the step length is adaptively reduced.

The Dynamic Scaling Factor based Differential Evolution ($DSFDE$) algorithm is similar to the basic DE algorithm except the range of scale factor in mutation operation. $DSFDE$ is a simple algorithm which, despite its simplicity, can be a very efficient possibility for optimization of various real world optimization problems.

3.3 Control Parameters in DSFDE

As stated by Storn et al. [15, 18], DE is very sensitive to the choice of F and CR. Some settings of control parameters are suggested by Stron in [15, 18]:

- $F = [0.5, 1]$;
- $CR \in [0.8, 1]$;
- $NP = [5D, 10D]$, where D is the number of decision variables in the problem.

$DSFDE$ introduces a dynamic scale factor (DSF) whose value varies randomly from range [-0.8 0.8] to [-0.4 0.4]. It provides more amplication of the differential variation in early iteration and less in letter iterations (as step size reduces gradually as number of iterations increases), for generating trial vector.

4 Experimental Results and Discussion

4.1 Test Problems under Consideration

In order to see the effect of Dynamic Scale Factor on DE, 10 scalable (the number of decision variables may be varied as per user's choice) test problems of optimization are selected (Listed in Table 1). These problems are of continuous variables and have different degree of complexity and multimodality. For this study, number of decision variables is set to 30.

Table 1. Test problems

S. No.	Test Problem	Objective function	Search Space	Dimension
1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	[-5.12 5.12]	30
2	De Jong f4	$f(x) = \sum_{i=1}^{n} i.(x_i))^4$	[-5.12 5.12]	30
3	Griewank	$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	[-600 600]	30
4	Rosenbrock	$f(x) = \sum_{i=1}^{n} (100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$	[-30 30]	30
5	Rastrigin	$f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12 5.12]	30
6	Ackley	$f(x) = -20 + e + exp(-\frac{0.2}{n} \sqrt{\sum_{i=1}^{n} {x_i}^3})$ $-exp(\frac{1}{n} \sum_{i=1}^{n} \cos\ (2\pi.x_i)x_i)$	[-1 1]	30
7	DropWave	$f(x) = -\frac{1+cos(\ 12\sqrt{\sum_{i=1}^{n} {x_i}^2})}{\frac{1}{2}\sum_{i=1}^{n} {x_i}^2+2}$	[-5.12 5.12]	30
8	Alpine	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin\ x_i + 0.1x_i \rvert$	[-10 10]	30
9	Michalewicz	$f(x) = -\sum_{i=1}^{n} \sin\ x_i (\sin\ (\frac{i.{x_i}^2}{\pi}))$	[0 π]	30
10	Cosine Mixture	$f(x) = \sum_{i=1}^{n} {x_i}^2 - 0.1(\sum_{i=1}^{n} \cos\ 5.\pi.x_i) + 0.1n$	[-1 1]	30

4.2 Experimental Setting for DSFDE

To test DE and $DSFDE$ over test problems following experimental setting is adopted:

- The crossover probability CR=0.33[7],
- The scale factor which controls the implication of the differential variation F=0.5; [15] and DSF=rand[-0.8, 0.8] to rand[-0.4, 0.4],

- Population size NP=100,
- The stopping criteria is either maximum number of generation which is set to be 2000 is reached or the objective function value ≤ 0.01; 0.01 is the minimum error criteria,
- The number of simulations =100,
- The number of decision variables in scalable test problems $D = 30$.

4.3 Comparison of DSFDE with DE

Numerical results with experimental setting of subsection 4.2 are tabulated in Table 2. In Table 2, success rate (SR) (a simulation is said to be successful if the objective function value is ≤ 0.01 in maximum 2000 generations), mean objective function value ($MOFV$), average function evaluations (AFE) and standard deviation (SD) is reported. Table 2 shows that most of the time use of dynamic scale factor in DE improves the reliability, efficiency and accuracy. Except for Rosenbrock problem $DSFDE$ improves the results over DE. Some more intensive statistical analysis based on t test, performance index and boxplots has been carried out for results of basic DE and $DSFDE$.

Table 2. Comparison of the results of test problems

Test Problem	**Algorithm**	**SR**	**AFE**	**MOFV**	**SD**
Sphere	DE	45	28200	0.00616	0.001101
	DSFDE	100	21693	0.005331	0.001355
De Jong	DE	97	29112.37109	0.003383	0.001476
	DSFDE	100	21673	0.003859	0.001888
Griewank	DE	47	50700	0.006073	0.001162
	DSFDE	100	36901	0.005265	0.001319
Rosenbrock	DE	0	200000	23.63907	0.378742
	DSFDE	0	200000	29.581992	14.708954
Rastrigin	DE	0	200000	69.516951	6.304635
	DSFDE	99	85755.55469	0.015099	0.098487
Ackley	DE	0	200000	0.014171	0.002078
	DSFDE	100	36811	0.007205	0.000871

Table 2. (*continued*)

Test Problem	Algorithm	SR	AFE	MOFV	SD
DropWave	DE	100	270	0.000001	0.000002
	DSFDE	100	243	0.000001	0.000002
Alpine	DE	0	200000	0.017717	0.002948
	DSFDE	100	54401	0.00686	0.001071
Michalewicz	DE	100	25219	0.000312	0.000517
	DSFDE	100	23356	0.000332	0.000691
Cosine Mixture	DE	60	28325	0.006254	0.001346
	DSFDE	100	20059	0.005194	0.001104

4.3.1 Statistical Analysis

In this section, statistical comparison is carried out between DE and $DSFDE$, using t-test, boxplots and performance index [5].

The t-test is quite popular among researchers in the field of evolutionary computing. In this paper students t-test is applied according to the description given in [3] for a confidence level of 0.95. Table 3 shows the results of the t-test for the null hypothesis that there is no difference in the mean number of function evaluations of 100 runs using DE and DSFDE. Note that here '+' indicates the significant difference (or the null hypothesis is rejected) at a 0.05 level of significance, '-' implies that there is no significant difference while '=' indicates that comparison is not possible. It is observed from Table 3 that significant differences observed in 7 comparisons out of 10 comparisons. Therefore, it can be concluded that results of $DSFDE$ is significantly different from the basic DE algorithm. For the purpose of comparison in terms of performance, Boxplot analysis is carried out for all the considered algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [20]. The Boxplots for average function evaluation of DE and $DSFDE$ are shown in Figure 1. It is clear from this figure that $DSFDE$ is better than DE as Interquartile range and Median are low for it. In order to compare the consolidated performance of DE with $DSFDE$, the

Table 3. Results of the Students t test with DSFDE

S. No.	Test Problem	DE	S. No.	Test Problem	DE
1	Sphere	+	6	Ackley	+
2	De Jong f4	+	7	DropWave	-
3	Griewank	+	8	Alpine	+
4	Rosenbrock	=	9	Michalewicz	-
5	Rastrigin	+	10	Cosine Mixture	+

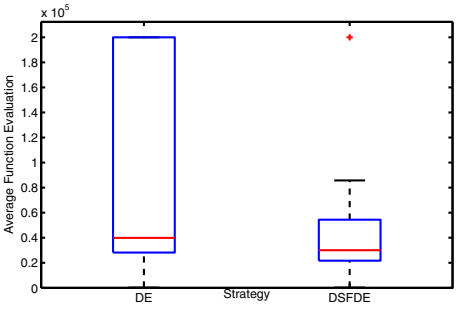


Fig. 1. Boxplot graph for Average Function Evaluation: (1) DE (2) $DSFDE$

value of a performance index PI [5] is computed. This index gives a weighted importance to following parameters.

1. Success rate,
2. Average number of function evaluations,
3. Mean objective function value.

The graphs corresponding to each of the cases (1), (2) and (3) are shown in Figures 2(a), 2(b), and 2(c) respectively. In these figures the horizontal axis represents the weight W and the vertical axis represents the performance index PI. In case (1), average number of function evaluations of successful runs and the average objective function value are given equal weights. PIs of both the algorithms $(DSFDE, DE)$ are superimposed in the Figure 2(a) for comparison and to get a ranking of the performance of both the algorithms. It is observed that PI value of $(DSFDE)$ is more than DE and perform in the order $DSFDE > DE$. In case (2), equal weights are assigned to the success rate and average function evaluations of successful runs. From Figure 2(b), same conclusion is drawn as in case (1). In case (3), equal weights are assigned to the success rate and average objective function value. Again the same

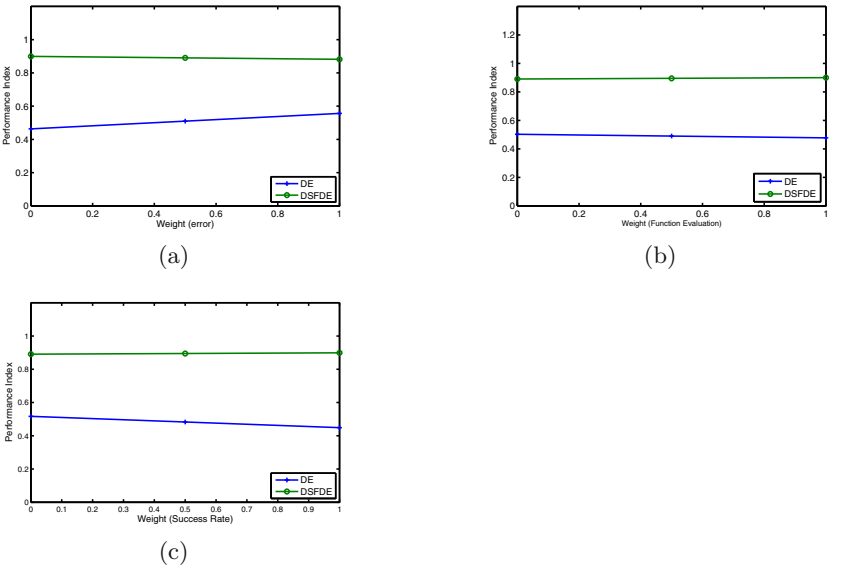


Fig. 2. Performance index; (a) for case (1), (b) for case (2) and (c) for case (3).

results are found through Figure 2(c) as in 2(a) and 2(b). Therefore, it can be stated that the $DSFDE$ is perform better than the basic DE.

5 Conclusion

In this paper, basic differential evolution algorithm is improved by introducing Dynamic Scaling Factor in DE search procedure and the proposed strategy is named as Dynamic Scaling Factor based Differential Evolution ($DSFDE$) Algorithm. With the help of experiments over test problems it is showed that the reliability (due to success rate), efficiency (due to average number of function evaluations) and accuracy (due to mean objective function value) of DSFDE algorithm with this modified scaling factor is higher than that of its original versions. Based on this study, it is concluded that $DSFDE$ is a better candidate in the field of nature inspired algorithms for function optimization.

The future scope of this work is the implementation of dynamic scaling factor to other biologically inspired algorithms.

Acknowledgement. Harish Sharma acknowledges ABV- Indian Institute of Information Technology and Management Gwalior for providing research grant to carry out this work.

References

[1] Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation 10(6), 646–657 (2006)
[2] Chakraborty, U.K.: Advances in differential evolution. Springer, Heidelberg (2008)
[3] Croarkin, C., Tobias, P.: Nist/sematech e-handbook of statistical methods (2010) (retrieved March 1, 2010)
[4] Das, S., Konar, A.: Two-dimensional iir filter design with modern search heuristics: A comparative study. International Journal of Computational Intelligence and Applications 6(3), 329–355 (2006)
[5] Thakur, M., Deep, K.: A new crossover operator for real coded genetic algorithms. Applied Mathematics and Computation 188(1), 895–911 (2007)
[6] Engelbrecht, A.P.: Computational intelligence: an introduction. Wiley (2007)
[7] Gamperle, R., Muller, S.D., Koumoutsakos, A.: A parameter study for differential evolution. Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation 10, 293–298 (2002)
[8] Holland, J.H.: Adaptation in natural and artificial systems (1975)
[9] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
[10] Lampinen, J., Zelinka, I.: On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL, Citeseer, pp. 76–83 (2000)
[11] Liu, P.K., Wang, F.S.: Inverse problems of biological systems using multiobjective optimization. Journal of the Chinese Institute of Chemical Engineers 39(5), 399–406 (2008)
[12] Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 485–492. ACM (2006)
[13] Omran, M.G.H., Engelbrecht, A.P., Salman, A.: Differential evolution methods for unsupervised image classification. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 966–973. IEEE (2005)
[14] Onwubolu, G.C., Sharma, A.: Intrusion detection system using hybrid differential evolution and group method of data handling approach
[15] Price, K.V.: Differential evolution: a fast and simple numerical optimizer. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS, pp. 524–527. IEEE (1996)
[16] Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)
[17] Rogalsky, T., Kocabiyik, S., Derksen, R.W.: Differential evolution in aerodynamic optimization. Canadian Aeronautics and Space Journal 46(4), 183–190 (2000)

[18] Storn, R., Price, K.: Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute-Publications-TR (1995)
[19] Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Congress on Evolutionary Computation, CEC 2004, vol. 2, pp. 1980–1987. IEEE (2004)
[20] Williamson, D.F., Parker, R.A., Kendrick, J.S.: The box plot: a simple visual method to interpret data. Annals of Internal Medicine 110(11), 916 (1989)

Performance Improvement in Vector Control of Induction Motor Drive Using Fuzzy Logic Controller

P. Tripura[1] and Y. Srinivasa Kishore Babu[2]

[1] Department of EEE, Vignan's Nirula Institute is Technology of Science for Women, Guntur, A.P., India
tripura.pidikiti@gmail.com
[2] School of Electrical Engineering, Vignan University, Vadlamudi, A.P.
yskbabu@gmail.com

Abstract. This paper presents an intelligent speed control system based on fuzzy logic for a voltage source PWM inverter-fed indirect vector controlled induction motor drive. Traditional indirect vector control system of induction motor introduces conventional PI regulator in outer speed loop; it is proved that the low precision of the speed regulator debases the performance of the whole system. To overcome this problem, replacement of PI controller by an intelligent controller based on fuzzy set theory is proposed. The performance of the intelligent controller has been investigated through digital simulation using MATLAB-SIMULINK package for different operating conditions such as sudden change in reference speed and load torque. The simulation results demonstrate that the performance of the proposed controller is better than that of the conventional PI controller.

Keywords: Fuzzy Logic, Intelligent controllers, Conventional PI controller, Induction motor drives, indirect vector control, Speed control.

1 Introduction

For electrical drives good dynamic performance is mandatory so as to respond to the changes in command speed and torques. These requirements of AC drives can be fulfilled by the vector control system. With the advent of the vector control method, an induction motor has been controlled like a separately excited DC motor for high performance applications. This method enables the control of field and torque of induction motor independently (decoupling) by manipulating corresponding field oriented quantities [1], [2].

The traditional indirect vector control system uses conventional PI controller in the outer speed loop because of the simplicity and stability. However, unexpected change in load conditions or environmental factors would produce overshoot, oscillation of motor speed, oscillation of the torque, long settling time and thus

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 87–97.
springerlink.com

causes deterioration of drive performance. To overcome this, an intelligent controller based on Fuzzy Logic can be used in the place of PI regulator [4]. The fuzzy logic has certain advantages compared to classical controllers such as simplicity of control, low cost, and the possibility to design without knowing the exact mathematical model of plant [3].

In this paper application of fuzzy logic to the intelligent speed control of indirect vector controlled induction motor drive is investigated. The analysis, design and simulation of controller have been carried out based on the fuzzy set theory.

When a new control strategy of a converter or a drive system is formulated, it is often convenient to study the system performance by simulation before building the breadboard or prototype. The simulation not only validates the systems operation, but also permits optimization of the systems performance by iteration of its parameters. Besides the control and circuit parameters, the plant parameter variation effect can be studied. Valuable time is thus saved in the development and design of the product, and the failure of components of poorly designed systems can be avoided. The simulation program also helps to generate real time controller software codes for downloading to a microprocessor or digital signal processor.

Many circuit simulators like PSPICE, EMTP, MATLAB/ SIMULINK incorporated these features. The advantages of SIMULINK over the other circuit simulator are the ease in modeling the transients of electrical machines and drives and to include controls in the simulation. To solve the objective of this paper MATLAB/ SIMULINK software is used. The superior control performance of the proposed controller is demonstrated at SIMULINK platform using the fuzzy logic tool box [5] for different operating conditions.

The complete paper is organized as follows: Section 2 describes the indirect vector control system. The design and description of intelligent controller is provided in section 3. The simulation results, comparison and discussion are presented in Section 4. Section 5 concludes the work.

2 Indirect Vector Control System

For the high performance drives, the indirect method of vector control is preferred choice [1], [2]. The indirect vector control method is essentially same as the direct vector control, except that the rotor angle θ_e is generated in an indirect manner (estimation) using the measured speed ω_r and the slip speed ω_{sl}. To implement the indirect vector control strategy, it is necessary to take the following dynamic equations into consideration.

$$\theta_e = \int \omega_e dt = \int (\omega_r + \omega_{sl}) dt = \theta_r + \theta_{sl} \tag{1}$$

For decoupling control, the stator flux component of current i_{ds} should be aligned on the d^e axis, and the torque component of current i_{qs} should be on q^e axis, that leads to $\psi_{qr} = 0$ and $\psi_{dr} = \psi_r$ then:

$$\frac{L_r}{R_r}\frac{d\psi_r}{dt}+\psi_r = L_m i_{ds} \tag{2}$$

As well, the slip frequency can be calculated as:

$$\omega_{sl} = \frac{L_m}{\psi_r}\frac{R_r}{L_r} i_{qs} = \frac{R_r}{L_r}\frac{i_{qs}}{i_{ds}} \tag{3}$$

It is found that the ideal decoupling can be achieved if the above slip angular speed command is used for making field-orientation. The constant rotor flux ψ_r and $\frac{d\psi_r}{dt}=0$ can be substituted in equation (2), so that the rotor flux sets as

$$\psi_r = L_m i_{ds} \tag{4}$$

The Simulink model for such an indirect vector control system is shown in the Fig. 3. This control technique operates the induction motor as separately excited DC motor so as to achieve high dynamic performance [1], [2].

3 Design and Description of Intelligent Controller

Since the implementation of off-line tuning of PI controller is difficult in dealing with continuous parametric variation in the induction motor as well as the non-linearity present in the entire system, it becomes of interest to go for intelligent controller. It is known that the stator and rotor resistances of induction motor may change with the temperature up to 50% and motor inductance varies with the magnetic operating point. Furthermore, the load torque may change due to mechanical disturbances.

The problem can be solved by several adaptive control techniques such as model reference adaptive control, sliding-mode control, variable structure control, and self-tuning PI controllers, etc. The theory and survey on model reference adaptive system has been reported by H. Sugimoto et.al [6]. Secondary resistance identification of an IM applied with MRAS and its characteristics has been presented in their study. The improved version of sliding mode control for an IM has been proposed by C. Y. Won et.al [7]. The design of integral variable structure control system for servo systems has been proposed by T. L. Chern et.al [8]. The self tuning controllers are described by J. C. Hung [9]. However, in all these works, exact mathematical model of the system is mandatory to design the adaptive control algorithm. Thus they increase the complexity of design and implementation.

When fuzzy logic bases intelligent controller is used instead of the PI controller, excellent control performance can be achieved even in the presence of parameter variation and drive non-linearity [1], [3].

The vector control of IM with fuzzy PI controller has been proposed by I. Miki et. al [10] and W. P. Hew et.al [11]. As they reported, the FLC automatically

updates the proportional and integral gains on-line and thus help in achieving fast dynamic response. However, this technique does not fully utilize the capabilities of the fuzzy logic. Moreover, the inherent disadvantages associated with the PI controller cannot be avoided. The fuzzy PI controllers are less useful in industrial applications.

The performances of the fuzzy logic based indirect vector control for induction motor drive has been proposed by M. N. Uddin et.al [12], E. Cerruto et.al [13], B. Heber et.al [14], and G. C. D. Sousa et.al [15]. The novel speed control for current regulated VSI-fed IM has been discussed by them. The fuzzy logic based controller for IM drives has been proposed by Minh Ta-Cao et.al [16]. The performance of the proposed system is compared with the conventional vector control on the basis of Integral of time by Absolute Time Error (IATE).

The Simulink implementation of current regulated VSI-fed IM is proposed by Norman Mariun et.al [17] and Vinod Kumar et.al [18]. They proposed a fuzzy logic controller in place of PI controller in the vector control system. However, the power system block set used by them makes use of S-functions and it is not as easy to work with as the rest of the Simulink blocks.

The work presented in [12]-[18] uses a fuzzy logic controller to set the torque component of reference current based on speed error and change of speed error. The inverter is then switched to follow the reference current within hysteresis band. However, the constant hysteresis band of the current regulated PWM inverter of the fuzzy logic based indirect vector control system possesses problem in achieving superior dynamic performance, even the drive control system includes the efficient fuzzy logic controller. This paper discusses the fuzzy logic speed control for VSI fed indirect vector controlled induction motor drives.

Fig. 1 shows the block diagram of fuzzy logic based speed control system. Such a fuzzy logic controller consists of four basic blocks viz., Fuzzification, Fuzzy Inference Engine, Knowledge base and defuzzification.

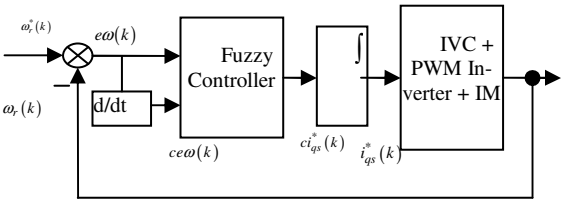


Fig. 1. Block diagram of Fuzzy logic speed control system for indirect vector controlled induction motor drive

3.1 Input/ Output Variables

The design of the fuzzy logic controller starts with assigning the input and output variables. The most significant variables entering the fuzzy logic speed controller has been selected as the speed error and its time variation. Two input variables $e\omega(k)$ and $ce\omega(k)$, are calculated at every sampling instant as:

$$e\omega(k) = \omega_r^*(k) - \omega_r(k) \tag{5}$$

$$ce\omega(k) = e\omega(k) - e\omega(k-1) \tag{6}$$

Where $\omega_r^*(k)$ is the reference speed, $\omega_r(k)$ is the actual rotor speed and $e\omega(k-1)$ is the value of error at previous sampling time.

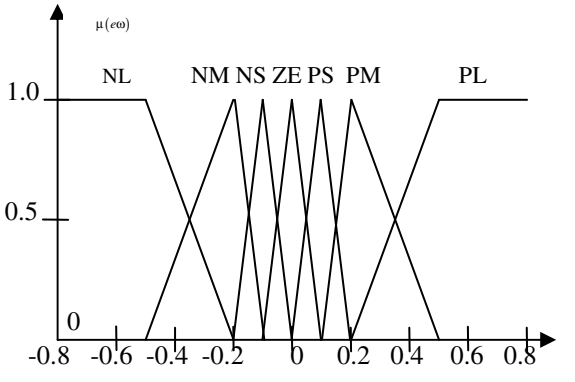
The output variable of the fuzzy logic speed controller is the variation of command current, $ci_{qs}^*(k)$ which is integrated to get the reference command current, $i_{qs}^*(k)$ as shown in the following equation.

$$i_{qs}^*(k) = i_{qs}^*(k-1) + ci_{qs}^*(k) \tag{7}$$

3.2 Fuzzification

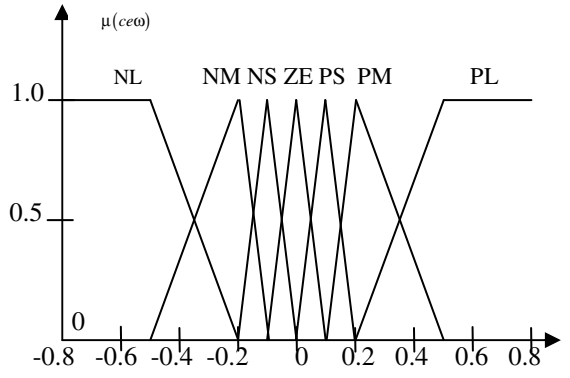
The success of this work, and the like, depends on how good this stage is conducted. In this stage, the crisp variables $e\omega(k)$ and $ce\omega(k)$ are converted in to fuzzy variables $e\omega$ and $ce\omega$ respectively. The membership functions associated to the control variables have been chosen with triangular shapes as shown in Fig. 2.

The universe of discourse of all the input and output variables are established as (-0.8, 0.8). The suitable scaling factors are chosen to brought the input and output variables to this universe of discourse. Each universe of discourse is divided into seven overlapping fuzzy sets: NL (Negative Large), NM (Negative Medium), NS (Negative Small), ZE (Zero), PS (Positive Small), PM (positive Medium), and PL

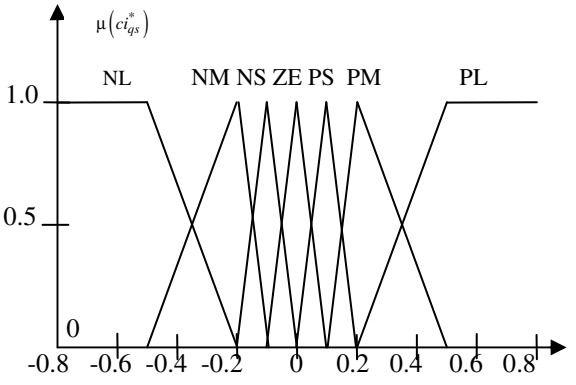


(a)

Fig. 2. Membership functions for (a) speed error (b) change of speed error (c) Change of command current



(b)



(c)

Fig. 2. (*continued*)

(Positive Large). Each fuzzy variable is a member of the subsets with a degree of membership μ varying between 0 (non-member) and 1 (full-member). All the membership functions have asymmetrical shape with more crowding near the origin (steady state). This permits higher precision at steady state [3].

3.3 Knowledge Base and Inference Stage

Knowledge base involves defining the rules represented as IF-THEN statements governing the relationship between input and output variables in terms of membership functions. In this stage, the variables $e\omega$ and $ce\omega$ are processed by an inference engine that executes 49 rules (7x7) as shown in Table I. These rules

are established using the knowledge of the system behavior and the experience of the control engineers. Each rule is expressed in the form as in the following example: IF ($e\omega$ is Negative Large) AND ($ce\omega$ is Positive Large) THEN (ci_{qs}^{*} is Zero). Different inference engines can be used to produce the fuzzy set values for the output fuzzy variable ci_{qs}^{*}. In this paper, the Max-product inference method [3] is used.

Table 1. Fuzzy Control Rules

ce \ *e*	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NL	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	NM	NS	ZE	PS	PM
ZE	NL	NM	NS	ZE	PS	PM	PL
PS	NM	NS	ZE	PS	PM	PL	PL
PM	NS	ZE	PS	PM	PL	PL	PL
PL	ZE	PS	PM	PL	PL	PL	PL

3.4 Defuzzification

In this stage a crisp value of the output variable $ci_{qs}^{*}(k)$ is obtained by using height defuzzufication method, in which the centroid of each output membership function for each rule is first evaluated. The final output is then calculated as the average of the individual centroid, weighted by their heights (degree of membership) as follows:

$$ci_{qs}^{*}(k) = \frac{\sum_{i=1}^{n} \mu\left[\left(ci_{qs}^{*}\right)_i\right]\left(ci_{qs}^{*}\right)_i}{\sum_{i=1}^{n} \mu\left[\left(ci_{qs}^{*}\right)_i\right]} \tag{8}$$

The reference value of command current $i_{qs}^{*}(k)$ that is applied to vector control system is computed by the equation (7).

The overall model for fuzzy logic based speed control system for indirect vector controlled induction motor drive is shown in Fig. 3. The parameters of the motor are given in appendix.

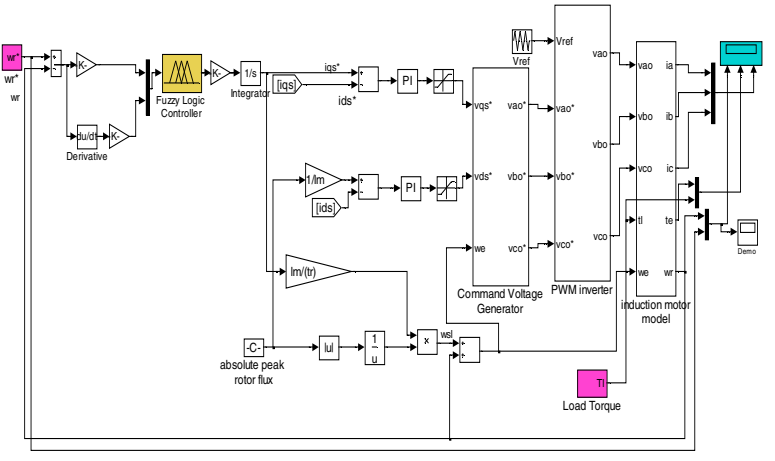


Fig. 3. Indirect vector controlled induction motor diagram with the Fuzzy Logic Controller

4 Simulation Results and Discussion

A series of simulation tests were carried out on indirect vector controlled induction motor drive using both the PI controller and fuzzy logic based intelligent controller for various operating conditions. The time response and steady state errors were analyzed and compared.

Figures 4 and 5 shows speed response with both the PI and FL based controller. The FL controller performed better performance with respect to rise time and steady state error.

Figure 6 shows the load disturbance rejection capabilities of each controller when using a step load from 0 to 20 N-m at 0.8 seconds. The FL controller at that moment returns quickly to command speed, where as the PI controller maintains a steady state error.

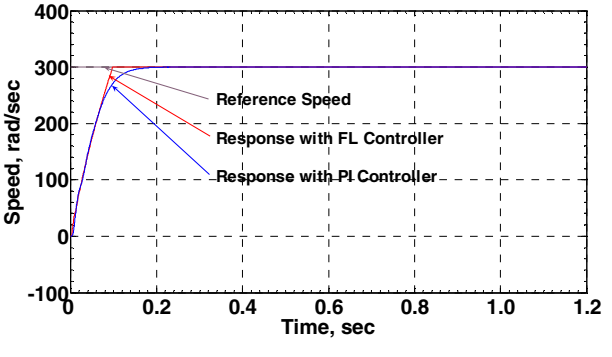


Fig. 4. Speed response comparison at no-load

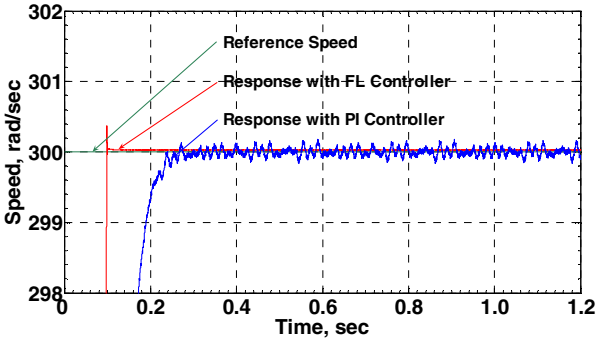


Fig. 5. Enlarged speed response comparison at no-load

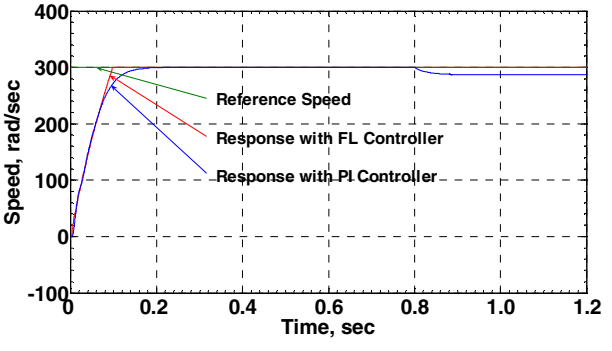


Fig. 6. Speed response comparison during sudden load change

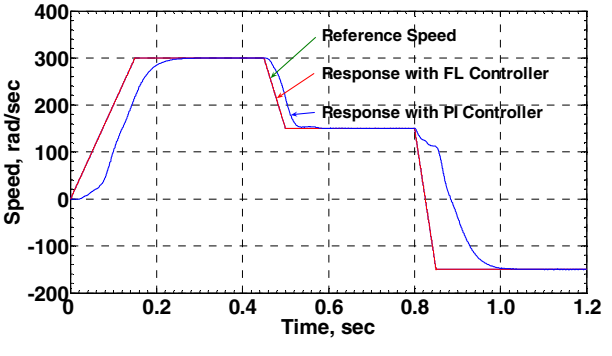


Fig. 7. Speed tracking response comparison

Figure 7 shows the speed tracking performance test, when sudden change in speed reference is applied in the form of look-up table. The intelligent controller exhibited better speed tracking compared to PI controller.

5 Conclusions

The performance of fuzzy logic based intelligent controller for the speed control of indirect vector controlled, PWM voltage source inverter fed induction motor drive has been verified and compared with that of conventional PI controller performance. The simulation results obtained have confirmed the very good dynamic performance and robustness of the fuzzy logic controller during the transient period and during the sudden loads. It is concluded that the proposed intelligent controller has shown superior performance than that of the parameter fixed PI controller and earlier proposed system [4].

Appendix

3-Phase Induction Motor Parameters
Rotor type: Squirrel cage,
Reference frame: Synchronous
10 hp, 314 rad/sec, 4 Poles, $R_s = 0.19\ \Omega$, $R_r = 0.39\ \Omega$, $L_{ls} = 0.21\text{e-}3$ H, $L_{lr} = 0.6\text{e-}3$ H, $L_m = 4\text{e-}3$ H, $J = 0.0226$ Kg-m^2.

References

[1] Bose, B.K.: Modern Power Electronics and AC Drives, Third impression. Pearson Education, Inc., India (2007)
[2] Blaschke, F.: The Principle of Field-Orientation as applied to the New Transvector Closed-Loop Control System for Rotating-Field Machines. Siemens Review 34, 217–220 (1972)
[3] Lee, C.C.: Fuzzy Logic in Control Systems: Fuzzy Logic Control – Part 1. IEEE Transactions on Systems, Man and Cybernetics 20(2), 404–418 (1990)
[4] Uddin, M.N., Radwan, T.S., Rahman, M.A.: Performances of Fuzzy-Logic-Based Indirect Vector Control for Induction Motor Drive. IEEE Transactions on Industry Applications 38(5), 1219–1225 (2002)
[5] http://www.mathworks.com/ (The official site for MATLAB&SIMULINK as well as Fuzzy Logic Toolbox)
[6] Sugimoto, H., Tamai, S.: Secondary resistance identification of an Induction Motor Applied Model reference Adaptive Systems and its Characteristics. IEEE Trans. on Ind. Appl. IA-23(1), 296–303 (1987)
[7] Won, C.Y., Bose, B.K.: An induction Motor servo Systems with Improved Sliding Mode Control. In: Proc. IEEE IECON 1992, pp. 60–66 (1992)
[8] Chern, T.L., Wu, Y.C.: Design of Integral Variable Structure Controller and Applications to Electro Hydraulic Velocity Servo Systems. Proc. In Elec. Eng. 138(5), 439–444 (1991)
[9] Hung, J.C.: Practical Industrial Control techniques. In: Proc. IEEE IECON 1994, pp. 7–14 (1994)
[10] Miki, I., Nagai, N., Nishigama, S., Yamada, T.: Vector control of induction motor with fuzzy PI controller. In: IEEE IAS Annu. Meet. Conf. Rec., pp. 342–346 (1991)

[11] Hew, W.P., Tamjis, M.R., Saddique, S.M.: Application of Fuzzy Logic in Speed Control of Induction Motor Vector Control. In: Proc. of the International Conference on Robotics, Vision and Parallel Processing for Industrial Automation, Ipoh, Malasiya, November 28-30, pp. 767–772 (1996)

[12] Uddin, M.N., Radwan, T.S., Rahman, M.A.: Performances of Fuzzy-Logic-Based Indirect Vector Control for Induction Motor Drive. IEEE Transactions on Industry Applications 38(5), 1219–1225 (2002)

[13] Cerruto, E., Consoli, A., Raciti, A., Testa, A.: Fuzzy Adaptive Vector Control of Induction Motor Drives. IEEE Trans. on Power Electronics 12(6), 1028–1039 (1997)

[14] Hebetler, B., Xu, L., Tang, Y.: Fuzzy Logic Enhanced Speed Control of Indirect Field Oriented Induction Machine Drive. IEEE Trans. On Power Electronics 12(5), 772–778 (1997)

[15] Sousa, G.C.D., Bose, B.K., Cleland, J.G.: Fuzzy Logic based On-Line Efficiency Optimization Control of an Indirect Vector Controlled Induction Motor Drive. IEEE Trans. On Industrial Electronics 42(2), 192–198 (1995)

[16] Ta-Cao, M., Silva Neto, J.L., Le-Huy, H.: Fuzzy Logic based Controller for Induction Motor Drives. In: Canadian Conference on Electrical and Computer Engineering, May 26-29, vol. 2, pp. 631–634 (1996)

[17] Mariun, N., Noor, S.B.M., Jasni, J., Bennanes, O.S.: A Fuzzy Logic based Controller for an Indirect Vector Controlled Three-Phase Induction Motor. In: IEEE Region 10 Conference, TENCON 2004, November 21-24, vol. 4D, pp. 1–4 (2004)

[18] Kumar, V., Joshi, R.R.: Hybrid Controller based Intelligent Speed Control of Induction Motor. Journal of Theoretical and Applied Information Technology 3(1), 71–75 (2006)

A Fuzzy Particle Swarm Optimization for Solving the Economic Dispatch Problem

Sanjeev Kumar and D.K. Chaturvedi

Dayalbagh Educational Institute, Agra, India
{sanjeev.85ee,dkc.foe}@gmail.com

Abstract. This paper presents a fuzzy particle swarm optimizer to solve the economic load dispatch (ELD) problem of thermal generators of a power system. Several factors such as quadratic cost functions with valve point loading is considered in the computation models. The Fuzzy particle swarm optimization (FPSO) provides a new mechanism to avoid premature convergence problem with optimum solution. The proposed method has been applied to 3 and 40 generator power system whose cost functions are non-convex in nature. Results obtained by this method have been compared with those obtained by PSO method. The experimental results show that proposed FPSO method is capable of obtaining optimum solution in fewer numbers of iterations.

1 Introduction

Economic Load Dispatch (ELD) seeks "the best" generation schedule for the generating plants to supply the required demand plus transmission losses with the minimum production cost. Economic load dispatch is a method to schedule power generator outputs with respect to the load demands and to operate a power system economically, so as to minimize the operation cost of the power system.

Previously a number of conventional approaches [1] – [15] such as *gradient method, linear programming algorithm, lambda iteration method, quadratic programming, non-linear programming algorithm, Lagrange relaxation algorithm* etc have been applied for solving the ELD problems [1]. These traditional classical dispatch algorithm require incremental cost curves that are monotonically increasing/ piece-wise linear in nature. But normally the input output characteristics of modern generating units are highly non-linear in nature due to valve-point effects, ramp-rate limits etc having multiple local minimum points in the cost function. Thus the characteristics of ELD problems are multimodal, discontinuous and highly nonlinear. Due to such approximation the solution is subnormal and hence a huge amount of revenue loss occurs over the time. To overcome these problem soft computing techniques came in existence. The soft computing techniques are super set of fuzzy, artificial neural network [16]-[20],

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 99–110.
springerlink.

evolutionary technique, genetic algorithm [21]-[26], particle swarm optimization [27]-[42], simulated annealing [43] and other hybrid techniques.

2 Problem Description

The objective of economic dispatch is to minimize the total generation cost of a power system over some appropriate period while satisfying various constraints.

The most simplified cost function of each generator can be represented as a quadratic function as given below:

Minimize fuel cost

$$C = \sum_{i=1}^{NG} F(P_i) \tag{1}$$

$$F(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{2}$$

Inequality constraint

$$P_{i\,min} \leq P_i \leq P_{i\,max} \tag{3}$$

Equality Constraint

$$\sum_{i=1}^{NG} P_i = P_d \quad (i = 1, 2, 3, \ldots \ldots NG) \tag{4}$$

After including equality constraint fuel cost function become

$$C = \sum_{i=1}^{NG} F(P_i) + PF * \left(\sum P_i - P_d - P_l \right) \tag{5}$$

Where

P_i = Generated power of ith plant (MW)
P_d = Total power demand (MW)
P_l = power loss (MW)
$P_{i\,min}$ = Minimum power of ith plant (MW)
$P_{i\,max}$ = Maximum power of ith plant (MW)
PF = Penalty Factor
NG = Total number of generators
a_i, b_i, c_i, = generator constant

The generator with multi-valve steam turbines has very different input-output curve compared with the smooth cost function.

Fuel cost function with valve point loading

$$F(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i sin\left(f_i(P_{i\,min} - P_i)\right)\right| \quad (6)$$

Where

e_i and f_i are valve point constant.

3 Particle Swarm Optimization

PSO is invented by Eberhart and Kennedy in 1995. It is inspired by natural concepts such as bird flocking and fish schooling. The basic PSO algorithm consists of three steps:

1. Generating particle' positions and velocities
2. Velocity update
3. Position update.

First, the position, x_k^i and velocities, v_k^i of the initial swarm of particles are randomly generated using upper and lower bounds on the design variables values, x_{min} and x_{max}. The positions and velocities are given in the vector format with the subscript and superscript denoting ith particle at the k[th] time instant.

The second step is to update the velocities of all the particles at (k+1)[th] time instant using particle objective or fitness values which are functions of the particles current positions in the design space at time k. Description of velocity updates and positions are given in Figure 1. The fitness function value of a particles determines which particle has the best global value in the current swarm, p_k^g and also determines the best position of each particle over time, p^i, i.e. in current and all previous moves. The velocity update formula uses these two pieces of information for each particle in the swarm along with the effect of current motion, v_k^i, to provide a search direction, v_{k+1}^i, for the next iteration.

$$\boldsymbol{v_{k+1}^i = A_1 + A_2 + A_3} \quad (7)$$

where

Current motion

$$\boldsymbol{A_1 = \omega * v_k^i} \quad (8)$$

Particle memory influence

$$\boldsymbol{A_2 = c_1 rand \frac{\left(p^i - x_k^i\right)}{\Delta t}} \quad (9)$$

Swarm influence

$$\boldsymbol{A_3 = c_2 rand \frac{\left(p_k^g - x_k^i\right)}{\Delta t}} \quad (10)$$

The three values that effect the new search direction, namely, current motion, particle memory influence and swarm influence, are incorporated via a summation approach as shown in above equation with three factor, namely, inertia factor (ω), self confidence factor (c_1) and swarm confidence factor (c_2) respectively.

Position update is the last step in each iteration. The position of each particle is updated using its velocity vector given below

$$\boldsymbol{x_{k+1}^i = x_k^i + x_{k+1}^i \, \Delta t} \tag{11}$$

The three steps of velocity update, position update and fitness calculations are repeated until a desired convergence criterion is met.

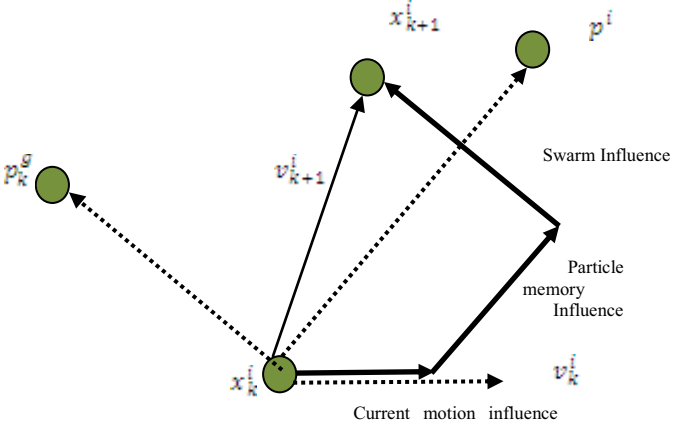


Fig. 1. Depiction of the velocity and position update in PSO

4 Fuzzy Particle Swarm Optimization

Knowing the fact that, tuning PSO parameters, (c_1, c_2, ω), can considerably affect the algorithm performance. There have been many attempts to adjust the PSO inertia weight parameter. In paper [47], authors proposed a new kind of Fuzzy PSO which has the better convergence performance accuracy and faster evolution velocity by measuring the distance of particle.

In this paper one input variable and one output variable are taken for Fuzzy Logic Controller (FLC). The input variable is iteration and output variable is inertia weight. The schematic diagram of FPSO is shown in Figure 2.

An FLC is composed of knowledge base, that includes the information given by fuzzy rules, a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets, an inference system, that uses them together with the knowledge base to make inference by means of a reasoning method, and the Defuzzification interface, that translate the fuzzy control action into crisp output [48, 49].

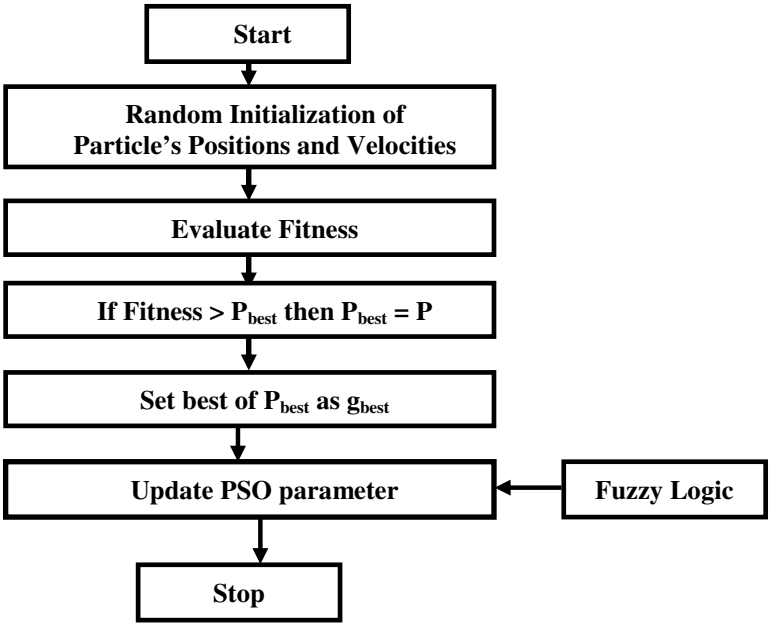


Fig. 2. Schematic diagram of fuzzy particle swarm optimization

5 Simulated Result Analysis

To access the efficiency and applicability of proposed FPSO for economic load dispatch has been tested on two different power system cases. The obtained results are compared with PSO and reported results of IEP [45], EPSO [45] methods.

5.1 Three Unit System

The system contains three thermal generating units. The total load demand on the system is 850 MW. The results of three unit system without valve point loading are reported in Table 1. The result of FPSO is better than PSO because the convergence is very fast than PSO as given in Figure 3.

The results of three unit system with valve point loading are reported in Table 2. The result of FPSO is better than PSO because the convergence is very fast than PSO as given in Figure 4.

Table 1. Results for three- unit system for demand of 850 MW without valve point loading

Generator Power Output	Lambda Iteration	PSO	FPSO	IEP[45]	EPSO[45]
P_1	393.17	393.158257	393.188342	393.170	393.16982
P_2	334.60	334.623034	334.581670	334.603	334.60375
P_3	122.23	122.218707	122.229986	122.227	122.22642
Total power generation	850.00	850.00	850.00	850.00	850.00
Minimum Fuel cost ($/h)	8194.36	8194.356121	8194.356121	8194.35614	8194.356121

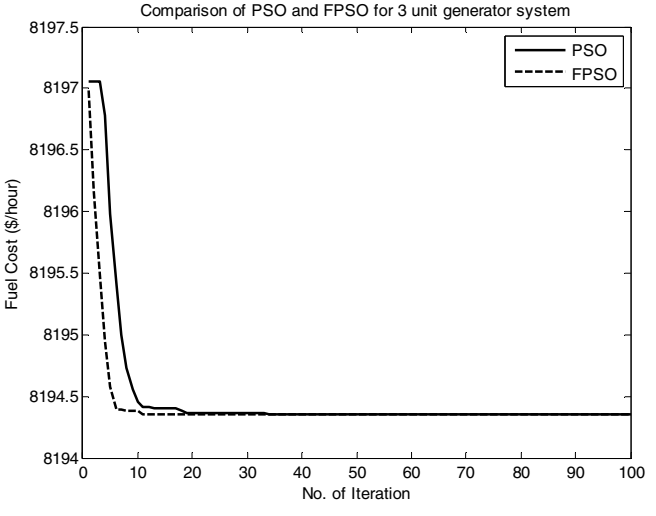


Fig. 3. Convergence plot of three unit system without valve point loading

Table 2. Results for three- unit system for demand of 850 MW with valve point loading

Method	Lambda Iteration	PSO	FPSO	GA[45]	IEP[45]	EPSO[45]
P_1	393.17	349.4830	349.5397	300.00	300.23	300.26449
P_2	334.60	400.5061	400.5759	400.00	400	400.00
P_3	122.23	100.0107	99.8842	150.00	149.77	149.73550
Total power generation	850.00	850.00	850.00	850.00	850.00	850.00
Minimum Fuel cost ($/h)	8719.87	8220.6361	8220.0599	8237.60	8234.09	8234.0730

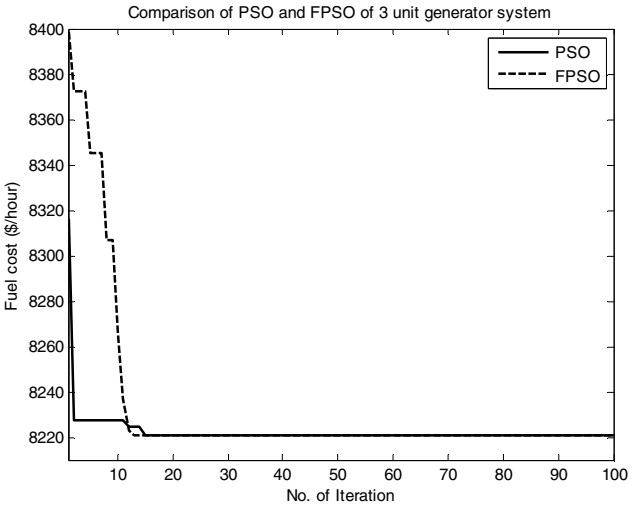


Fig. 4. Convergence plot of three unit system with valve point loading

5.2 Forty Unit System

The system contains forty thermal generating units. The total load demand on the system is 10500 MW. The results of three unit system without valve point loading are reported in Table 3. The result of FPSO is better than PSO because the convergence is very fast than PSO as given in Figure 5.

Table 3. Results for forty- unit system for demand of 10500 MW with valve point loading

Unit	PSO	FPSO	EPSO[45]	Unit	PSO	FPSO	EPSO[45]
1	110.87	96.7	114.00	**21**	450.1	343.77	550.00
2	111.58	185.6	114.00	**22**	523.52	433.5	550.00
3	208.31	172.2	60.00	**23**	434.81	433.52	550.00
4	129.88	179.73	190.00	**24**	345.75	523.28	550.00
5	171.61	169.38	97.00	**25**	525.21	343.76	550.00
6	182.1	142.77	140.00	**26**	434.93	433.52	550.00
7	259.65	334.37	300.00	**27**	9.10	10.00	10.00
8	211.22	284.6	300.00	**28**	10.12	9.55	10.00
9	284.81	284.6	290.1619	**29**	9.78	10.00	10.00
10	204.15	273.39	130.00	**30**	129.13	128.59	97.00
11	242.17	240.87	94.00	**31**	159.55	259.47	190.00
12	244.49	310.28	94.00	**32**	262.08	159.74	190.00

Table 3. (*continued*)

13	303.79	394.18	125.00	**33**	263.88	259.47	190.00
14	486.35	393.2	125.00	**34**	240.26	164.79	200.00
15	396.56	214.76	394.28	**35**	89.61	164.8	200.00
16	483.26	394.28	394.28	**36**	90.94	239.58	200.00
17	399.55	489.28	500.00	**37**	155.87	25.00	110.00
18	490.09	489.28	500.00	**38**	89.94	7.05	110.00
19	422.49	511.27	550.00	**39**	89.91	57.05	110.00
20	421.51	421.52	550.00	**40**	421.06	511.28	511.28
Fuel Cost ($/hour)			**124162.64**		**123860.22**		**124577.273**
Total power generation			10,500		10,500		10,500

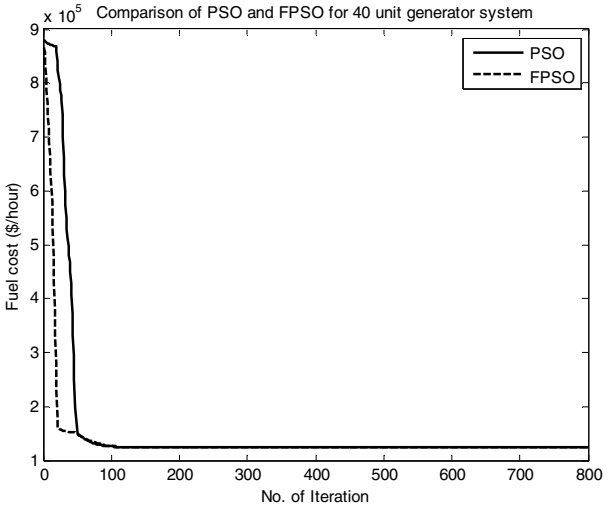


Fig. 5. Convergence plot of forty unit system with valve point loading

6 Conclusions

This paper presents a fuzzy particle swarm optimization approach to non smooth economic load dispatch problem. A new fuzzy technique is incorporated in the PSO to increase the fast convergence to provide the minimum fuel cost while satisfying the equality and inequality constraints. The FPSO algorithm had been successfully applied to ELD problem with and without valve point loading. FPSO is an efficient optimization tool for power system problem.

References

[1] Wood, A.J., Wollenberg, B.F.: Power Generation, Operation, and Control, 2nd edn. Wiley, New York (1996)

[2] Nicholson, H., Lynn, J.W.: The economic loading of transmission systems. Proceedings of the IEE - Part C: Monographs (IET Journal) 105(8), 407–419 (1958)

[3] Hill, E.F., Stevenson, W.D.: An Improved Method of Determining Incremental Loss Factors from Power System Admittances and Voltages. IEEE Transactions on Power Apparatus and Systems 87(6), 1419–1425 (1968)

[4] Akhtar, M.Y.: New approach for economic load scheduling. Proceedings of the Institution of Electrical Engineers (IET Journal) 116(4), 533–538 (1969)

[5] Fink, L.H., Kwatny, H.G., Mcdonald, J.P.: Economic Dispatch of Generation via Valve-Point Loading. IEEE Transactions on Power Apparatus and Systems 88(6), 805–811 (1969)

[6] Rindt, L.J.: Economic Scheduling of Generation for Planning Studies. IEEE Transactions on Power Apparatus and Systems 88(6), 801–804 (1969)

[7] Sasson, A.M.: Nonlinear Programming Solutions for Load-Flow, Minimum-Loss, and Economic Dispatching Problems. IEEE Transactions on Power Apparatus and Systems 88(4), 399–409 (1969)

[8] Dhar, R.N., Mukherjee, P.K.: Reduced-gradient method for economic dispatch. Proceedings of the Institution of Electrical Engineers 120(5), 608–610 (1973)

[9] Wadhwa, C.L., Nanda, J.: New approach to modified co-ordination equations for economic load dispatch. Proceedings of the Institution of Electrical Engineers 123(9), 923–925 (1976)

[10] Megahed, I., Abou-Taleb, N., Iskandrani, E., Moussa, A.: A modified method for solving the economic dispatching problem. IEEE Transactions on Power Apparatus and Systems PAS-96(1), 124–133 (1977)

[11] Malik, J.N., Rao, G.K.: Economic load-dispatch solution using fast-decoupled load flow. Proceedings of the Institution of Electrical Engineers 125(4), 334–335 (1978)

[12] Aoki, K., Satoh, T.: New Algorithms for Classic Economic Load Dispatch. IEEE Transactions on Power Apparatus and Systems 103(6), 1423–1431 (1984)

[13] Mota-Palomina, R., Quintana, V.H.: A penalty Function-Linear programming method for solving Power System Constrained Economic Operation Problems. IEEE Power Engineering Review, 55 (June 1984)

[14] Guoyu, X., Galiana, F.D., Low, S.: Decoupled Economic Dispatch Using the Participation Factors Load Flow. IEEE Transactions on Power Apparatus and Systems PAS-104(6), 1337–1384 (1985)

[15] Sheble, Gerald, B.: Real-Time Economic Dispatch and Reserve Allocation Using Merit Order Loading and Linear Programming Rules. IEEE Power Engineering Review 9(11), 37 (1989)

[16] Warsono, King, D.J., Ozveren, C.S.: Economic Load Dispatch for a power system with renewable energy using Direct Search Method. In: 42nd International Universities Power Engineering Conference, September 4-6, pp. 1228–1233 (2007)

[17] Fukuyama, Y., Ueki, Y.: An application of artificial neural network to dynamic economic load dispatching. In: Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, July 23-26, pp. 261–265 (1991)

[18] Park, J.H., Kim, Y.S., Eom, I.K., Lee, K.Y.: Economic load dispatch for piecewise quadratic cost function using Hopfield neural network. IEEE Transactions on Power Systems 8(3), 1030–1038 (1993)
[19] Su, C.-T., Chiou, G.-J.: A fast-computation Hopfield method to economic dispatch of power systems. IEEE Transactions on Power Systems 12(4), 1759–1764 (1997)
[20] Yoshikawa, M., Toshida, N., Nakajima, H., Harada, Y., Tsurugai, M., Nakata, Y.: On-line economic load dispatch based on fuel cost dynamics. IEEE Transactions on Power Systems 12(1), 315–320 (1997)
[21] Walters, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve point loading. IEEE Transactions on Power Systems 8(3), 1325–1332 (1993)
[22] Bakirtzis, A., Petridis, V., Kazarlis, S.: Genetic algorithm solution to the economic dispatch problem. IEE Proceedings Generation, Transmission and Distribution 141(4), 377–382 (1994)
[23] Wong, K.P., Wong, Y.W.: Genetic and genetic/simulated-annealing approaches to economic dispatch. IEE Proceedings Generation, Transmission and Distribution 141(5), 507–513 (1994)
[24] Park, J.H., Yang, S.-O., Lee, H.-S.: Economic load dispatch using evolutionary algorithms. In: Proceedings International Conference on Intelligent Systems Applications to Power Systems, 28 January-2 February, pp. 441–445 (1996)
[25] Park, J.H., Yang, S.O., Mun, K.J., Lee, H.S., Jung, J.W.: An application of evolutionary computations to economic load dispatch with piecewise quadratic cost functions. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, May 4-9, pp. 289–294 (1998)
[26] Kumaran, G., Mouly, V.S.R.K.: Using evolutionary computation to solve the economic load dispatch problem. In: Proceedings of 2001 Congress on the Evolutionary Computation, vol. 1, pp. 296–301 (2001)
[27] Chen, C.H., Yeh, S.N.: Particle Swarm Optimization for Economic Power Dispatch with Valve-Point Effects. In: IEEE PES Transmission and Distribution Conference and Exposition, pp. 1–5 (2006)
[28] Chakrabarti, R., Chattopadhyay, P.K., Basu, M., Panigrahi, C.K.: Particle Swarm Optimization Technique for Dynamic Economic Dispatch. IE(I) Journal, 48–54 (2006)
[29] Bouktir, T., Labdani, R., Slimani, L.: Economic Power Dispatch Of Power System With Pollution Control Using Multiobjective Particle Swarm Optimization. Journal of Pure & Applied Sciences 4(2), 57–77 (2007)
[30] Selvakumar, A.I., Thanushkodi, K.: A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems 22(1), 42–51 (2007)
[31] Devi, L., Vamsi Krishna, O.: Combined Economic and Emission Dispatch using Evolutionary Algorithms-A Case Study. ARPN Journal of Engineering and Applied Sciences 3(6), 28–35 (2008)
[32] Subramani, S.S., Rajeswari, P.R.: A Modified Particle Swarm Optimization for Economic Dispatch Problems with Non-Smooth Cost Functions. International Journal of Soft Computing 3(4), 326–332 (2008)
[33] Bhattacharya, A., Chattopadhyay, P.K.: A modified Particle Swarm Optimization for solving the non-convex economic dispatch. In: 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), vol. 01, pp. 78–81 (2009)

[34] Chandram, K., Subrahmanyam, N., Sydulu, M.: Secant method with PSO for economic dispatch with valve point loading. In: IEEE Power & Energy Society General Meeting, July 26-30, pp. 1–6 (2009)
[35] Chen, P., Zhao, C., Li, J., Liu, Z.: Solving the Economic Dispatch in Power System via a Modified Genetic Particle Swarm Optimization. In: International Joint Conference on Computational Sciences and Optimization, vol. 1, pp. 201–204 (2009)
[36] Sinha, N., Purkayastha, B., Purkayastha, B.S.: Hybrid PSO/ Self-Adaptive Evolutionary Programs For Economic Dispatch With Nonsmooth Cost Function. International Journal of Recent Trends in Engineering 1(3), 195–200 (2009)
[37] Tao, Z., Cai, J.-D.: A New Chaotic PSO with Dynamic Inertia Weight for Economic Dispatch Problem. In: International Conference on Sustainable Power Generation and Supply (SUPERGEN), pp. 1–6 (2009)
[38] Vlachogiannis, J.G., Lee, K.Y.: Economic Load Dispatch—A Comparative Study on Heuristic Optimization Techniques With an Improved Coordinated Aggregation-Based PSO. IEEE Transactions on Power Systems 24(2), 991–1001 (2009)
[39] Zaraki, A., Bin, O.M.F.: Implementing Particle Swarm Optimization to Solve Economic Load Dispatch Problem. In: International Conference of Soft Computing and Pattern Recognition, pp. 60–65 (2009)
[40] Deep, K., Bansal, J.C.: Particle Swarm Optimization for economic dispatch problems with valve-point effects. In: IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), pp. 1395–1398 (2010)
[41] Meng, K., Wang, H.G., Dong, Z.Y., Wong, K.P.: Quantum-Inspired Particle Swarm Optimization for Valve-Point Economic Load Dispatch. IEEE Transactions on Power Systems 25(1), 215–222 (2010)
[42] Khamsawang, S., Wannakarn, P., Jiriwibhakorn, S.: Hybrid PSO-DE for solving the economic dispatch problem with generator constraints. In: The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 5, pp. 135–139 (2010)
[43] Saber, A.Y., Senjyu, T., Tsukasa, M., Naomitsu, U., Toshihisa, F.: Unit commitment by heuristics and absolutely stochastic simulated annealing. IET Generation Transmission Distribution 1(2), 234–243 (2007)
[44] Sinha, N., Chakrabarti, R., Chattopadhyay, P.K.: Evolutionary programming techniques for economic load dispatch. IEEE Transactions on Evolutionary Computation 7(1), 83–94 (2003)
[45] Thanushkodi, K., Muthu Vijaya Pandian, S., Dhivyapragash, R.S., Jothikumar, M., Sriramnivas, S., Vinodh, K.: An Efficient Particle Swarm Optimization for Economic Dispatch Problems With Non-smooth cost functions. WSEAS Transactions on Power Systems 3(4), 257–266 (2008)
[46] Arijit, B., Sambarta, D., Panigrahi, B.K., Pandi, V.R., Das, S., Abraham, A., Badr, Y.: Economic Load Dispatch Using a Chemotactic. Differential Evolution Algorithm, 252–260 (2009)
[47] Yadmellat, P., Salehizadeh, S.M.A., Menhaj, M.B.: Fuzzy Parameter Particle Swarm Optimization. In: First International Conference on Intelligent Networks and Intelligent Systems, pp. 93–98 (2008)
[48] Chaturvedi, D.K.: Soft Computing: Applications to electrical engineering problem. Springer, Berlin (2008)
[49] Chaturvedi, D.K.: Modeling and Simulation of Systems Using Matlab@ /Simulink@. CRC Press, New York (2009)

Author Biographic

First Author Sanjeev Kumar, Research Scholar, is with the Dayalbagh Educational Institute, (Deemed University) Dayalbagh, Agra, India. He did B.Tech. in Electrical and Electronics Engineering and M. Tech. in Engineering System in 2007 and 2009 respectively. At present he is pursuing his Ph.D. in Electrical Power System. His areas of interest are soft computing and electrical power system.

Second Author D. K. Chaturvedi is currently working as professor in Dayalbagh Educational Institute, (Deemed University) Dayalbagh, Agra, India. He did B.Tech in Electrical Engineering and M.Tech. in Engineering System and Management in 1988 and 1993 respectively. He did his Ph.D. in power system and Post Doctoral research in 1998 and 2002 respectively. His areas of interest are electrical power system, soft computing and electrical machine. **http://works.bepress.com/dk_chaturvedi**

System Identification of Single Machine Infinite Bus Using GA-Fuzzy Technique

D.K. Chaturvedi, Himanshu Vijay, and Sanjeev Kumar

Department of Electrical Engineering, Faculty of Engineering, D.E.I. (Deemed Univ.), Dayalbagh, Agra, India
dkc.foe@gmail.com

Abstract. Soft Computing Technique mainly consisting of ANN, Fuzzy system and GA. GA optimization is slow and depending on the number of variables. To improve the convergence of GA, a modification in normal GA is proposed in which the GA parameters like cross over probability(Pc), mutation probability (Pm) and population size (POPSIZE) are modified using fuzzy system dynamically during execution. The proposed integrated approach of GA-Fuzzy is used for system identification of single machine infinite bus system and the results are compared with conventional ARX and ARMAX methods.

Keywords: System identification, Genetic Algorithms, Fuzzy System, Synchronous machine.

1 Introduction

The synchronous machine parameter identification problem is very important to know the performance condition of single machine infinite bus system. The accurate identification of system parameters will help in accurate and better control, protection and planning. In this paper, soft computing methodology is used for system parameter identification. Soft Computing (SC) is an emerging collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty, and partial truth to achieve robustness, tractability and total low cost. *Soft-Computing* is defined as a collection of techniques spanning many fields that fall under various categories in Computational Intelligence [3]. Soft-Computing has three main branches: Fuzzy Systems, Evolutionary Computation [6], Artificial Neural Computing, with the latter subsuming Machine Learning (ML) and Probabilistic Reasoning (PR), belief networks, chaos theory, parts of learning theory and Wisdom based Expert system (WES), etc. It is very good tool for system parameter identification because it can handle sensor fuzziness and also learn from the past experimentation (parameter calculations).

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 111–119.
springerlink.com

2 Fuzzy Logic

The logic of fuzzy sets was proposed by Prof. Lofti A. Zadeh, who introduced the concept in systems theory, and later extended it for approximate reasoning in expert systems. Fuzzy logic is concerned with fuzzy sets and logical connectives for modeling the human-like reasoning problems of the real world [4].

A fuzzy expert system consists of a *fuzzy rule base*, a *fuzzification module*, an *inference engine*, and a *defuzzification module* as shown in Fig. 1.

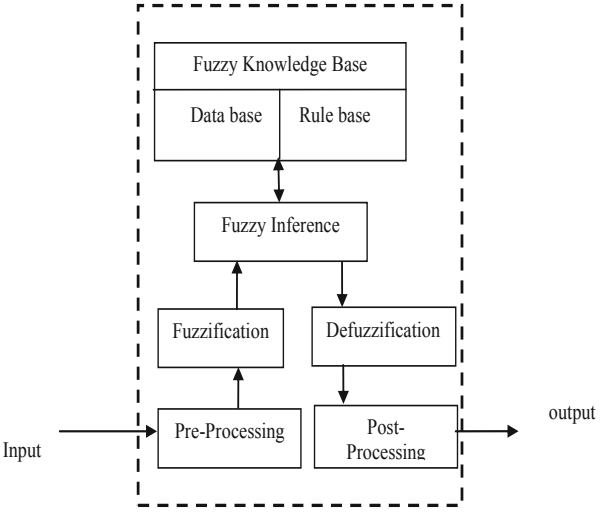


Fig. 1. Functional Module of a Fuzzy System

Following are the advantages of the fuzzy systems:

- Capacity to represent inherent uncertainties of the human knowledge with linguistic variables.
- Simple interaction of the expert of the domain with the engineer designer of the system;
- Easy interpretation of the results, because of the natural rules representation.
- Easy extension of the base of knowledge through the addition of new rules.
- Robustness in relation of the possible disturbances in the system.

and its disadvantages are:

- Incapable to generalize, or either, it only answers to what is written in its rule base.
- Not robust in relation the topological changes of the system, such changes would demand alterations in the rule base.
- Depends on the existence of a expert to determine the inference logical rules.

3 Genetic Algorithms

Prof. John Holland in 1975 proposed Genetic Algorithms (GA) that mimics the biological evolution process for solving problems in a wide domain. These

Genetic Algorithms (GA) work with one large population, i.e. in the re-combination step, each individual may potentially choose any other individual from the population as a mate. Then **GA** operators are performed to obtain the new child offspring. The performance of the strings, often called fitness, is then evaluated with the help of some functions, representing the constraints of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. The second phase in the genetic manipulation process is termed mutation, where the bits at one or more randomly selected positions of the chromosomes are altered. The mutation process helps to overcome trapping at local maxima. The offsprings produced by the genetic manipulation process are the next population to be evaluated as shown in Fig.2.

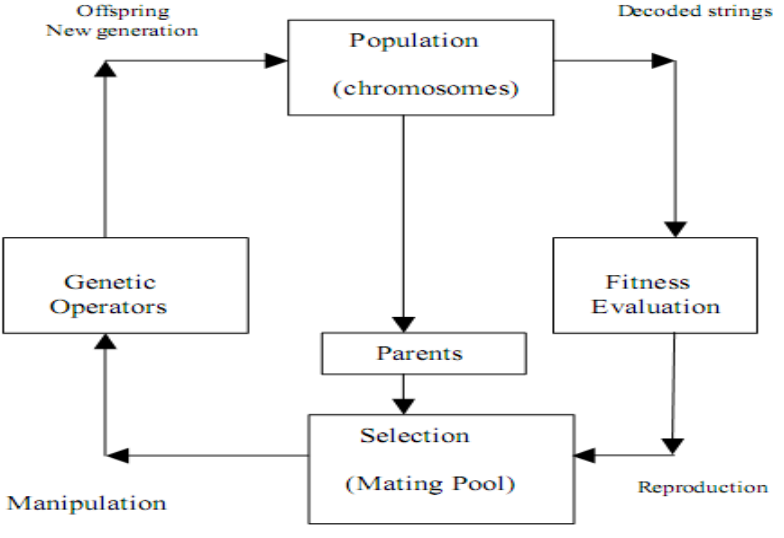


Fig. 2. Cycle of Genetic Algorithms

The GA cycle consists of the following three sequential steps:

a) Generation of population.
b) Genetic evolution through crossover (Ref. Fig. 3) followed by mutation (ref. Fig. 4).
c) Selection of better candidate states from the generated population

In step (a) of the above cycle, a few initial problem states are first identified. The step (b) evolves new chromosomes through the process of crossover and mutation. In step (c) a fixed number of better candidate states are selected from the generated population. The above steps are repeated a finite number of times for obtaining a solution for the given problem.

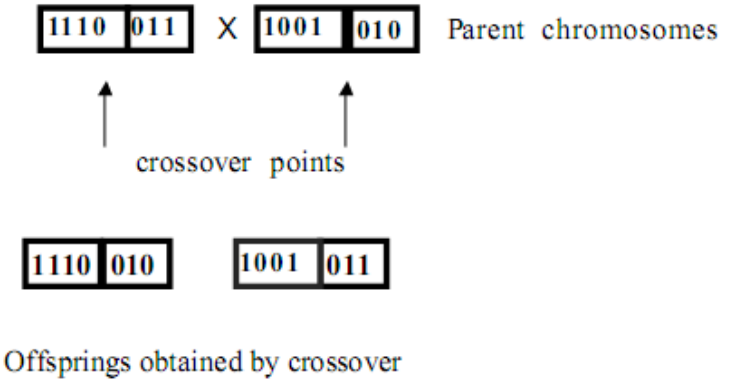


Fig. 3. The Crossover operation

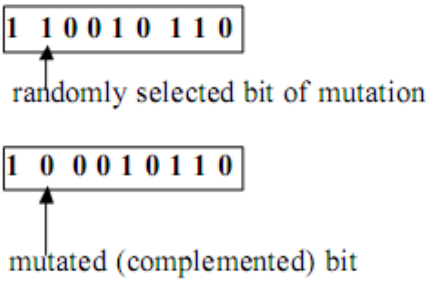


Fig. 4. The Mutation Operation

4 GA- Fuzzy Approach for System Identification

It is well known that the GA optimization is slow and depending on the number of variables [2, 5].

To improve the convergence of GA, a modification in normal GA is required in which the GA parameters like cross over probability (P_c), mutation probability (P_m) and population size (POPSIZE) are modified using some fuzzy rules dynamically during execution.

GA-fuzzy approach of system identification is shown in Fig. 5, in which GA-Fuzzy approach is used to identify the suitable model parameters to reduce the error.

At the starting stage, high crossover probability and low mutation probability yield good results, because a large number of crossover operations produce better chromosomes for a finite number of generations, after that the fitness value of each chromosome vector becomes almost equal. Beyond this the effect of crossover is insignificant due to little variation in the chromosome vectors in

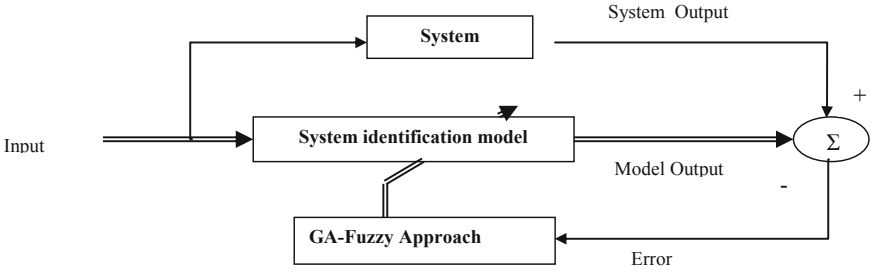
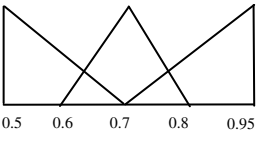
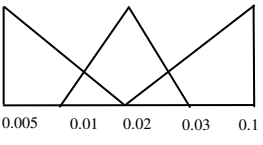
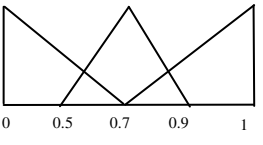
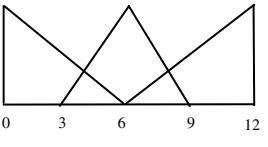
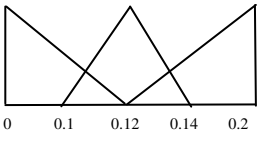


Fig. 5. GA-Fuzzy Approach for system identification

Table 1. Membership functions and range of variables

Variable	Linguistic Terms	Membership Functions
Crossover Probability (P_c)	Low Medium High	
Mutation Probability (P_m)	Low Medium High	
Best Fitness (*BF*)	Low Medium High	
Number of generations for unchanged BF (*UN*)	Low Medium High	
Variance of Fitness (*VF*)	Low Medium High	

that particular population. At later stages, increasing the mutation rate of the chromosomes inculcates new characteristics in the existing population and therefore diversifies the population.

Therefore, philosophy behind varying Pc and Pm is that the response of the optimization procedure depends largely on the stage of optimization, i.e. a high fitness value may require relatively low crossover and high mutation probabilities for further improvement, alternatively, at low fitness values the response would be better with relatively high crossover and low mutation probabilities.

Schuster [13] proposed heuristics for optimal setting of the mutation probability (Pm). Fogarty [7] and Booker [1] investigated time dependencies on the mutation and crossover probabilities respectively. Grefenstette [10] and Schaffer [12] found optimal settings for all these parameters of the GA by experiment.

In this research work, a GA-Fuzzy approach is used in which ranges of parameters- crossover probability (Pc) and mutation probability (Pm) have been divided into LOW, MEDIUM and HIGH membership functions and each is given some membership values as shown in Table 1.

The GA parameters (Pc and Pm) are varied based on the fitness function values as per the following logic:

(1) The value of the best fitness for each generation *(BF)* is expected to change over a number of generations, but if it does not change significantly over a number of generations *(UN)* then this information is considered to cause changes in both P_c and P_m
(2) The diversity of a population is one of the factors, which influences the search for a true optimum. The variance of the fitness values of objective function *(VF)* of a population is a measure of its diversity. Hence, it is also considered as another factor on which both P_c and P_m may be changed.

The membership functions and membership values for these three variables *(BF, UN* and *VF)* are selected after several trials to get optimum results.

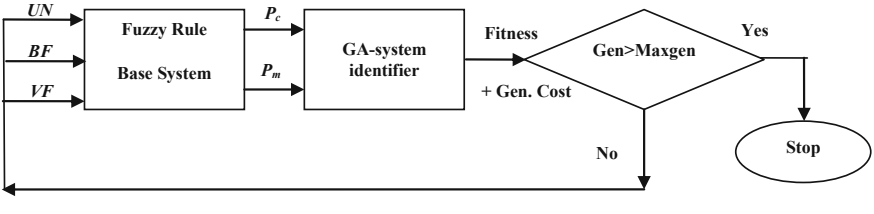


Fig. 6. Block diagram of GA-Fuzzy Approach for system identification

5 Results and Discussions

The proposed GA-Fuzzy system identifier is used for identifying the single machine infinite bus systems with the following GA parameters:

Population size (POPSIZE) : 50, Maximum number of Generations: 200, Selection operator: Stochastic Remainder, Initial crossover probability P_c: 0.9, Initial Mutation probability P_m: 0.01.

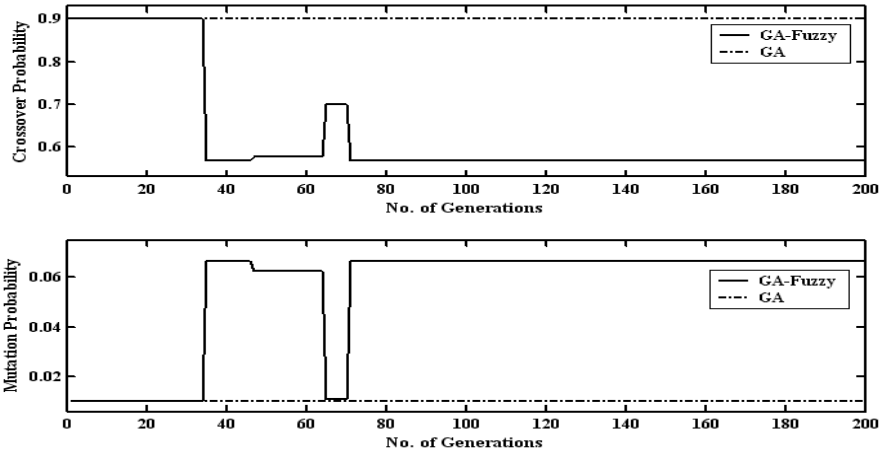


Fig. 7. Crossover and Mutation Probabilities variations for GA-Fuzzy system identifier

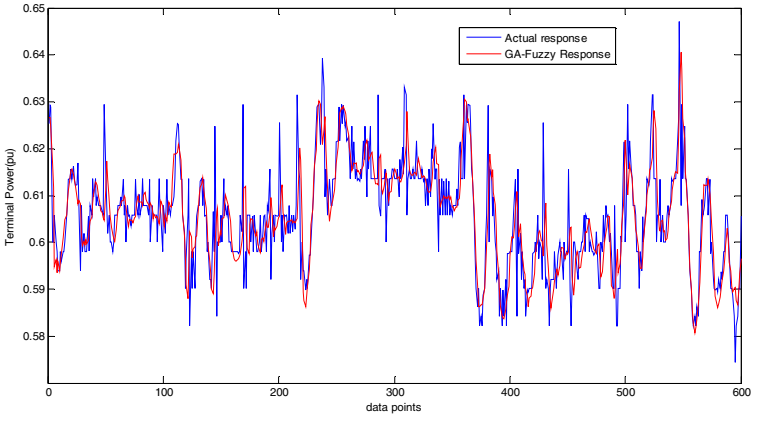


Fig. 8. Terminal power (Pt) response curves using GA-Fuzzy Approach

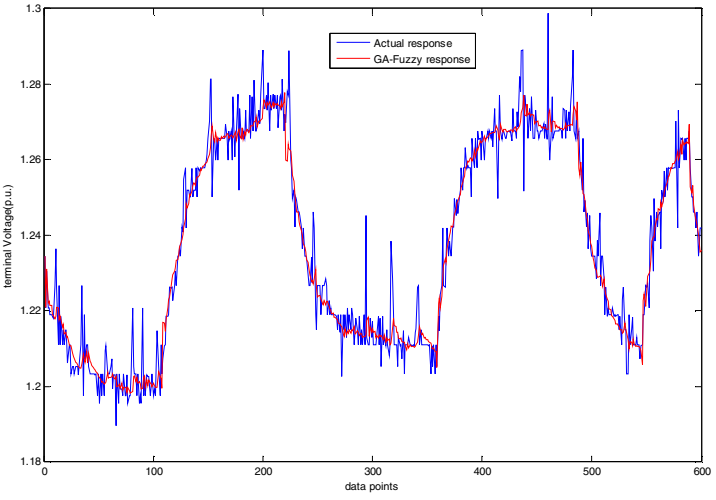


Fig. 9. Terminal Voltage (Vt) Response curves using GA-Fuzzy Approach

The value of P_c and P_m varies during execution as shown in Fig. 7. The system model parameters are identified using GA-Fuzzy approach and the results are compared with conventional ARX and ARMAX model as shown in Table – 2 and Fig. 8-9.

Table 2. Comparison of RMS, MAX and MIN Errors for different model outputs

Model Name	**ARX**	**ARMAX**	**GA-Fuzzy (Power)**	**GA-Fuzzy (Voltage)**
RMS Error	0.0076	0.0076	0.0068	0.0060
Max. Error	0.0307	0.0310	0.0316	0.0274
Min. Error	-0.0305	-0.0354	-0.0309	-0.0296

6 Conclusions

In this paper, the problems of simple Genetic Algorithm (GA) like slow convergence rate, Random roulette wheel selection etc. are overcome in GA-Fuzzy approach. In proposed GA-Fuzzy approach the GA - parameters like cross over probability (P_c), mutation probability (P_m) are changed dynamically during execution through fuzzy rules to get better convergence rate. The proposed GA-Fuzzy approach is used for system identification of single machine infinite bus

system. The results are compared with ARX and ARMAX models and it is found that this approach is better than conventional methods and rms error is consistently less as compared to other models.

References

1. Booker, L.: Improving Search in Genetic Algorithms. In: Genetic Algorithms and Simulated Annealing, Pitman, Landon (1987)
2. Chaturvedi, D.K., Das, V.S.: Optimization of Genetic Algorithm Parameters. In: National Conference on Applied Systems Engineering and Soft Computing (SASESC 2000), pp. 194–198. Organized by Dayalbagh Educational Institute, Dayalbagh (2000)
3. Chaturvedi, D.K.: Soft Computing and Its Application in electrical Engineering. Springer, Heidelberg (2008)
4. Chaturvedi, D.K.: Modeling and Simulation of Systems Using MATLAB® / Simulink®. CRC Press (2009)
5. Deb, K.: Optimization for Engineering Design. Prentice Hall of India, New Delhi (1995)
6. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
7. Fogarty, T.C.: Varying the Probability of Mutation in the Genetic Algorithm. In: Proc. 3rd Int. Conf. on Genetic Algorithms & Applications, Arlington, VA, pp. 104–109 (1981)
8. Goldberg, D.E., Holland, J.H.: Genetic Algorithms and Machine Learning, vol. 3. Kluwer Acedamic Publisher (1988)
9. Goldberg, D.E., Holland, J.H.: Genetic Algorithms, in Search, Optimization and Machine Learning. Addison Wesley (1989)
10. Grefenstette, J.J.: Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics SMC-16(1), 122–128 (1981)
11. Kazarlis, S.A., Bakirtzis, A.G., Petridis, V.: A Genetic Algorithm Solution to the Unit Commitment Problem. IEEE Trans. on Power Systems 11(1), 83–92 (1996)
12. Schaffer, J.D., Caruna, R.A., Eshelman, I.J., Das, R.: A Study of Control Parameters affecting Online Performance of Genetic Algorithms for Function Optimization. In: Proceedings of 3rd International Conference on Genetic Algorithms and Applications, Arlington, VA, pp. 51–60 (1981)
13. Schuster, P.: Effects of Finite Population size and Other Stochastic Phenomena in Molecular Evolution. In: Complex System Operational Approaches Neurobiology, Physics and Computers. Springer, Heidelberg (1985)
14. Suh, Y.H., Van Gucht, D.: Incorporating Heuristic Information into Genetic Search. In: Proc. of 2nd Int. Conf. on Genetic Algorithms, pp. 100–107. Lawrence Emlbaum Associates (1987)

EAVD: An Evolutionary Approach Based on Voronoi Diagram for Node Deployment in Wireless Sensor Networks

Naeim Rahmani[1] and Farhad Nematy[2]

[1] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz,Iran
naeim.rahmani@yahoo.com
[2] Young Researchers Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran
farhad_nematy@yahoo.com

Abstract. Wireless sensor networks composed of a large number of micro sensors can be an effective tool for gathering data in a variety of environments. Wireless sensor network consist of hundreds or thousands of these sensor nodes that are capable of sensing the environment and sending them to other sensor nodes or to base station through wireless links. The area that can be covered by sensor network is called coverage of the sensor network and is dependent to number of nodes and their locations. In this paper an evolutionary approach based on voronoi diagram is presented. Proposed algorithm divides the field into voronoi cells then tries to deploy additional mobile nodes in each cell to heal coverage holes. In this work the overlap of nodes in the same voronoi cell and neighbor cells is determined. Simulations result show that proposed algorithm has better coverage than previous works in literature.

Keywords: coverage, wireless sensor networks, genetic algorithm, node placement, voronoi diagram.

1 Introduction

The advancements in technological improvements has lead to the use of small, inexpensive, low-power, distributed The category of routing algorithms that facilitate the devices for local processing and wireless communication establishment of more than one path between the source and [1]. These devices are called sensor nodes. Wireless sensor network consist of hundreds or thousands of these sensor nodes that are capable of sensing the environment and sending them to other sensor nodes or to base station through wireless links [2]. The WSN has many advantages, including the miniaturization of sensor nodes, easy deployment, low cost, and tolerance to fault conditions. Sensor nodes can be deployed randomly or

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 121–129.
springerlink.com

in predefined locations. Robot assisted deployment can be used in wireless sensor networks [3-5]. Wireless sensor networks usually were deployed in remote and hostile surroundings, and people cannot attend the sensor nodes and usually a random deployment is used [6]. Because of random deployment nodes scatter through the field with different densities which means in some regions number of deployed nodes are high and in other regions number of deployed nodes are low. Different density will lead to coverage holes, uncovered regions inside sensor filed, which is not good in wireless sensor networks. Existence of coverage holes means that if an event occurs inside the hole there is no sensor node to report it. Many researchers studied the coverage problem and many algorithms are presented in literature [7-11]. In most of these researches they assume the network to be mobile or hybrid. In a hybrid network most of nodes are stationary and there are few mobile nodes. The main objective for using mobile sensor nodes is to heal coverage holes after the initial network deployment, such that the area coverage can be maximized while incurring the least moving cost. In most researches voronoi diagram is used to specify coverage holes. Voronoi diagram is partition of sites in such a way that points inside a polygon are closer to the site inside the polygon than any other sites, thus one of the vertices of the polygon is the farthest point of the polygon to the site inside it [12]. Therefore Voronoi diagram can be used as one of the sampling method in determining WSN coverage; with the sensors act as the sites, if all Voronoi polygons vertices are covered then the field is fully covered otherwise coverage holes exist [7]. Figure 1 presents a voronoi diagram for 24 random points.

The main idea in our research is to deploy some stationary nodes inside the field then the voronoi diagram can be calculated for these stationary nodes. After that a genetic algorithm tries to deploy mobile nodes inside each voronoi cell in

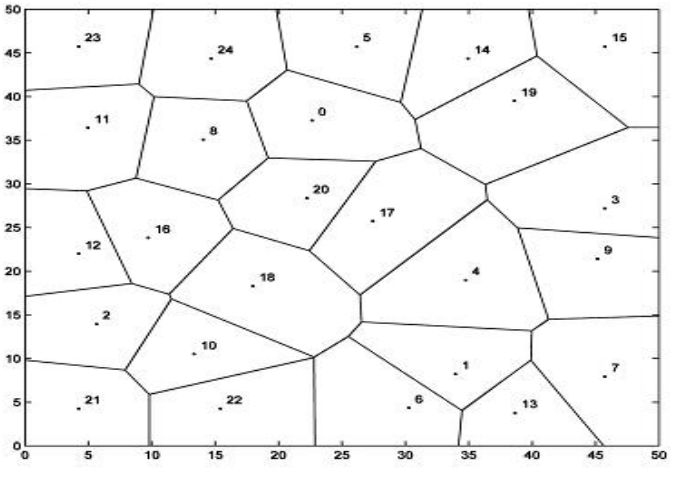


Fig. 1. Voronoi diagram with 24 random points

such a way that deployed mobile nodes have minimum overlap with other nodes inside the same voronoi cell and neighbor voronoi cells. In this way coverage can be maximized and coverage holes will be healed. The rest of this paper is organized as follows: section 2 discusses the related work and section 3 presents the proposed algorithm. Simulations and experimental results discussed in section 4. Section 5 concludes the paper.

2 Related Works

In most of coverage researches it is supposed that some stationary nodes scattered in a field, and then some mobile nodes try to cover existing coverage holes. In [8], an algorithm proposed using a combination of mobile and static sensors to construct sensor networks to balance cost and sensor coverage. They identified the problem of deploying mobile sensors in a mixed sensor network as an NP-complete problem and designed bidding protocols to tackle this problem in a distributed fashion, in which static sensors act as bidders and mobile sensors act as hole-healing servers. In [10] a genetic algorithm for layout optimization of sensor nodes is presented that uses a clustering architecture and tries allocating different nodes to different cluster heads to achieve the best coverage. Also in [11] cluster based architecture is used to determine best places for cluster heads to maximize the coverage. In this research a genetic algorithm is used to relocate cluster heads to best possible locations. Also in [8] a used of voronoi diagram for finding best places for hole coverage is discussed. They assume that the field is divided into voronoi cells and because each voronoi vertex is farthest point form center of it, the vertex is best location to deploy mobile node. Figure 2 illustrates this idea.

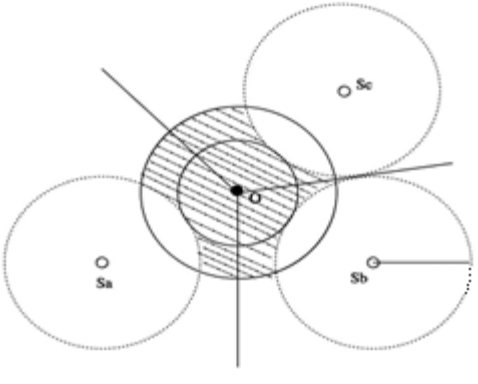


Fig. 2. Deployment of mobile node in voronoi vertex

The main disadvantage of this approach is that voronoi vertices are very close to each other in some places and deployment of nodes in these places will led to overlap. Figure 3 shows this problem.

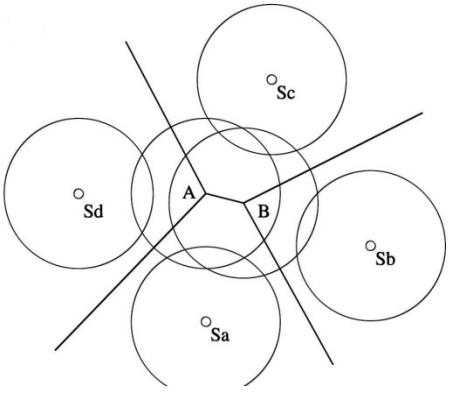


Fig. 3. Overlap problem for deployment on voronoi vertices

3 Proposed Genetic Algorithm

In our network model we assume that all of sensors are of the same type and are deployed randomly in the sensing field. Our proposed algorithm at first stage uses voronoi diagram to divide the field in to cells then for each cell it uses genetic algorithm to deploy additional mobile nodes to the holes with regard to other nodes deployed in neighbor cell to avoid overlap in boundary of cells. Figure 4 illustrates the overlaps inside the voronoi cell and overlaps of nodes of neighbor voronoi cells.

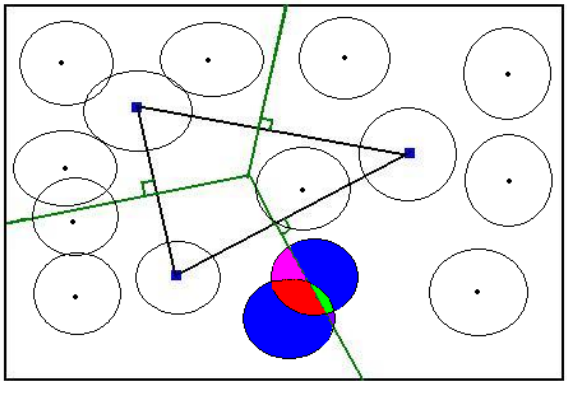


Fig. 4. Overlap inside and outside of cell

Initial population

A 2×n array is used to represent the solutions in which i_{th} cell is coordination of a node inside a cell. Each coordination in each chromosome is random but it must be inside the corresponding voronoi cell.

x	34	14	75	22	18	3
y	69	93	6	51	50	24

Fig. 5. Chromosome representation

Fitness function

To calculate the fitness of each cell the Euclidean distance between each two node (p1, p2) is calculated then the overlap distance can be calculated via equation 1. Figure 6 depicts overlap distance.

$$X = 2r - \text{distance}(p1,p2) \quad (1)$$

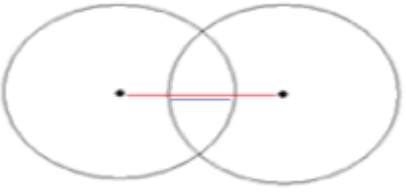


Fig. 6. Overlap distance

Total ideal coverage is equal to area of corresponding voronoi cell which is presented in equation 2.

$$\text{Total\_ideal\_coverage} = \text{polygonarea}(v(i)) \quad (2)$$

Amount of penalty is equal to half of a circle with radius equal to overlap distance that can be calculated via equation 3. Figure 7 demonstrates the amount of penalty for two points.

$$\text{Penalty}(i) = ((x^{\wedge} 2)*3.14)/2 \quad (3)$$

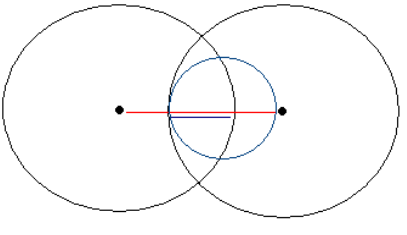


Fig. 7. Penalty area as a circle with radius equal to penalty distance

The overall penalty is calculated through equation 4 as sum of penalty of nodes inside the voronoi cell and penalty of nodes in neighbor cells.

$$\text{Total\_penalty}=\Sigma\ \text{Penalty}(i) + \text{penalty\_neighbors}(i) \quad (4)$$

The fitness value is difference of total penalty and total ideal coverage and is calculated through equation 5.

$$\text{Fitness}=\text{Total\_ideal\_coverage}-\text{Total\_penalty} \quad (5)$$

Selection mechanism

Here a tournament selection mechanism is used to select chromosomes to produce next generation with k=5. In this selection mechanism, k chromosomes is selected randomly and one of them which has maximum fitness value moves to mating pool.

Crossover

Two point crossover is used to produce new offsprings from selected parents. To crossover selected parents two random points is determined and the gens between two points in chromosomes exchanged. The crossover probability determined 0.7.

Mutation

Mutation is done through determining a new random point inside chromosome and generating new random coordination inside voronoi cell with mutation probability equal to 0.4.

Termination criteria

There are different termination criteria such as constant number of iterations or reaching predetermined fitness value or until in few recent iterations no improvement could be achieved. Here the third method is used and iterations continued until in 20 recent iterations no improvement achieved.

4 Simulations and Experimental Results

A 10 × 10 field is used to evaluate proposed algorithm and 10 stationary nodes scattered in the field. Voronoi diagram calculated for these 10 stationary nodes and 4 mobile nodes for each voronoi cell is allocated. After that for each voronoi cell genetic algorithm used to optimize their coordination. The resulted voronoi diagram is shown in figure 8. Simulation parameters are presented in table 1.

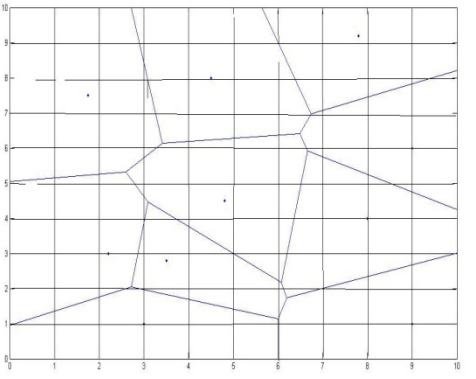


Fig. 8. Voronoi diagram for 10 stationary nodes

Table 1. Simulation Parameters

parameter	value	description
n	10	Number of nodes in field
radius	1	Sensing radius of each node
Number of iterations	Until no improvement achived	Number of iterations
Population size	100	Number of individuals in population
Number of nodes	4	Number of additional nodes to heal holes
P_C	0.7	Percentage of crossover
P_M	0.4	Percentage of mutation

Figure 9 depicts performance of EAVD and as can be seen the coverage percentage was almost 40 percent at initial deployment and improved through 500 iterations to 97 percent. We compared EAVD with [13] and simulation result show that EAVD has better coverage. Figure 10 shows the deployment of mobile nodes inside each voronoi cell and coverage improvement.

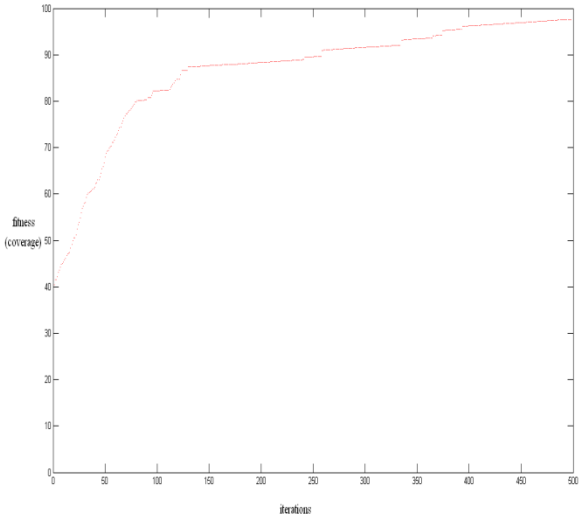


Fig. 9. Coverage percentage of EAVD

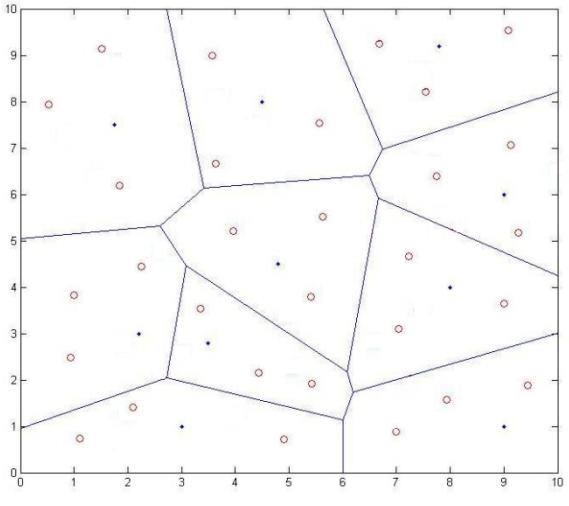


Fig. 10. Deployed mobile nodes, 3 mobile node per cell

5 Conclusion

In this paper an evolutionary approach based on voronoi diagram (EAVD) proposed which aims to divide sensor field into voronoi cells and optimize the location of mobile nodes in a way that coverage be maximized. Simulations done to evaluate proposed algorithm and simulation results show that EAVD has better

coverage than recent proposed algorithms in the literature. As future work, proportional number of mobile nodes will be used to have more mobile nodes in bigger voronoi cells and fewer nodes in smaller voronoi cells.

References

[1] Bharathidasan, A., Ponduru, V.A.S.: Sensor Networks: An Overview. IEEE Potentials 22(2), 20–23 (2003)

[2] Akkaya, K., Younis, M.: A Survey on Routing Protocols for Wireless Sensor Networks. Elsevier Ad Hoc Network Journal (2005)

[3] Howard, A., Matarić, M.J., Sukhatme, G.S.: An Incremental Self-Deployment Algorithm for Mobile Sensor Networks. Autonomous Robots 13(2)

[4] Suzuki, T., Kawabata, K., Hada, Y., Tobe, Y.: Deployment of Wireless Sensor Network using Mobile Robots to Construct an Intelligent Environment in a Multi-Robot Sensor Network. In: Advances in Service Robotics, Austria (July 2008); ISBN: 978-953-7619-02-2

[5] Nematy, F., Rahmani, A.M., Teshnelab, M., Rahmani, N.: Ant Colony Based Node Deployment and Search in Wireless Sensor Networks. In: International Conference on Computational Intelligence and Communication Networks (CICN), pp. 363–366 (2010)

[6] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38, 393–422 (2002)

[7] Ghosh, A.: Estimating coverage holes and enhancing coverage in mixed sensor networks. In: IEEE International Conference on Local Computer Networks, pp. 68–76 (2004)

[8] Wang, G., Cao, G., Berman, P., Porta, T.F.L.: Bidding protocols for deploying mobile sensors. IEEE Transactions on Mobile Computing 6(5), 515–528 (2007)

[9] Wang, P.-C., Hou, T.-W., Yan, R.-H.: Maintaining coverage by progressive crystallattice permutation in mobile wireless sensor networks. In: IEEE International Conference on Systems and Networks Communication (ICSNC), pp. 1–6 (2006)

[10] Bhondekar, A.P., Vig, R., Singla, M.L., Ghanshyam, C., Kapur, P.: Genetic Algorithm Based Node Placement Methodology for Wireless Sensor Networks. In: Proceedings of International Multi Conference of Engineers and Computer Scientists (IMECS 2009), Hong Kong, March 18-20, vol. I (2009)

[11] Nematy, F., Rahmani, N., Yagouti, R.: An Evolutionary Approach for Relocating Cluster Heads in Wireless Sensor Networks. In: IEEE International Conference on Computattonal Intelligence and Communicafron Networks, Bhopal, India, November 26-28 (2010)

[12] Aurenhammer, F., Klein, R.: Voronoi diagrams. In: Sack, J., Urrutia, G. (eds.) Handbook of Computational Geometry, ch. V, pp. 201–290. Elsevier Science Publishing (2000)

[13] Rahmani, N., Nematy, F., Rahmani, A.M., Hosseinzadeh, M.: Node Placement for Maximum Coverage Based on Voronoi Diagram Using Genetic Algorithm in Wireless Sensor Networks, Scientific and Research Essay, pp. 456–467 (2011)

Adiabatic Amplifier and Power Analysis of Different Adiabatic Inverters

Shilpa Katre[1], Prachi Palsodkar[2], and Minal Ghute[1]

[1] Department of Electronics & Telecommunication
[2] Department of Electronics Engineering
Yeshwantrao Chavan College of Engineering, Nagpur, India
shilpa.katre@gmail.com, bisen_tejeshwari@yahoo.co.in,
minal_ghute@rediffmail.com

Abstract. This paper proposes a new method of reducing the energy dissipation. Adiabatic logic style is proving to be an attractive solution for low power digital design. Many researchers have introduced different adiabatic logic styles in last few years and proved that these are better than CMOS. Adiabatic switching technique based on energy recovery principle is one of the innovative solutions at circuit and logic level to achieve reduction in power dissipation This paper mainly consist of implementation of Adiabatic amplifier and Basic Adiabatic inverters (ECRL,CAL,CPAL).Its comparative power analysis with conventional CMOS inverter is carried out. In this paper, we show that the power dissipation in adiabatic inverters is less as compared to CMOS inverter. All circuits are implemented using Chartered 0.35µm CMOS technology Tanner EDA 13.0 tool.

Keywords: CPAL, adiabatic, power dissipation.

1 Introduction

Power consumption is an increasing concern in VLSI circuits. Sharply increasing need for portable electronic devices has reinforced the need of low power design methodologies in the recent years .Adiabatic logic style is proving to be an attractive solution for low power digital design. The term Adiabatic comes from the fact that an adiabatic process is one in which the total heat or energy in the system remains constant. Research in this area has mainly been fueled by the fact that as circuits gets smaller and faster, their energy dissipation greatly increases, a problem that an adiabatic circuit promises to solve. In this logic energy is recycled back instead of being wasted. Adiabatic circuitry uses a special type of power supply instead of constant DC supply as in Conventional CMOS logic. These circuits use power-clock voltage in the form of a ramp or sinusoidal signal. This paper mainly deals with the implementation of Adiabatic Amplifier and

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 131–139.
springerlink.com

different adiabatic inverters using 0.35 micron technology and Tanner EDA 13.0 tool.

1.1 CMOS Logic

The three contributory factors to the total power dissipation in CMOS are

a) Static power dissipation due to leakage current flowing through reverse biased p-n junctions and sub threshold current.
b) Dynamic power dissipation due to charging and discharging of load capacitor during the time the output is switching and
c) The short circuit current power dissipation during switching due to n-channel and p-channel transistors of the CMOS structure conducting in saturation for a short time during switching.

The contribution due to dynamic power dissipation is the highest and is about 70% while that due to static power dissipation is the lowest and is about 10%. The remaining contribution to the total power dissipation is due to short circuit current dissipation. In Fig.1 it is seen that C_L charges to V_{DD} through F while discharges to ground through F'. During charging an energy = (1/2) $C_L V_{DD}^2$ is lost in the pull up circuit while during discharging energy = (1/2) $C_L V_{DD}^2$ (which was stored in the capacitor) is lost to the ground. Thus in one cycle of charge and discharge, energy $C_L V_{DD}^2$ is dissipated. If the output is switching at frequency f and the switching activity is α, then the dynamic power dissipation is given by,

$$\mathbf{P}_{\mathbf{dynamic}} = \alpha\, \mathbf{C_L V_{DD}^2 f}$$

A considerable amount of energy saving can be obtained if the energy which is generally lost to the ground during discharging period in a conventional CMOS logic is returned back to the supply itself. If recycling of the energy drawn from the supply is done then the energy efficiency of the logic circuits can be increased. Adiabatic logic design offers this possibility.

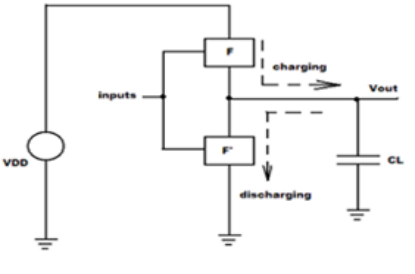


Fig. 1. Conventional CMOS Logic Gate

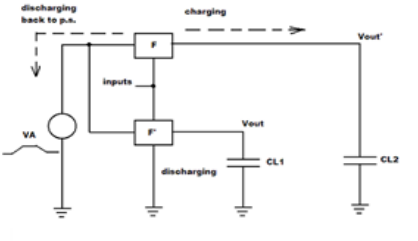


Fig. 2. An Adiabatic Logic Gate

1.2 Adiabatic Logic

The Fig.2 shows the general circuit topology of an adiabatic counterpart. To convert a conventional CMOS logic gate into an adiabatic gate, the pull-up and pull-down networks must be replaced with complementary transmission-gate networks. The T-gate network implementing the pull-up function is used to drive the true output of the adiabatic gate, while the T-gate network implementing the pull-down function drives the complementary output node. Note the all inputs should also be available in complementary form. Both networks in the adiabatic logic circuit are used to charge-up as well as charge-down the output capacitances, which ensure that the energy stored at the output node can be retrieved by the power supply, at the end of each cycle. To allow adiabatic operation, the DC voltage source of the original circuit must be replaced by a pulsed-power supply with ramped voltage output. The reduction of energy dissipation comes at the cost of slower switching speed, which is the ultimate trade-off in all adiabatic methods.

Typical adiabatic switching circuit is as shown in Fig.3 Here, the load capacitance is charged by a constant current source, which corresponds to a linear voltage ramp. The main difference between the conventional CMOS circuits and the adiabatic circuits is that in adiabatic circuit the load capacitor is charged by a constant current source while in Conventional CMOS circuit, it is charged by a constant voltage source.

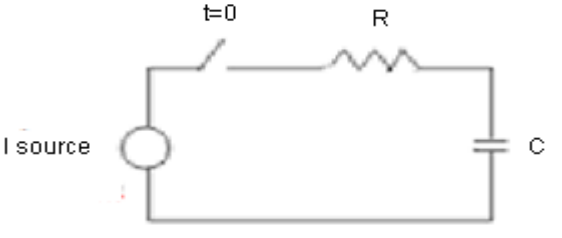


Fig. 3. Adiabatic Switching

In the Fig. 3, let R be the on-state resistance of pull-up network of the circuit. Assuming $V_c(t)=0$ at t=0

$$Vc(t) = (1/C)\, I_{source}\, t$$

Where,

$$I_{source} = C\,(V_c(t)/t)$$

Energy dissipated in the resistor R from t=0 to t=T can be found as:

$$E_{diss} = R\int I_{source}^2 dt$$
$$= I_{source}^2 RT$$
$$= (RC/T)C\, V_c^2(t)$$

From this equation it is observed that;

(1) If charging time is greater than 2RC then the dissipated energy is smaller than that for conventional CMOS circuit.
(2) Dissipated energy is inversely proportional to T, which means that dissipated energy can be made arbitrarily smaller by increasing the charging time.
(3) Dissipated energy is proportional to R in contrast to conventional CMOS case wherein dissipated energy depends on load capacitor and voltage swing.
(4) As charging resistance decreases, the energy dissipated decreases.

Fig.2 depicts the charge flow in adiabatic circuit. Pull-up circuit drives the true output of the adiabatic gate while pull-down circuit drives the complementary output node. Both the networks in adiabatic charge up as well as charge down the output capacitor. At the end of the cycle, the energy flows back into the power supply.

2 Adiabatic Amplifier

A typical adiabatic switching circuit is as shown in Fig. 4. It has two transmission gates, two NMOS clamps and power-clock supply VA. The operation of the adiabatic circuit is divided into three phases; *Evaluation, Hold* and *Recovery*. The load capacitor is charged during the *Evaluation* phase i.e. when VA rises from zero to maximum.

During the *Hold* time, output voltages are stable and can be used by next logic block(s). The load capacitor discharges into VA as it ramps down to zero during *Recovery* phase. The energy-recovery is shown in Fig.5 Approximately, 90% energy is recycled back after every input cycle and thus achieving.

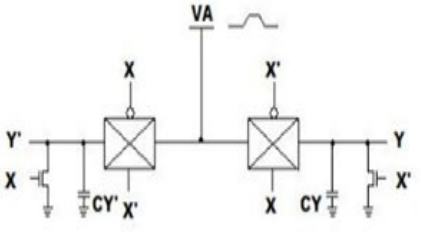


Fig. 4. Adiabatic Amplifier

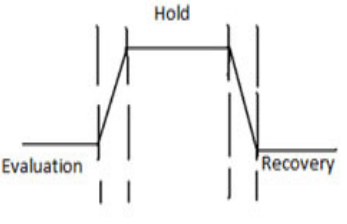


Fig. 5. Energy Recovery

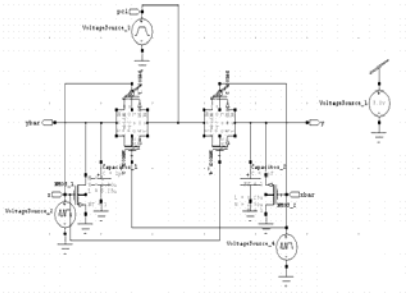


Fig. 6. Circuit Implementation of Adiabatic Amplifier

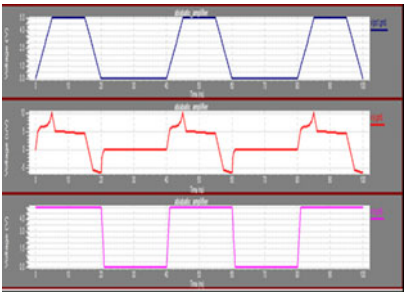


Fig. 7. Output Waveforms of Adiabatic Amplifier

3 Adiabatic Inverters

The different adiabatic families are as follows

- ECRL (efficient charge recovery logic)
- PAL-2N (pass-transistor adiabatic logic with NMOS pull-down configuration)
- CPAL (complementary pass-transistor adiabatic logic)
- CAL(Clocked CMOS adiabatic logic)

3.1 ECRL Adiabatic Inverter

ECRL performs precharge and evaluation simultaneously. It dissipates less energy as it does not use any precharge diode. ECRL has the same circuit structure as cascade voltage switch logic with differential switching. The circuit schematic of a simple ECRL inverter is shown in Figure 8. The signals are dual rail type. Let us assume that 'in' is at high and 'inbar' is at low. When the power-clock signal 'pc' rises from zero to Vdd, the 'out' is grounded through PMOS1. During the hold phase the pc signal is fixed at Vdd and the valid output voltage level is used as input for the next state. After the hold phase, the pc voltage ramps down to zero, 'out' node recycles its energy back to power-clock supply. The input applied is a pulse type signal. If the inverters are to be cascaded then multi-phase clock supply needs to be applied. This will allow the next stage to evaluate logic values in the precharge and evaluation phase when the present stage is in the hold state. A chain of four inverters would require four-phase clocking to efficiently recover the charge [2]. Actual input and output waveforms are shown in Figure 9.

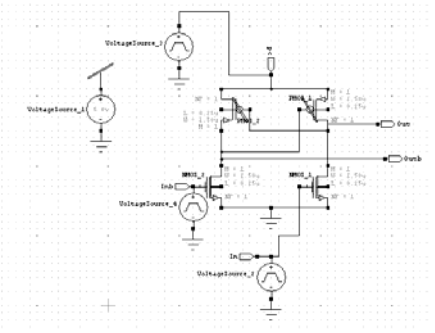


Fig. 8. Circuit Implementation of ECRL Inverter

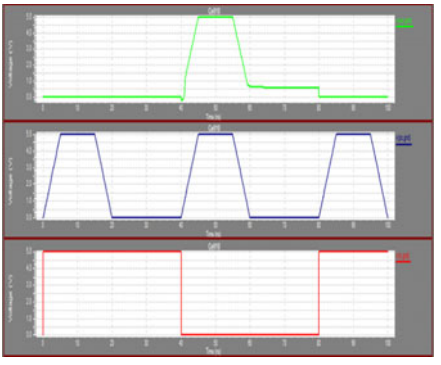


Fig. 9. Input and Output Waveforms of ECRL Inverter

3.2 CAL Inverter

CAL logic circuits can be implemented with integrated single-phase power-clock supply and thus eliminates high complexity of both the logic and the required power-clock generator. The basic circuit structure of CAL inverter is shown in Figure 10 and uses across coupled CMOS inverters made up of PMOS2, NMOS1, PMOS1, and NMOS2. An auxiliary timing control signal 'CX' controls transistors NMOS5 and NMOS3 those are in series with the logic blocks; for inverters the logic blocks are NMOS transistors NMOS6and NMOS4. The CX-enabled transistors allow the use of single power-clock. Actual waveforms are shown in Figure 11. The auxiliary signal CX is a pulse type signal whereas input 'F0' and power-clock supply 'PC' are ramp-type signals. When the CX signal is logic high

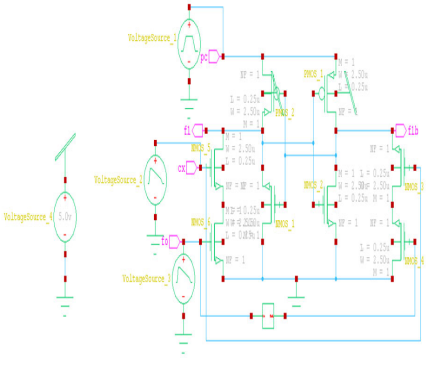


Fig. 10. Circuit Implementation of CAL inverter

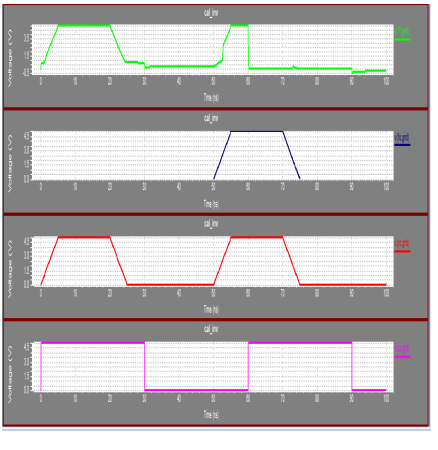


Fig. 11. Input and Output Waveforms of CAL inverter

it allows NMOS6 to invert the input signal and therefore the output signal high i.e. ramp.[1-2]

3.3 CPAL Inverter

The basic structure of the CPAL buffer is composed of two main parts: the logic function circuit and the load driven circuit. The logic circuit consists of four NMOS transistors with complementary pass transistor logic (CPL) function block. The load driven circuit consists of a pair of transmission gates (N1, P1 and N2, P2). The clamp transistors (N3 and N4) ensure stable operation by preventing from loating of the output nodes. Cascaded CPAL gates are driven by the two-phase non-overlap power-clocks. The detailed description on the CPAL circuits can be found in [1,2].

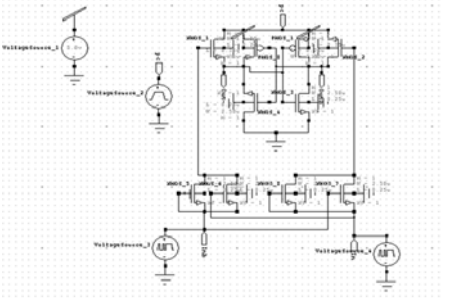


Fig. 12. Circuit Implementation of CPAL Inverter

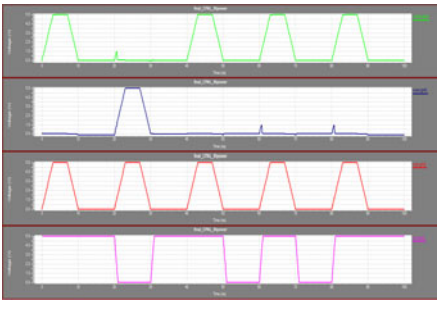


Fig. 13. Input and Output Waveforms of CPAL Inverter

4 Power Analysis and Comparison

Power analysis of different adiabatic inverters with Conventional CMOS inverter was carried out. It was seen that the power dissipation of CAL inverter was the minimum and that of CMOS inverter was the maximum. So the conclusion is that, digital circuits implemented using Adiabatic Logic consumes less energy as compared to that of Conventional CMOS logic.

Power dissipation at 10Mhz of different adiabatic inverters was carried out and comparative analysis with CMOS inverter was done. The power dissipation of CAL, ECRL, CPAL, CMOS inverter is as given below:

Average Power Dissipated of CAL inverter =1.806e-005 w
Average Power Dissipated of CPAL inverter =2.5206e-005w
Average Power Dissipated of ECRL inverter =3.7606-005w
Average Power Dissipated of CMOS inverter =4.5714-005w

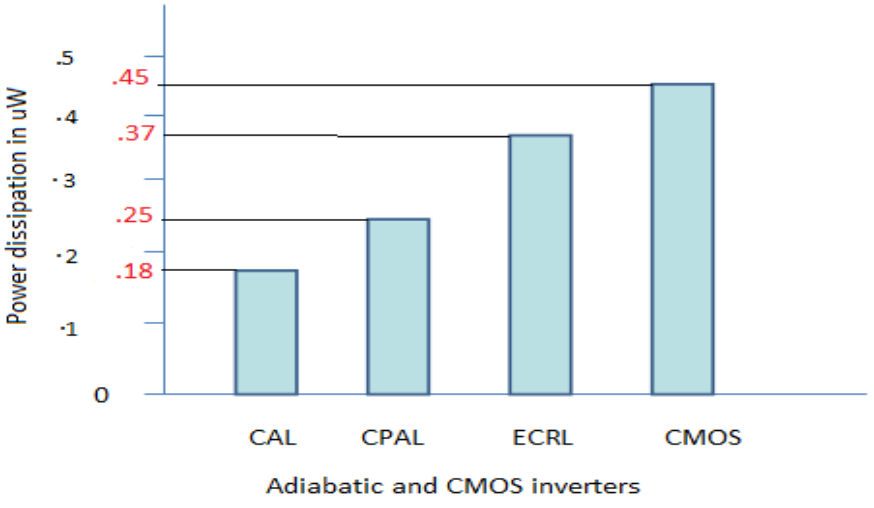


Fig. 14. Power Analysis of Different Adiabatic Inverters with CMOS Inverter

5 Conclusion

Power analysis of different adiabatic inverters with Conventional CMOS inverter was carried out. It was seen that the power dissipation of CAL inverter was the minimum and that of CMOS inverter was the maximum. So the conclusion is that, digital circuits implemented using Adiabatic Logic consumes less energy as compared to that of Conventional CMOS logic.

References

[1] Zhou, D., Hu, J., Wang, L.: The Implementations of Adiabatic Flip-Flops and Sequential Circuits with Power-Gating Schemes. In: International Conference of Circuits and Systems, MWSCA. IEEE (2008)

[2] Khandekar, P.D., Subbaraman, S.: Low Power Inverter and Barrel Shifter Design Using Adiabatic Principal. International Conference Of Advances in Computational Sciences and Technology 3(1), 57–65 (2010); ISSN 0973-6107

[3] Zhang, W., Zhang, Y., Xuhua, S., Hu, J.: Leakage Reduction of Power- Gating Sequential Circuits Based on, Complementary Pass-transistor Adiabatic Logic Circuits. In: International Conference on Innovative Computing and Communication (2010)

[4] Hu, J., Fu, J.: Leakage Dissipation Reduction of Single-Phase Power Gating Adiabatic Sequential Circuits Using MTCMOS. In: International Conference on Electrical and Control Engineering (2009)

[5] Ni, H., Hu, J.: Single-Phase Adiabatic Flip-Flops and Sequential Circuits with Power-Gating Scheme. In: International Conference of ASICON (2009)

[6] Zhou, D., Hu, J., Wang, L.: Adiabatic Flip-Flops for Power-Down Applications. In: IEEE ISIC 2007, pp. 493–496 (2007)
[7] Hu, J.P., Xu, T.F., Xia, Y.S.: Low-power adiabatic sequential circuit with complementary pass-transistor logic. In: IEEE MWSCAS 2005, USA, August 7-10, pp. 1398–1401 (2005)
[8] Moon, Y., Jeong, D.K.: An efficient charge recovery logic circuit. IEEE J. of Solid-State Circuits 31(4) (1996)
[9] Khandekar, P.D., Subbaraman, S., Patil, M.: Low Power Digital Design Using Energy-Recovery Adiabatic Logic. International Journal of Engineering Research and Industrial Applications 1(III), 199–208 (2009)
[10] Bisen, T., Palsodkar, P.: Implementation of Energy Efficient Adiabatic Barrel Shifter. International Journal of Micro &Nano Electronics Circuit & Systems (2010)

A Improved Artificial Fish Swarming Optimization for Economic Load Dispatch with Dynamic Constraints

Potuganti Prudhvi, Adapa Sudarshan, and Chakradhar Bezawada

Department of Electrical Engineering, National Institute of Technology, Warangal
{prudhvi2121991,sudarshan.a.2009}@gmail.com

Abstract. Nature Inspired algorithms have been so far proved as some of the best problem solvers. The Artificial Fish Schooling Algorithm (AFSA) which falls under this category has been deployed to solve the classic Economic Load Dispatch problem. A modified implementation has been illustrated in this paper to enhance the precision and speed of the algorithm.

Keywords: Artificial fish, Optimization, Economic load dispatch, Cost, Swarm, IAFSA (Improvised Artificial Fish Swarm Algorithm).

1 Introduction

Nature has always inspired man with its intuitive and unconquerable design. Our nature has one of the best designs which man has always tried to implement them to his problems. The result was ease along better design and efficiency. Nature inspired algorithms fall under this category. These algorithms have found applications in widely varying domains from tracking of missiles to load dispatch problems. In the present work, efforts have been focused on applying a latest entrant into the field 'Artificial Fish Schooling Algorithm' (AFSA) to a classic power systems problem. The work also focused on tuning the algorithm to the problem's requirements and thus a novel implementation technique has been proposed. Implementation is done in two ways by us. In one way we have considered the search space to be constant (Basic algorithm) and in the other we gradually reduced (Improvised one) the search space with iterations. The results obtained from both ways of implementations are compared and are found to be impressive in case of improvised.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 141–149.
springerlink.com

2 State of Art

2.1 AFSA

The Artificial Fish Swarm Algorithm was first introduced in [1] 2002, the authors designed an optimizing algorithm grounded on self-governing animal, fish. The convergence speed, good global convergence and ease of computation for several standard functions have been illustrated. Later a new algorithm was developed using AFSA and hybridizing cellular learning automata for optimization in continuous and static environments [2], each dimension of search space is assigned to one cell of cellular learning automata and in each cell a swarm of artificial fishes are located which have the optimization duty of that specific dimension. Also irrigation zone water optimization allocation problem is checked with AFSA approach [3].It demonstrated the efficiency of the algorithm in handling complex mathematical problems as well as non-convex water resource management problems**.** Further an AFSA based on chaos search is proposed [4], which has not only overcome disadvantage of easily getting into the local optimum in the later evolution period, but also kept the rapidity of the previous period.

To deal with the problem of low optimizing precision and low speed of convergence a paper [5] proposed an improved algorithm based on meta cognition, which could make self-study by using its knowledge of the surrounding environment. To advance the precision, it changed parameters adaptively resulting better convergence. In [6], an artificial fish algorithm approach for TSP which depicted a precise result and good convergence. An approach using AFSA to solve the multi objective optimization problem with the concept of pareto dominance is proposed in [7].

3 The Basic AFSA Algorithm

Artificial fish schooling algorithm is a nature inspired heuristic technique which simulates the behavior pattern of a fish searching for food in its habitat. Although the behavioral patterns of fishes widely vary for different type of fish, a generalized approach has been proposed in the basic algorithm. It essentially has certain behaviors attributed for all artificial fish simulated in a certain environment. As many as 20 behaviors have been proposed so far. Prominent among them are Swarming, preying, grouping.

3.1 Swarming

It is the behavior of a fish to move in groups. The fish tends to move in groups for it provides protection from a sudden predator attack and also for efficient food searching. A group of fish can better search and survive than individual ones.

Similarly the Artificial Fish (AF) also exhibits the tendency to get into nearby group so that it can get better food along with the group.

For instance, X_i be the current status of a AF. The centroid for the present fish neighborhood is defined as the weighted mean of the fish location in the visibility of the present fish. Once the centroid is calculated, the fish observes the food at particular location and if the food availability is better at centroid (X_c), then it moves to the centroid. As a result of this behavior, all the fish in a selected region gets around a best point nearby.

$$x_c = \frac{\sum_{k=i-v/2}^{i+v/2} n_k x_k}{\sum_{k=i-v/2}^{i+v/2} x_k} \tag{1}$$

3.2 Following / Forwarding

In a practical scenario, some fish tend to behave in a follow fashion i.e. they follow a nearby fish when it understands that the other fish has best food in its location. This is similar to preying and this behavior also results in grouping of fish at a better location.

Let X_i be the location of a current fish. It searches for a best location in its region of visibility and once it finds a best location, it moves a step ahead in that direction. X_b be the best point in region bounded by $X_{i-v/2}$ and $X_{i+v/2}$. Then

$$n_p = n_p - 1 \; \& \tag{2}$$

$$n_{p-S} = n_{p-S} + 1 \quad (b<c) \tag{3}$$

$$n_{p+S} = n_{p+S} + 1 \quad (b>c) \tag{4}$$

Where 'b' is the Best location of the AF. 'p' is the present location. n_p is the number of fish at 'p' and S is the step for the fish.

3.3 Preying

It is the most basic behavior of a Artificial Fish (AF). A fish always watches its surroundings and moves to a place where food is in plenty than the current position. Similarly a AF too looks into its surroundings and moves a step ahead towards a position where objective function is least (minimization problem).

Suppose X_i be the current position of the fish. It randomly selects a position in its visible range X_r ($X_i - V < X_r < X_i + V$). The objective function F_r is evaluated and if $F_i < F_r$ then the fish moves a step towards $X_i + S$. Here V is the visibility of the fish which is a predefined parameter for a specific problem. S is a step predefined for a problem as if V. It should be noted that S<<V so that the behavior of fish is alike the natural ones.

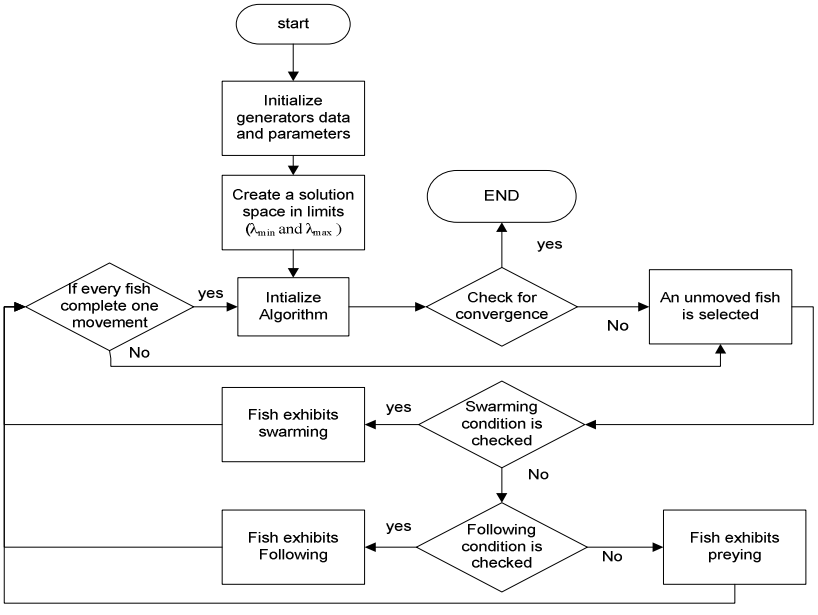


Fig. 1. Basic AFSA Flowchart

4 Economic Load Dispatch Problem

Static Economic Dispatch primarily contemplates on the optimal and economic production of power to satisfy the load demand from time to time and at the same time respecting all the system and security constraints. The Dynamic Economic Dispatch is an extension of the Static Economic dispatch, where an inclusion of the dynamic constraints. The lives of the generating units are mostly guarded by the Thermal and Mechanical gradients/limitations on the turbine, which is turn is formulated as Ramp Rate Limits and Prohibited Zones of operation for each unit. The Generalized problem statement can be formulated as follows,

$$\text{Minimize } Fc \equiv Min \sum_{i=1}^{Ng} F(P_i)$$

$$\sum_{i=1}^{Ng} F(P_i) \equiv \sum_{i=1}^{Ng} \left(a + b_i P_{gi} + c_i P_{gi}^2\right) \frac{\$}{hr} \tag{5}$$

Where a_i, b_i and c_i are the cost coefficients of the particular generator and derived from the generator characteristics. 'i' represents the generator number among the available 'Ng' generators, P_i represents the power of i^{th} generator and 'F' is the fuel cost. The optimality of the fuel and operating cost need to in concurrence with the essential conditions viz.,

$$P_{Gmin} \leq P_{Gi} \leq P_{Gmax} \tag{6}$$

Together satisfying the demand and the transmission losses, i.e.,

$$\sum_{i=1}^{Ng} P_{Gi} = P_d + P_l \tag{7}$$

Where P_d and P_l are the demand and losses in the system.

5 Basic Implementation

Basically, AFSA has many parameters to be set by the programmer like number of fish, visibility, step, choice of behaviors, size of search space etc. The search space factor affects the convergence characteristics, accuracy and speed to a great extent. For ELD, we considered that search space comprised of lambda values between 2.53,2.54,2.55….5.45 randomly placed. A programmer needs to fine tune the algorithm such that the best solution is obtained in least time. The swarm algorithm is discrete in nature i.e. the fish need to be placed at a particular location which is well defined. In our algorithm, the user has the provision to choose number of places after decimal which is symbolized by n. For instance, if n=3 then whole of the calculation in the algorithm takes numbers with three decimal points example: search space contains 2.526,2.527,2.528… 5.450.

PSEUDO CODE for Basic Implementation:

```
pmin=[70 25 19 15 13 14];
pmax=[200 63 49 30 28 35];
for i=1:6
lmin(i)=abc(i,2)+2*abc(i,3)*pmin(i);
lmax(i)=abc(i,2)+2*abc(i,3)*pmax(i);
end
lamin=min(lmin);lamax=max(lmax);
%pool creation
n=2;
pomin=round(lamin*10^n)/10^n;
pomax=round(lamax*10^n)/10^n;
for i=1:(pomax-pomin)*10^n
pp(i,1)=(i*10^-n+pomin);
pp(i,2)=0;pp(i,3)=0;
end
for i=1:(length(pp)/10)  %  fish placement approx 1 per 10
k=round(rand(1)* (length(pp)-1))+1;
pp(k,2)=pp(k,2)+1;
end
for iteration=1:50
for i=1:length(pp)
% In this loop three behaviours are defined based on the  fish position %
end
end
```

6 IAFSA and its Implementation

The basic implementation of AFSA if applied directly takes a very long time to converge. For accuracy of up to 3 digits after decimal, for 6 generator system ELD, the solution converges after 150- 200 seconds and >10 minutes for 4 digits which is a very long time for problem like ELD. Hence a new implementation methodology has been adopted wherein the search starts with very low precision (small search space: 1 digit after decimal). Once the convergence occurs at a precision level, then the search space is updated to neighborhood of the obtained solution with more precision (i.e. 2 digits after decimal). The following table illustrates the implementation.

Iterations	Search space	Convergence point	Precision
1-w1	2.54-5.45	3.38	n=2
w1-w2	3.340-3.420	3.373	n=3
w2-w3	3.3690-3.3770	3.376	n=4

In the above table, w1,w2, w3 are the iteration numbers where the convergence is achieved for particular precision. In the next iteration, precision level is increased and also the search space is narrowed down. The scale of narrowing down is a parameter and problem specific. It must be carefully evaluated to fine tune the algorithm. Here we have taken four points below and above the convergence point and precision is increased i.e. n=n+1. As the fish number is dependent on search space, it is also updated while updating search space. If more than half number of fish in entire space is at a single point, then the solution is said to be converged at that point. Thus the process is repeated till the required precision is obtained.

IAFSA version avoids searching a lot of search space while not compromising on quality of solution and improving the precision. This version also paves the way for finding a solution with varying precision which no algorithm can offer. A solution with 3 digit precision after decimal can be obtained as well as a solution with precision 8 digits after decimal can be obtained with the same algorithm.

7 Results and Discussions

The simulation of ELD with basic AFSA and IAFSA has been done on a standard 6 generator system. The work has been focused on improving the performance of the algorithm. The simulations were done in MATLAB 7.10.0. Tables 1-3 illustrate the results based on basic implementation of AFSA with three different

cases i.e. n=1,2,3 , where n is the number of digits after the decimal point of the solution i.e. lambda. Later the IAFSA has been applied to obtain a precision of 4 digits after the decimal point in solution (Illustrated by Table- IV. Also all other performance parameters are also noted. Both ways of implementations are compared as well as the performance of the IAFSA is compared with other algorithms [8] and is found to be better.

Table 1. n=1; Total Fishes= 3; Search Space: 29 locations; Iterations: 200

Time(seconds)	Fishes converged	Lambda(n=1)	Cost ($/hr)
0.1972	2	3.4	822.8769
0.2103	2	3.4	822.8769
0.1949	2	3.4	822.8769
0.1984	2	3.4	822.8769
0.2231	2	3.4	822.8769

Table 2. n=2; Total Fishes =29; Search Space: 292 locations; Iterations: 200

Time (seconds)	Fishes	Lambda(n=2)	Cost ($/hr)
2.62	27	3.38	811.3814
2.17	28	3.38	811.3814
2.75	27	3.38	811.3814
2.72	27	3.38	811.3814
2.38	27	3.38	811.3814

Table 3. n=3; Total Fishes =292; Search Space:2924 locations ; Iterations : 200

Time(seconds)	Fishes	Lambda(n=3)	Cost ($/hr)
163.51	108	3.376	809.0904
174.03	109	3.376	809.0904
168.39	107	3.376	809.0904
179.30	105	3.376	809.0904
183.09	103	3.376	809.0904

Table 4. IAFSA Results

Time(seconds)	Iterations	Lambda	Cost ($/hr)
1.25	55	3.3760	809.0904
0.75	**42**	**3.3752**	**808.6362**
1.12	65	3.3755	808.8043
1.14	64	3.3756	808.8615
1.03	77	3.3753	808.6898

Table 5. Improved AFSA Results comparison

	EP [8] (Lahouari et al, 2008)	**ISFLA [8]**	**IAFSA**
Fuel Cost ($/hr)	840.21	808.775	808.63
Lambda ($/MWh)	-	3.001903	3.3752
Losses	14.1	10.5465	10.8346
Iterations	-	1	42

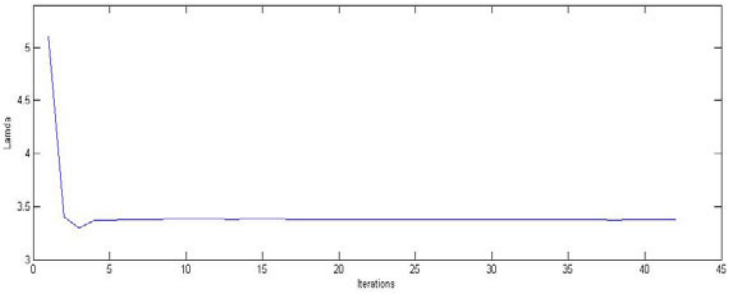


Fig. 2. Converged MATLAB Figure of IAFSA

8 Conclusion

The basic AFSA has been implemented to ELD. It has taken very long time to converge for higher levels of precision hence a new technique deploying dynamic search space reduction technique was implemented. The improved version has yielded better results and the time taken has reduced drastically.

References

[1] Li, X.L., Shao, Z.J., Qian, J.X.: An optimizing method based on autonomous animals: fish-swarm algorithm. Systems Engineering Theory & Practice 22(11), 32–38 (2002)

[2] Yazdani, D., Golyari, S., Meybodi, M.R.: A new hybrid algorithm for optimization based on artificial fish swarm algorithm and cellular learning automata. In: 2010 5th International Symposium on Telecommunications (IST), December 4-6, pp. 932–937 (2010), doi:10.1109/ISTEL.2010.5734156

[3] Gao, Y.F., Chen, Y.D.: The optimization of water utilization based on artificial fish-swarm algorithm. In: 2010 Sixth International Conference on Natural Computation (ICNC), August 10-12, vol. 8, pp. 4415–4419 (2010), doi:10.1109/ICNC.2010.5583509

[4] Ma, H., Wang, Y.: An Artificial Fish Swarm Algorithm Based on Chaos Search. In: Fifth International Conference on Natural Computation, ICNC 2009, August 14-16, vol. 4, pp. 118–121 (2009), doi:10.1109/ICNC.2009.148

[5] Xu, H., Li, R., Guo, J., Wang, H.: An Adaptive Meta-cognitive Artificial Fish School Algorithm. In: International Forum on Information Technology and Applications, IFITA 2009, May 15-17, vol. 1, pp. 594–597 (2009), doi:10.1109/IFITA.2009.352

[6] Wang, J.-P., Hu, M.-J.: A Solution for TSP Based on Artificial Fish Algorithm. In: International Conference on Computational Intelligence and Natural Computing, CINC 2009, June 6-7, vol. 2, pp. 26–29 (2009), doi:10.1109/CINC.2009.72

[7] Jiang, M., Zhu, K.: Multiobjective optimization by Artificial Fish Swarm Algorithm. In: IEEE International Conference on Computer Science and Automation Engineering (CSAE 2011), June 10-12, vol. 3, pp. 506–511 (2011), doi:10.1109/CSAE.2011.5952729

[8] Prudhvi, P., Bhaskar, M.M., Maheswarapu, S.: A Big Bang Big Crunch Algorithm (BBBC) for Economic Dispatch with Network Losses and Dynamic Constraints. In: National Conference on Machines and Power System, vol. 1, pp. 203–208 (February 2011)

A New Approach for Recovering Nodes from Faulty Cluster Heads Using Genetic Algorithm

Farhad Nematy[1] and Naeim Rahmani[2]

[1] Department of Computer Engineering, Azarshahr Branch, Islamic Azad University, Azarshahr, Iran
[2] Young Researchers Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran
{farhad_nematy,naeim.rahmani}@yahoo.com

Abstract. In recent years there has been a growing interest in wireless sensor networks (WSN) applications. Such sensor networks can be used to control temperature, humidity, contamination, pollution etc. Energy consumption and reliability are two critical issues in WSNs. Faults occurring to sensor nodes are common due to lack of power or environmental interference. In this paper recovery of faults in cluster head studied and genetic algorithm is used to recover cluster members to other cluster heads. Simulation results show that proposed genetic algorithm can recover nodes efficiently.

Keywords: wireless sensor network, recovery, clustering, cluster head recovery, genetic algorithm.

1 Introduction

In recent years, the rapid technology development accelerates the prosperity of wireless sensor networks (WSN) [1]. Wireless sensor networks composed of a large number of micro sensors can be an effective tool for gathering data in a variety of environments [2]. These sensor networks usually comprise small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities [3]. Sensor nodes in a WSN may be deployed deterministically or randomly [4]. Also deployment of nodes could be done by robot assistance [5]. Typically WSNs follow the model of a sink node, where sensors relay sensor data streams to the sink node either periodically or based on events [6]. Also a clustering architecture may be used [7, 8, 9]. One of important issues in WSN is energy consumption, so every operation, computation, or interaction becomes important as they inevitably drain the battery. It is critical for designers to take energy preservation into account when designing protocols for wireless sensor networks [10]. Faults may occur in WSN as energy level of nodes decreases and if energy level of a node becomes law it could not send and

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 151–158.
springerlink.com

receive data and is called a dead node. Death of an ordinary node does not affect the work of whole network but if a cluster head dies, all members of that cluster can't send and receive their data to sink. To solve this problem, re-clustering must be done or members of cluster must be recovered by using other cluster heads. Genetic algorithm has many applications in WSNs [4, 11-14]. In this paper a genetic algorithm is presented to recover nodes of faulty cluster to other clusters. The rest of this paper organized as follows: section ii discussed related works and in section iii the proposed genetic algorithm is demonstrated. Section iv presents simulation and experimental results and section v concludes the paper.

2 Related Work

Many researchers studied fault recovery in wireless sensor networks. In [16] FTPASC (Fault Tolerant Power Aware protocol with Static Clustering), a static clustering based protocol, which engages high power sensor nodes for power consuming tasks and as a result extends the network lifetime is presented. Furthermore, another node in each cluster is considered as the organizer. One of the organizer node duties is to setup the network before normal operation. Also, it establishes new connections with mobile nodes, monitors the operation of the cluster-head, and briefly speaking forms a fault tolerant framework for correct functioning of the sensor nodes. In [15], Gupta et al. proposed a fault-tolerant mechanism based on inter-cluster monitoring. When a gateway, say *I* in Figure 1, cannot communicate with some gateway *E*, *I* would consult to its neighboring gateways *F* and *H* whether *E* is failed. If both *F* and *H* confirm its broken link with *E*, gateway *I* would declare the failure of *E*. Then the neighboring cluster heads would further negotiate with each other to transfer the nodes of *E* into their clusters. The main drawback of this approach is that it cannot deal with the failures of multiple gateways because of the mutual inter-cluster-head consulting mechanism. Moreover, the nodes would have to transmit the control messages in a large distance when running the cluster-head based mechanism.

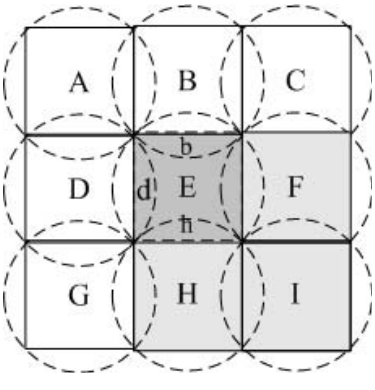


Fig. 1. Illustration of Gupta algorithm

3 Proposed Genetic Algorithm

The proposed genetic algorithm is discussed here and different parts of it illustrated. Every genetic algorithm is composed of few part such as specifying presentation and initial population, fitness function for evaluation each individual in population which is the most important part of genetic algorithm, selection mechanism which tries to select individuals with more fitness value for mating, crossover operation which combines two parents to produce two offsprings and mutation operator that changes an individual to produce a better solution. All of these parts of genetic algorithm are done in genetic algorithm process and iterated until the termination criteria satisfied. Here all of these parts will be discussed.

a. Presentation and initial population

A vector with size of n is used to show recovery process in which n is number of nodes of faulty cluster. Each cell of this vector is number of a cluster head and means that the i_{th} node must be recovered to that cluster head. Population size is set to 200. An instance for representation of solutions is presented in figure 2.

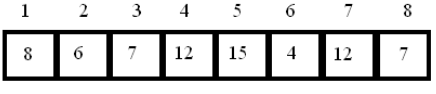


Fig. 2. A representation for solution

b. Fitness function

Fitness function is used to evaluate each solution by considering distance of node and cluster head, residue energy of cluster head, number of members of cluster head and traffic in cluster head. Equation 1 illustrates presented fitness function.

$$\text{Fitness (I)} = \alpha \times (1/\text{distance (I, Ch)}) + \beta \times (\text{energy (Ch)}) + \gamma \times (1/\text{members (Ch)}) + \eta \times (1/\text{traffic(ch)}) \quad (1)$$

c. Selection mechanism

For selecting individuals to transfer to mating pool a tournament selection mechanism is used. K individual is selected randomly and the individual with greater value is selected to transfer to mating pool and this process continues 200 times.

d. Crossover operation

Different kind of crossover exists and here two point crossover is used with crossover probability equal to 0.8. Two random numbers is generated that shows

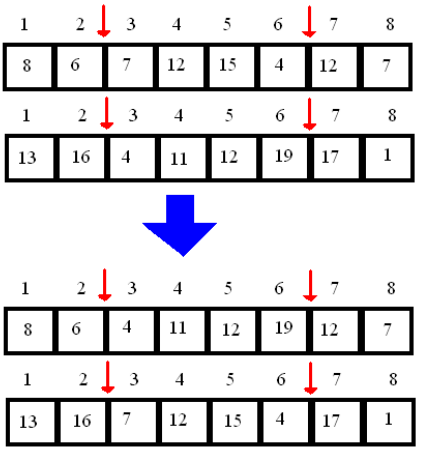


Fig. 3. Crossover process

the points for crossover. After that, the gens between these two points exchanges. The process of crossover is depicted in figure 3.

e. Mutation operation

For mutating each chromosome a random point is selected and after that a random number between 1 and number of cluster heads is generated. Mutation probability set to 0.5. Figure 4 depicts mutation process.

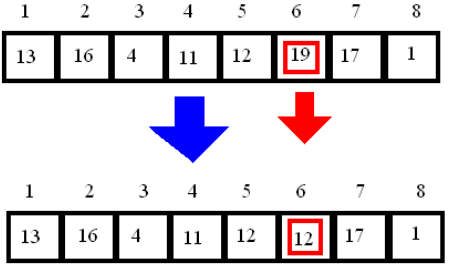


Fig. 4. Mutation process

f. Termination criteria

The process of genetic algorithm will continue until the termination criteria satisfied. Here constant number of iterations is selected and number of iterations set 200.

4 Simulation

MATLAB 7 is used to simulate the proposed algorithm. 400 nodes distributed in a 200m * 200m field randomly. Each node has 1 joule initial energy and 20 cluster heads are responsible to gather information and send them to sink. Clustering is done considering distance of each node from cluster head. Fault injection done by marking a cluster head as faulty at a random time. After that genetic algorithm is used to recover the nodes. Population size is set to 200 and number of iterations considered 200. Other simulation parameters are presented in table 1.

Table 1. Simulation parameters

value	parameter
50nj	E_{TX}
50nj	E_{RX}
10pj	E_{fs}
0.013pj	E_{amp}
5nj	E_{DA}
6400	Packet size
200	Control Packet size
2m	Sensing range

Different experiments done using different combination of parameters. Figure 5 shows the recovery of nodes using all 3 parameters. Comparison of all combinations of parameters with considering number of cluster heads is shown in figure 6. blue line shows using only distance from cluster head(α=1,β=0,γ=0, η=0), Green line shows using only energy(α=0,β=1,γ=0,η=0), black line shows using only number of members (α=0,β=0,γ=1,η=0) and red line shows using all four parameters. Comparison show that using all of parameters can result in load balancing and proper assign of nodes to cluster heads.

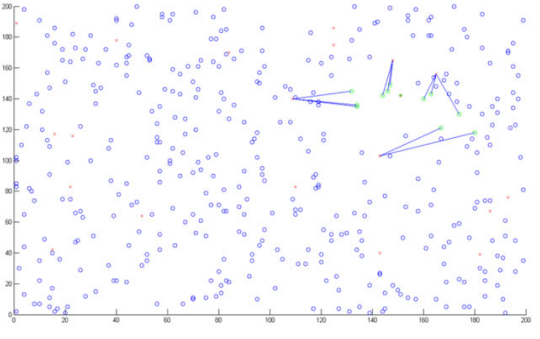


Fig. 5. Recovery of nodes using 3 parameters

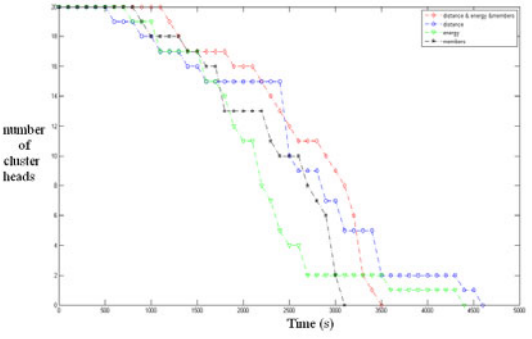


Fig. 6. Comparison of number of cluster heads, using different parameters

The evolution process of genetic algorithm presented in figure 7.

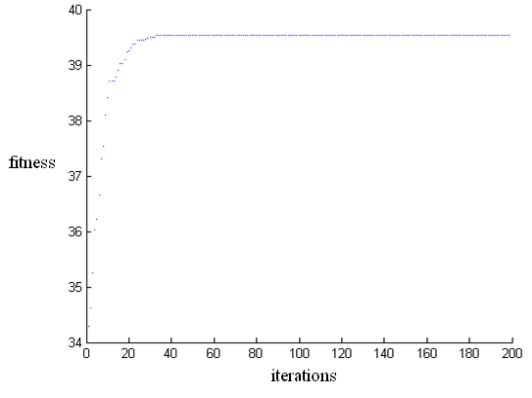


Fig. 7. Evolution of fitness value

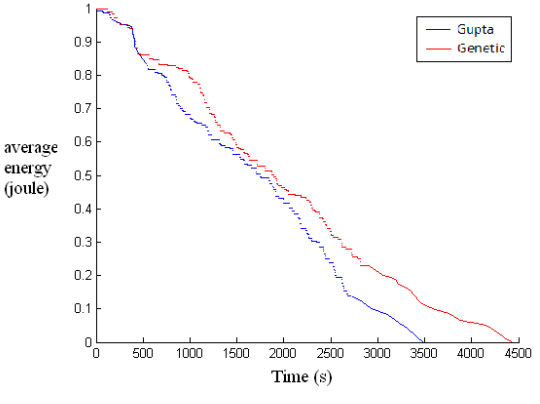


Fig. 8. Comparison of energy consumption

The proposed algorithm compared to Gupta [15] and results show that proposed algorithm can reduce energy consumption for node recovery and because it uses three parameters to recover, the overall energy consumption of network reduces. Figure 8 shows this comparison.

5 Conclusion

In this paper a genetic algorithm for recovering nodes of faulty cluster proposed that tries to assign nodes to other cluster heads by considering distance, residue energy and number of members. Simulations done to evaluate the proposed algorithm and results shows that it can outperform all previous works. As future work, adding more parameters like traffic and message delivery can be considered.

References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks, 393–422 (2002)

[2] Mainwaring, A., Polastre, J., Szewczyk, R., et al.: Wireless sensor networks for habitat monitoring. In: Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications, September 28, pp. 88–97. ACM press, Atlanta (2002)

[3] Shao, K., Bang, H.-W.: Data fusion in wireless sensor network: A survey. Journal of Chongqing University of Posts and Telecommunications: Natural Science 18(1), 53–59 (2006)

[4] Nematy, F., Rahmani, N., Yagouti, R.: An Evolutionary Approach for Relocating Cluster Heads in Wireless Sensor Networks. In: International Conference on Computational Intelligence and Communication Networks (CICN), pp. 323–326 (2010)

[5] Nematy, F., Rahmani, A.M., Teshnelab, M., Rahmani, N.: Ant Colony Based Node Deployment and Search in Wireless Sensor Networks. In: International Conference on Computational Intelligence and Communication Networks (CICN), pp. 363–366 (2010)

[6] Culler, D., Estrin, D., Srivastava, M.: Overview of Sensor Networks. Computer, 4149 (2004)

[7] Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)

[8] Heikalabad, S.R., Navin, A.H., Mirnia, M., Ebadi, S., Golesorkhtabar, M.: EBDHR: Energy Balancing and Dynamic Hierarchical Routing algorithm for wireless sensor networks. IEICE Electron. Express 7(15), 1112–1118 (2010)

[9] Golsorkhtabar, M., Hosinzadeh, M., Heydari, M.J., Rasouli, S.: New Power Aware Energy Adaptive protocol with Hierarchical Clustering for WSN. International Journal of Computer and Network Security 2(4), 38–40 (2010)

[10] Li, L., Dong, S.-S., Wen, X.-M.: An energy efficient clustering routing algorithm for wireless sensor networks. The Journal of China Universities of Posts and Telecommunications 13(8), 71–75 (2006)
[11] Bhondekar, A.P., Vig, R., Singla, M.L., Ghanshyam, C., Kapur, P.: Genetic Algorithm Based Node Placement Methodology for Wireless Sensor Networks. In: Proceedings of International MultiConference of Engineers and Computer Scientists, IMECS 2009, Hong Kong, March 18-20, vol. I (2009)
[12] Jin, S., Zhou, M., Wu, A.S.: Sensor network optimization using a genetic algorithm. In: 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL (2003)
[13] Jourdan, D.B., de Weck, O.L.: Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In: IEEE Semiannual Vehicular Technology Conference, Milan, Italy (May 2004)
[14] Ferentinos, K.P., Tsiligiridis, T.A.: Adaptive design optimization of wireless sensor networks using genetic algorithms. Computer Networks 51, 1031–1051 (2007)
[15] Gupta, G., Younis, M.: Fault-Tolerant Clustering of Wireless Sensor Networks. IEEE Wireless Communications and Networking (2003)
[16] Khadivi, A., Shiva, M.: FTPASC: A Fault Tolerant Power Aware Protocol with Static Clustering for Wireless Sensor Networks. In: Wireless and Mobile Computing, Networking and Communications, pp. 397–401 (2006)

A Fuzzy Clustering Method to Minimize the Inter Task Communication Effect for Optimal Utilization of Processor's Capacity in Distributed Real Time Systems

P.K. Yadav[1], P. Pradhan[2], and Preet Pal Singh[2]

[1] Central Buildingn Research Institute, Roorkee
prd_yadav@rediffmail.com
[2] Department of mathematics & Statistics,
Gurukula Kangri Vishwavidyalaya Hardwar (U.K.)
ppradhan14@gmail.com, ppsingh007@yahoo.com

Abstract. A distributed processing System is a collection of heterogeneous processors which requires systematic assignment of a set of "m" tasks T = {t_1, t_2….t_m} of a program to a set of "n" processors P = {p_1, p2….p_n}, (where, m >> n) to achieve the efficient utilization of available processor's capacity. If this step is not performed properly, an increase in the number of processors may actually result in a decrease in the total system throughput. The Inter-Task Communication (ITC) time is always the most costly and the least reliable factor in distributed processing environment. This paper deals a heuristic task allocation model which performs the proper allocation of task to most suitable processor to get an optimal solution. A fuzzy membership functions is developed for making the clusters of tasks with the constraints to maximize the throughput and minimize the parallel execution time of the system.

Keywords: Inter Task Communication Time, Execution Time, Distributed Real Time System, Fuzzy Membership.

1 Introduction

Although computer speed has been increased by several order of magnitude in recent decades the demand for processing capacity increases at an even faster pace. The required processing efficiency for many real time applications cannot be achieved with single processor system. One approach to solve this problem is to use Distributed processing System (DPS) so that an application program can process on multiple processors to achieve the real time computation. Task partitioning and task allocation are two major steps in the designing of DPS. If these steps are not properly implemented and an increase the number of processors in the system may resulted in a decrease of the total throughput of the system [1].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 159–168.
springerlink.com

Meanwhile, the traditional notions of best-effort and real-time processing have fractured into a spectrum of processing classes with different timeliness requirements [2], [3], [4], [5]. Many systems are hard and missing deadline is catastrophic [6], [7], [8], whereas in soft real-time system occasional violation of deadline constraints may not result in a useless execution of the application or calamitous consequences, but decreases utilization [9].

Dar-Tzen Peng et al [10] have derived optimal task assignments to minimize the sum of task execution and communication costs with the branch-and-bound method and evaluated the computational complexity of this method using simulation techniques. *Tzu*-Chiang Chiang et al [11] has been reported the traditional methods for optimization and give exact answers but often tends to get stuck on local optima [11]. The dynamic task scheduling considering the load balancing issue is an NP-Hard problem has also been discussed by Hans-Ulrich Heiss et al and Yadav et al [12] [21]. Consequently, another approach is needed when the traditional methods cannot be applied. Modern heuristics are general purpose optimization algorithms discussed by Page, A.J et al [13]. Their efficiency or applicability is not tied to any specific problem-domain. Multiprocessor scheduling methods can be divided into list heuristics and Meta heuristics. Some of the Meta heuristics include Simulated Annealing algorithms, Genetic Algorithms have been reported by [14] [15] [16] [17][18] [19] [20]. Particle Swarm Optimization PSO yields faster convergence when compared to GA, because of the balance between exploration and exploitation in the search space and reported by Chunming Yang et al and Van Den Bergh et al [22] [23]. Recently Yadav et.al [24] discussed a task allocation model for reliability and cost optimization in distributed computing system. In the present study we have considered two important parameters such as Execution Time (ET) and Inter task Communication (ITC) time for task allocation and also present a combinational approach of fuzzy logic for making the clusters of heavily communicated tasks to minimizing ITC time and developed a heuristic model for systematic allocation of tasks to maximize the throughput and minimize the Parallel Execution Time (PET) of the system.

2 Objective Function for Task Allocation

In this section we shall first describe two important parameters ET and ITC minimizes the Parallel Execution Time (PET) of the system. An allocation of tasks to processors is defined by a function A_{alloc} from the set T of tasks to the set P of processors such that:

$A_{alloc}: T \rightarrow P$ *,where* $A_{alloc}(i)=j$
if task t_i *is assigned to processor* p_j , $1 \leq i \leq m, 1 \leq j \leq n$.

For an assignment, the task set (TS_j) of a processor can now be defined as the set of tasks allocated to the processor [13]

$$TS_j = \{i: A_{alloc}(i) = j, \quad j = 1, 2 \ldots n\}$$

2.1 Inter Task Communication Time

Inter Task Communication (ITC), c_{ik} which is depends upon the data units exchanged between the task t_i and t_k during the process of execution when they are assign to different processors. If a group of tasks or cluster are assigned to the same processor, the ITC between them is zero. The inter-task communication times for each processor of a given allocation are calculated as:

$$ITC\ (A_{alloc})_j = \sum_{\substack{1\le i\le m \\ i+1\le j\le m \\ A_{alloc}(i)=j\neq A_{alloc}(k)}} c_{i,k} \tag{1}$$

2.2 Execution Time

Execution Time (ET) et_{ij} where $(1 \le i \le m \ \&\ 1 \le j \le n)$ of each task t_i depends on the processor p_j to which it is assigned. If a task t_k is not in position to execute on processor p_l due to absence of some resources the et_{kl} is as (∞) infinite. The execution time of each processor for an allocation is calculated using the following equation as:

$$ET(A_{alloc})_j = \sum_{\substack{1\le i\le m \\ i\in TS_J}} et_{i,A_{alloc}(i)} \tag{2}$$

2.3 Parallel Execution Time

The Parallel Execution Time (PET) is a function of the heaviest aggregate computation to be performed by each processor. The PET for a given assignment A_{alloc} is defined conservatively (assuming that computation cannot be overlapped with communication) as shown below:

$$PET(A_{alloc}) = \max_{1\le j\le n}\left\{ET(A_{alloc})_j + ITC(A_{alloc})_j\right\} \tag{3}$$

3 Proposed Method

Initially average load must be assigned to each processor p_j is calculated by using the equation 4 and total load to be assigned on the system has been evaluated and 5.

$$L_{avg}(P_j) = \frac{W_j}{n} \tag{4}$$

$$T_{lod} = \sum_{1}^{n} L_{avg}(P_j) \tag{5}$$

Where $W_j = \sum_{1\le i\le m} et_{i.j}$ $j = 1,2, \ldots n$

Because strictly balanced load distribution may not be possible to achieve, a system is considered to be balanced if the load on each processor is equal to the average load, within a reasonable tolerance. A tolerance of 20-35% of average load is generally chosen. Also, we assume that the number of program tasks is much larger than the number of processors, so that no processor remains idle.

3.1 Processors Clustering

Processors clustering attempts to identify a group of processors, which can be treated as single units. These groups of processors are clustered together. In the present study, the attempt has been made to form the cluster of processors based on their speed. Usually number of processor clusters should be equal to the number of task clusters so that one to one mapping may results.

Abdelzer and Shin [6] have defined the attraction force ($B_{ij}/\mu_i+\mu_j$) for the clustering of the processors where B_{ij} is the Bandwidth of the link connecting two processors P_i and P_j of μ_i and μ_j speed respectively. This work considers another aspect for the formulations of the processors clusters a fuzzy logic is applied to define the membership function of the processors and is sub-sequent used to form the processors. The fuzzy functions will try to keep processors of same processing speed in the same clusters Membership function is defined as follow.

$$\mu(P_i) = \frac{1}{1 + diff\,(S_i - S)} \text{ where } \mathrm{diff}(S_i - S) = |S_i - S| \tag{6}$$

Si = Sum of communication time of all tasks on the processor Pi

S=it is maximum value in Si $\quad 1 \leq i \leq n$

Using the above membership function each processors of the same DRTS will get a membership value, which lies between 0 and 1 and will help for formulation the processors cluster. We categories processors cluster in four categories according to the member value.

1st .Very high speed processor	[0.0, 0.4]
2nd .High speed processors	[0.4, 0.7]
3rd .Slow speed processors	[0.7, 0.9]
4th .Very slow speed processor	[0.9, 1.0]

3.2 Tasks Clustering

Tasks are clustered based on their communication time. Those tasks are heavily communicating are clustered together to reduce communication time and applied the same fuzzy membership function to categories the task in four categories. Thus each task of will receive a membership value is defined as:

$$\mu(T_i) = \frac{1}{1 + diff\,(C_i - C)} \tag{7}$$

where C_i is the sum of total communication time if i^{th} task say t_i

and C is the maximumu value of C_i $1 \le i \le m$

The tasks are categorise in following four categories base on the value of membership function

1st Very hard tasks	[0.0, 0.4]
2nd High tasks	[0.4, 0.7]
3rd Easy tasks	[0.7, 0.9]
4th Very easy tasks	[0.9, 1.0]

3.3 The Allocation Algorithm

The mapping of the tasks clusters to processors cluster takes place according to the following algorithm

(1) *AVERAGE_LOAD(:)// to be allocated to the processors and System*

Determine the $L_{avg}(P_j)$ to be assigned on processor p_j with 20 to 25% Tolerance Factor (TF) by using the equation (1) and T_{lod} to be assigned to the system by using the equation (2).

(2) *PROCESSOR_CLUSTER():// form the clusters of processors of DRTS.*

{

(a) Estimate the fuzzy membership value for all the processors starting with first processor.

(b) Categories processors in same category which have membership function between the given same ranges. Do it for all processors.

}

(3) *TASK_CLUSTER()://form the clusters of tasks.*

{

(a) Estimate the fuzzy membership value for all the tasks in respect to ITCT starting with first task.

(b) Categories tasks in same category which have membership function between the given same ranges, Do it for all tasks.

}

(4) *TASK_MAPPING():// Map the task clusters to the processors clusters using the scheduling policies –*

Very hard tasks ← to very high speed processors
Hard tasks ← to high speed processors
Easy tasks ← to slow processors
Very easy tasks ← to very slow processors

If $ET\,(A_{alloc})_j < L_{avg}(P_j)$
then
assign the task t_i to processor P_j

else
assign the task t_i to processor P_k of same category
endif

4. Results and Discussions

Let in a DRTS a program consists of a set of eight tasks T={t1,t2,t3,t4,t5,t6,t7,t8} and a set of four processes P={p1,p2,p3,p4} interconnected by communication links having different capabilities and inter- task communication between any two tasks are known and it is given in the ETM() and ITCM() respectively.

Input: m = 8, n = 4

ETM(,)

	p_1	p_2	p_3	p_4
t_1	6.00	3.00	5.00	2.00
t_2	4.00	2.00	3.00	2.80
t_3	3.00	1.00	2.00	5.00
t_4	2.00	2.00	1.00	6.00
t_5	3.00	3.00	2.00	3.00
t_6	4.00	3.00	6.00	5.00
t_7	5.00	6.00	7.00	5.00
t_8	3.00	2.00	5.00	2.00

ITCTM(,)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
t_1	0.00	3.00	4.00	2.00	0.00	3.00	2.00	1.00
t_2	3.00	0.00	2.00	3.00	5.80	2.00	2.00	2.00
t_3	4.00	2.00	0.00	3.00	2.70	2.00	1.00	5.00
t_4	2.00	3.00	3.00	0.00	3.00	3.00	3.00	3.00
t_5	0.00	5.80	2.70	3.00	0.00	2.50	5.00	1.00
t_6	3.00	2.00	2.00	3.00	2.50	0.00	2.00	3.00
t_7	2.00	2.00	1.00	3.00	5.00	2.00	0.00	4.00
t_8	1.00	2.00	5.00	3.00	1.00	3.00	4.00	0.00

Determine the $L_{avg}(P_j)$ to be assigned on processor p_j with 20% TF by using the equation (1) and T_{lod} to be assigned to the system by using the equation (2).

Load to be allocated	p_1	p_2	p_3	p_4	Total Load
Average Load	10.00	7.33	10.33	10.27	37.93
20% TF	2.00	1.47	2.07	2.05	7.59
Average Total Load	12.00	8.80	12.40	12.32	45.52

Membership function value for processor clustering are determine for categories the processes as follow:

Very High Speed Processor	High Speed Processor	Slow Speed Processor	Very Slow Speed Processor
p_2	p_1	p_4	p_3
0.10	0.50	0.83	1.00

Membership function value for task clustering is computed for making the categories the tasks as follow:

Very Hard Task		Hard Task		Easy Task		Very Easy Task	
t_1	0.17	t_7	0.50	t_3	0.77	t_4	1.00
t_6	0.29	t_8	0.50	t_2	0.83	t_5	1.00

As per the allocation algorithm, the tasks cluster are mapped onto processor cluster as follows:

Very hard tasks → very high speed processors.
Hard tasks → high speed processors.
Easy tasks → slow speed processors.
Very easy tasks → very slow processors.

It should be noted that after each assignment of a task to a processor the aggregate ITCT associated with that processor usually decreases. Table 1 shows the resulting allocation where it indicates the order in which each task is assigned to a processor. From the Table 1, it can be seen that maximum busy time of the system is 37.00 which associated with the processor p_2. Fig 1 show the total allocated and calculated load from the figure it is concluded that the load assigned to the processor is within the limit. Fig 2 depicted the through put and services rate, of the processor from the Fig. It is concluded that both are the ideally linked.

Table 1 Result of the model

Task Cluster	Processor	ET	ITCT	PET	Services Rate	Throughput of the processor
t_1+ t_6	p_2	8.00	29.00	37.00	0.027	0.054
t_7+ t_8	p_1	6.00	26.50	32.50	0.031	0.062
t_3+ t_2	p_4	3.00	33.70	36.70	0.027	0.054
t_4+ t_5	p_3	5.80	28.20	34.00	0.029	0.059

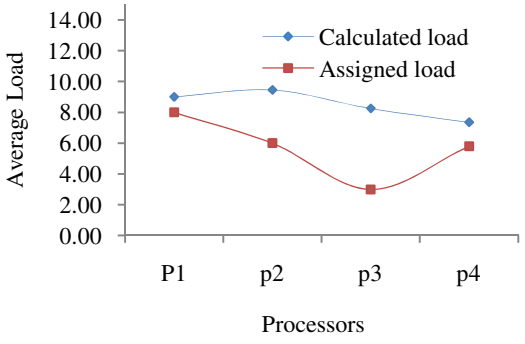


Fig. 1. Total allocated and calculated load

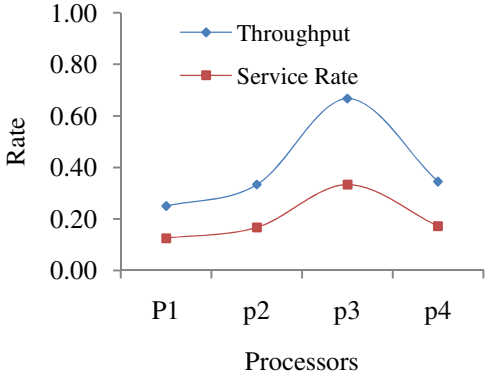


Fig. 2. Throughput & Service rate

Cluster based load partitioning and assignment is used for real time applications. A new fuzzy approach is applied to form the clusters. The proposed approach has the potential to optimise the use of processors capability and supports for system heterogeneity. The approach considers the communication aspect in the clusters formations as it incurs more overhead. This is also a realistic approach as the other algorithms, based on the on same, uses the prior knowledge of the execution time and inters task communication time.

Acknowledgment. Authors are thankful to Director, CBRI, the VC Gurukula Kangri Vishwavidyalaya Hardwar and Prof. M.P. Singh Head Faculty of Science for his kind permission to publish this paper as the part of my Ph.D. work.

References

[1] Chu, E.W., Lee, D., Iffla, B.: A Distributed processing system for naval data communication networks. In: Proceeding AFIPS Nat. Comput. Conference, vol. 147, pp. 783–793 (1978)

[2] Deng, Z., Liu, J.W., Sun, S.: Dynamic scheduling of hard real-time applications in open system environment, Tech. Rep., University of Illinois at Urbana-Champaign (1993)
[3] Buttazzo, G., Stankovic, J.A.: RED: robust earliest deadline scheduling. In: Proc. 3rd Intl. Workshop Responsive Computing Systems, Lincoln, pp. 100–111 (1993)
[4] Petters, S.M.: Bounding the execution time of real-time tasks on modern processors. In: Proc. 7th Intl. Conf. Real-Time Computing Systems and Applications, Cheju Island, pp. 498–502 (2000)
[5] Zhu, J., Lewis, T.G., Jackson, W., Wilson, R.L.: Scheduling in hard real-time applications. IEEE Softw. 12, 54–63 (1995)
[6] Taewoong, K., Heonshik, S., Naehyuck, C.: Scheduling algorithm for hard real-time communication in demand priority network. In: Proc. 10th Euromicro Workshop Real-Time Systems, Berlin, Germany, pp. 45–52 (1998)
[7] Liu, C.L., Layland, J.W.: Scheduling algorithms for multi-programming in a hard-real-time environment. J. ACM 20, 46–61 (1973)
[8] Babbar, D., Krueger, P.: On-line hard real-time scheduling of parallel tasks on partitionable multiprocessors. In: Proc. Intl. Conf. Parallel Processing, pp. 29–38 (1994)
[9] Lifeng, W., Haibin, Y.: Research on a soft real-time scheduling algorithm based on hybrid adaptive control architecture. In: Proc. American Control Conf., Lisbon, Portugal, pp. 4022–4027 (2003)
[10] Dar-Tzen, P., Shin, K.G., Abdelzaher, T.F.: Assignment and Scheduling Communicating Periodic Tasks in Distributed Real-Time Systems. IEEE Transactions On Software Engineering 23(12), 745–758 (1997)
[11] Chiang, T.-C., Chang, P.-Y., Huang, Y.-M.: Multi-Processor Tasks with Resource and Timing Constraints Using Particle Swarm Optimization. IJCSNS International Journal of Computer Science and Network Security 6(4), 71–77 (2006)
[12] Heiss, H.-U., Schmitz, M.: Decentralized Dynamic Load Balancing: The Particles Approach. Information Sciences 84(2), 115–128 (1995)
[13] Elsadek, A.A., Earl Wells, B.: A Heuristic model for task, allocation in heterogeneous distributed computing systems. The International Journal of Computers and Their Applications 6(1), 1–36 (1999)
[14] Page, A.J., Naughton, T.J.: Framework for task scheduling in heterogeneous distributed computing using genetic algorithms. In: 5th Artificial Intelligence and Cognitive Science Conference, Ireland, pp. 137–146 (2004)
[15] Page, A.J., Naughton, T.J.: Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing. In: Proceedings of the 19th Dynamic Task Scheduling with Load 487 IEEE/ACM International Parallel and Distributed Processing Symposium, Denver, USA, pp. 1530–2075 (2005)
[16] Wu Annie, S., Yu, H., Jin, S., Lin, K.-C., Schiavone, G.: An Incremental Genetic Algorithm Approach to Multiprocessor Scheduling. IEEE Transactions on Parallel and Distributed Systems 15(9), 824–834 (2004)
[17] Zomaya, A.Y., Teh, Y.H.: Observations on using genetic algorithms for dynamic load-balancing. IEEE Transactions on Parallel and Distributed Systems 12(9), 899–911 (2001)
[18] Edwin, S.H., Hou, N.A., Hong, R.: A genetic algorithm for multiprocessor scheduling. IEEE Transactions on Parallel and Distributed Systems 5(2), 113–120 (1994)

[19] Manimaran, G., Siva Ram Murthy, C.: A Fault-Tolerant Dynamic Scheduling Algorithm for Multiprocessor Real-Time Systems and Its Analysis. IEEE Transactions on Parallel and Distributed Systems 9(11), 1137–1152 (1998)

[20] Chen, R.-M., Huang, Y.-M.: Multiprocessor Task Assignment with Fuzzy Hopfield Neural Network Clustering Techniques. Journal of Neural Computing and Applications 10(1), 12–21 (2001)

[21] Yadav, P.K., Singh, M.P., Kumar, H.: Scheduling Algorithm: Tasks Scheduling Algorithm for Multiple Processors with dynamic Reassignment. Journal of Computer System, Network and Communication, 1–9 (2008)

[22] Yang, C., Simon, D.: A new particle swarm optimization technique. In: Proceedings of the International Conference on Systems Engineering, pp. 164–169 (2005)

[23] Van Den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Information Sciences, 937–997 (2006)

[24] Yadav, P.K., Singh, M.P., Sharma, K.: Task Allocation Model for Reliability and Cost Optimization in Distributed Computing System. International Journal of Modelling, Simulation and Scientific Computing (IJMSSC) 2(2), 1–19 (2011)

Novel Binary PSO for Continuous Global Optimization Problems

Pinkey Chauhan, Millie Pant, and Kusum Deep

Indian Institute of Technology Roorkee, India
pinkeychauhan030@gmail.com, millifpt@iitr.ernet.in,
kusumfma@iitr.ernet.in

Abstract. The present study proposes a novel variant of binary PSO employing gompertz function for generating binary numbers with an additional benefit of controlling its parametrs so as to induce an combined effect of sigmoid as well as linear function. The corresponding variant is termend as Gompertz Binary Particle Swarm Optimization (GBPSO). The proposed GBPSO is tested on a a well known suite of 5 continuous benchmark fuctions. Numerical results indicate the competence of the proposed mechanism.

Keywords: Binary PSO, Gompertz function, Paramters.

1 Introduction

Particle Swarm optimization (PSO), since its development by Kennedy and Eberhart in 1995 [3], has emerged as a potential global optimizer for solving real world complex problems, with an easily implementable and less complicated code. Primarily, it was introduced for solving continuous optimization problems, but later on a discrete version was proposed by Kennedy and Eberhart [2], to tackle complex problems having discrete feasible space. The continuous and binary version share the same searching process upto termination of algorithm, except the change of position update equation which includes a different strategy to generate required binary or discrete numbers. In Standard Binary PSO proposed by [2],binary patterns are generated by employing a sigmoid function having a range within [0,1] with an input from basic velocity update equations. Researchers have proposed several variants of BPSO ([4], [5], [6] etc). Interestingly, in most of the versions basic sigmoid function is used for generating the binary numbers. This paper proposes a new Binary PSO, where Gompertz function is employed as a binary number generator with an additional benefit of controlling its parameters so as to induce the effect of a sigmoid function and a linear function into signle one. The basic sigmoid function as employed in standard BPSO has the well exploration capability in initial stages but it may get stuck in a local attractor during the later stages due to lack of diversity. The Linear function has fair chances to produce a

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 169–176.
springerlink.com

diversified popoulation even in later stages. Here, in our study we try to combine the positive sides of both the function by controlling parameters of Gompertz function. The rest of this paper is organized as follows: Section 2 presents a brief introduction of Standard Binary PSO. The newly proposed Binary PSO is described in Section 3. The computational studies are carried out in Section 4. Discussion of results is performed under Section 5. The paper ends with concluding remarks in Section 6.

2 Standard Binary PSO

Here we briefly introduce continuous version of PSO and then further extend this to explain standard Binary PSO.

2.1 Standard Continuous PSO

The basic PSO update equations are as:

$$v_{id}(t+1) = w^{*}v_{id}(t) + c_1 r_1 (p_{best}(t) - x_{id}(t)) + c_2 r_2 (p_{gbest}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = v_{id}(t) + x_{id}(t) \quad (2)$$

Where $d = 1, 2\ldots D$ stands for the dimension of search space and $i = 1, 2\ldots S$ represents the size of the swarm. Parameters "c_1" and "c_2" are constants, called cognitive and social scaling parameters respectively which quantifies the contribution of personal and social experiences. The parameters " r_1" and "r_2" are used to maintain diversity of the population and are uniformly distributed in the range [0,1]. $x_{id}(t)$, $v_{id}(t)$ are defined respectively as the position and velocity of d^{th} dimension of i^{th} particle at t^{th} iteration. $p_{best}(t)$ is the best position of i^{th} particle in d^{th} dimension and is evaluated based on particle's personal experience. $p_{gbest}(t)$ is the best position found by whole swarm so far; w is inertia weight, large values of w facilitate exploration, with increased diversity, while a small value promotes local exploitation. Also, there is a parameter called *Vmax*, which bounds the velocities of particles and prevents them from leaving the feasible search space and jumping into infeasible space.

2.2 Standard Binary PSO (BPSO)

In standard BPSO, the basic searching process of particles is same as in continuous case, except a change in position update equation which turns to a binary number generator in case of binary PSO [3]. The whole swarm upgrade its moves using velocity update eq. (1) as in continuous PSO while the particle's positions are decided based on following process. The Binary PSO needs a function to generate values 0 and 1 or map all the positions in the range [0,1]. In standard Binary PSO the sigmoid function is applied to determine the probability to generate

binary numbers. The velocity update equation is fed as input to this function so that particles could be benefited with its previous positions and previous memory. The binary numbers are generated using sigmoid function as:

$$sigm(V_{id}) = \frac{1}{1+\exp(-V_{id})}$$

where V_{id} is the velocity obtained using basic update equation. Now the position update equation turns out as probabilistic update equation

$$x_{id}(t+1) = \begin{cases} 1 & if \;\; U(0,1) < sigm(V_{id}(t)) \\ 0 & otherwise \end{cases} \tag{3}$$

$U(0,1)$ is a quasi random number selected from a uniform distribution in the range [0,1]. It is evident from equation (3) that x_{id} will remain 0 if $sigm(V_{id}) = 0$. This happens when either $v_{id} < -10$ or $v_{id} > 10$ therefore to overcome this situation, it is suggested use velocity clamping with Vmax = 4.

3 GMPSO: Proposed Binary PSO

Gompertz Based Binary PSO*:** The newly proposed Binary PSO is based on the novel idea of replacing sigmoid Function(Logistic function) as used in PSO for generating binary numbers by a function named ***Gompertz Function a sigmoid function as well with additional parameters that could be further exploited to improve the performance of Binary PSO. Gompertz function [1] is widely applied in bioinformatics for modelling survival phenomenon of individuals with limited resources and restricted environmental conditions where birth rate initially increases and then slows down as resource limits are reached. The gompertz function is given by:

$$Gmpz(x) = a * \exp(b * \exp(cx)) \tag{4}$$

Where, a is the upper asymptote ; b, c are negative numbers where b sets the x displacement and c sets the growth rate (x scaling) or steepness of graph of Gompertz function; e is euler's Number(e = 2.71828...). The above mathematical relation typically represents a type of mathematical model for a time series, where growth is slowest at the start and end of a time period. In the present study the upper asymptote, a is set to be 1 so that values could be generated within the range [0,1]. The other parameters b and c are set to be change dynamically in a specified range calculated based on extensive calculations. The change in behaviour of gompertz function with the variation of parameters b, c and a= 1 could be demonstrated by Fig. 1, Fig. 2 and Fig. 3 as below:

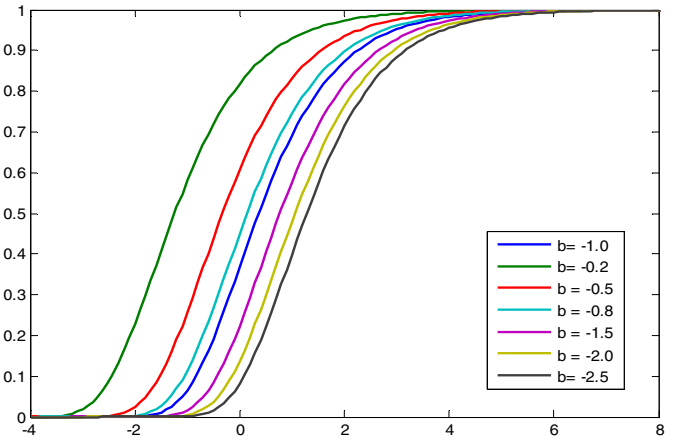


Fig. 1. Behaviour of gompertz function with a=1, c= -1 and for different values of parameter b

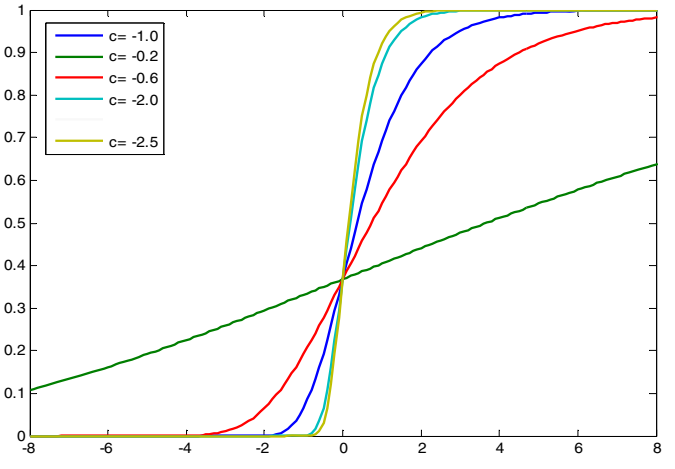


Fig. 2. Behaviour of gompertz function with a=1,b= -1 and for different values of parameter c

From the above discourse on change of behaviour of Gompertz function as the parameters b and c varies it is observed that the combined effect of changing b and c simultaneously could lead to significant amendments in binary PSO. If we change both the parameters simultaneously the gompertz function changes its shape from sigmoid function to a linear function that will further combine the

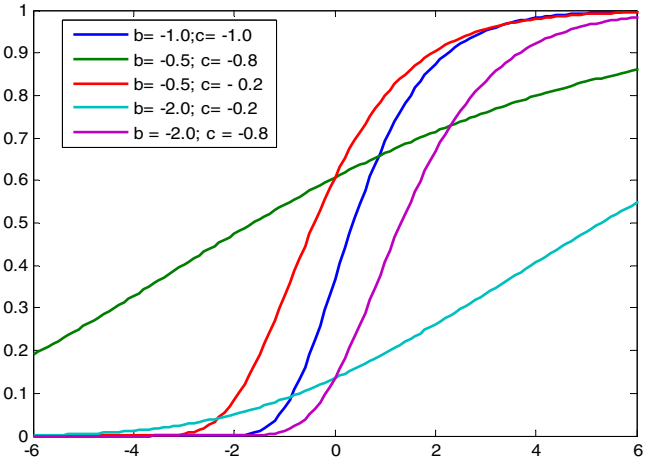


Fig. 3. Behaviour gompertz function with a=1 and varying values of parameter b and c

advantages of a linear function and a sigmoid function. The notion of how and why to change b and c is discussed in detail as below:

3.1 Variation of Parameter "b"

As in the Binary PSO the probability for changing a bit to 0 or 1, at the velocities near to zero is 0.5, here in Gompertz function, we could increase or decrease this probability in the range (0,1) by manipulating parameter b (as in Fig.1) appropriately in a range to be determined after extensive experiments and may also be problem dependent. We have observed that for, a= 1and *c*= -1, as the value of b varies it affects the probability of generating bits (0 or 1) in the following way: for b = -0.2, gmpz(v_{id} = 0) = 0.8214 and at b= -2.0, gmpz(v_{id} = 0) = 0.1379. from the above discourse, it is observed that by increasing b in the range [-0.2, -2.0], probability of generating number of 1's will be greater in initial stages and number of 0's in later stages. While decrement of b in the range [-2.0. -0.2] will increase probability of generating number of 0's in initial stages and 1's in later stages. Any of above settings for b could be used based on the problem at hand, the first setting is more beneficial in case of maximization, 0-1 knapsack type problems. While the second setting is more suitable for minimization type problems (Binary as well as continuous).

3.2 Variation of Parameter "c"

The parameter c decides the steepness of gompertz function. It could be observed from Fig.2 that as we increase c negatively the steepness of gompertz function increases and for lower values of c the GF is like a linear function while as we increase "c" it starts taking shape of alphabet "S". When function is linear it

provides better exploration so our idea is to change c as from to -0.2 to – 1.5 iteration wise, so that better exploration is provided in early iterations that also maintains diversity in population. The most noticeable fact is that from doing this we will be able to extract the benefit of linear function and sigmoid function as well within a single algorithm. The variation of parameters b and c with respect to iterations will help algorithm to produce a more diversified population as the gompertz function maintains a prolonged variation of probabilities within a greater range of maximum velocities. The pseudo code of gompertz based PSO is given below:

Algorithm: New Gompertz Based Binary PSO
BEGIN:
Create and Initialize a D-dimensional swarm, S
For t= 1 to the maximum number of iterations,
For i=1 to S,
For d=1 to D,
Apply the velocity update equation (1)
Update Position using equation as:
Calculate b and c iteration wise
Calculate Gmpz(V_{id}) using equation (5)
Calculate particle position as

$$x_{id}(t+1) = \begin{cases} 1 & if\ \ U(0,1) < Gmpz(V_{id}(t)) \\ 0 & otherwise \end{cases}$$

End- for-d;
Compute fitness of updated position;
If needed, update historical information for P_i and P_g;
End-for-i;
Terminate if P_g meets problem requirements;
End for t;
END

4 Experimental Evaluation

Benchmark Functions: The following benchmark problems are considered in the present study.

4.1 Parameter Setting

Swarm size(S) is taken as 5* number of variables. Acceleration coefficients c_1 and c_2 are taken as 2.8 and 1.3 respectively. Max Velocity ($V_{\max}$) is fixed as 4 and star topology is considered for particle's interaction. Each algorithm is run 50 times and the maximum number of iterations allowed is set to 500. All the algorithms are implemented in C++ and the experiments are performed on a DELL System with Intel(R) Xeon(R) CPU E5420 with 2.50 GHz speed and 16 GB of RAM under WINXP platform.

Table 1. List of Continuous benchmark functions

S.No	Function	Mathematical formula	Range & Dim	Minima
1.	**Sphere**	$f(x)=\sum_{i=1}^{n} x_i^2$	$[-5.12, 5.12]^{10}$	0
2.	**Griewank**	$f(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600, 600]^{10}$	0
3.	**Rosenbrock**	$f(x)=\sum_{i=1}^{n-1}\left(100*(x_{i+1}-x_i^2)^2+(x_i-1)^2\right)$	$[-30, 30]^{10}$	0
4.	**Rastrigin**	$f(x) = 10n+\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)\right)$	$[-5.12, 5.12]^{10}$	0
5.	**Ellipsoidal**	$f(x) = \sum_{i=1}^{n}(x_i-i)^2$	$[-30, 30]^{10}$	0

4.2 Performance Evaluation Criteria

For all benchmarks with known global optima, the termination criterion for all considered PSO variants is set to be as maximum number of iterations or when the known optimum is within 99% of accuracy, whichever occurs earlier while searching for a global optimal solution. A run in which the algorithm finds a solution satisfying $f_{\min} - f_{opt} \le 0.01$, where $f_{\min}$ is the best solution found when the algorithm terminates and f_{opt} is the known global minimum of the problem, is considered to be successful. For each method and problem the following are recorded:

- Success Rate (SR) = $\frac{Numberof\ successful\ runs}{Total\ numberof\ runs}\times 50$
- Average computational time (ACT) (seconds).
- Average number of function evaluations (AFE).
- Average Error (AE) = $\frac{\sum(f_{\min}-f_{opt})}{n}$ where, n is the total number of runs.
- Standard Deviation of Error (50) over 50 runs.
- Least Error (LE) over 50 runs.

5 Results and Discussion

The computational studies regarding algorithm verification are carried out in this section. All the considered PSO variants are tested against a set of five continuous benchmark. Table 2 summarizes the testing results for both the PSO variants in terms of various performance criteria given in the previous section. From this Table it can be easily observed that the proposed variant GBPSO outperforms the basic BPSO significantly in terms of various performance criteria and indicating the capability of proposed variants for providing a more accurate optimal solution.

Table 2. Results for continuous problems

Fun	Method	AE	LE	SD	SR	AFE	ACT
F1	BPSO	0	0	0	100	79660	4.26812
	GBPSO	0	0	0	100	68020	3.92092
F2	BPSO	0	0	0	100	123970	6.55374
	GBPSO	0	0	0	100	83090	5.47092
F3	BPSO	47.66	0	51.004161	55	166250	12.66718
	GBPSO	8.82	0	1.26	65	147430	8.74842
F4	BPSO	0	0	0	100	92370	5.14842
	GBPSO	0	0	0	100	82920	5.4403
F5	BPSO	0.8600	0	0.346987	14	247220	13.355
	GBPSO	0.02	0	0.140	98	162030	10.7153

6 Conclusion

This paper proposes a new variant of binary PSO called GBPSO which instead of basic sigmoid function makes use of Gompertz function for generating the binary numbers. GBPSO is tested on a suite of 5 benchmarks. Comparative analysis with basic BPSO proves the competence of proposed variant.

References

1. Gompertz, B.: On the Nature of the Function Expressive of the Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies. Phil. Trans. R. Soc. Lond. 115, 513–583 (1825), doi:10.1098/rstl.1825.0026
2. Kennedy, J., Eberhart, R.C.: A Discrete Binary Version of the Particle Swarm Algorithm. In: Proc. of IEEE 1997 Conference Systems Man Cybernetics, Piscataway, NJ, pp. 4104–4108 (1997)
3. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)
4. Luh, G.C., Lin, C.Y., Lin, Y.S.: A Binary Particle Swarm Optimization For Continuum structural topology Optimization. Applied Soft Computing 11, 2833–2844 (2011)
5. Sangwook, L., Park, H., Jeon, M.: Binary particle swarm optimization with bit change mutation. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E-90A(10), 2253–2256 (2007)
6. Sangwook, L., Sangmoon, S., Sanghoun, O., Pedrycz, W., Jeon, M.: Modified binary particle swarm optimization. Progress in Natural Science 18, 1161–1166 (2008)

Incorporating Genetic Algorithms in Transport Management

Kusum Deep[1], Om Prakash Dubey[2], and Atulya Nagar[3]

[1] Dept. of Mathematics, Indian Institute of Technology, Roorkee, India
kusumfma@iitr.ernet.in
[2] Dept. of Mathematics, Bengal College of Engg. & Tech., Durgapur, India
op_dubey2006@yahoo.co.in
[3] Head, Dept. of Computer Science, Liverpool Hope Univ., Hope Park, Liverpool, L169 JD, UK
nagara@hope.ac.uk

Abstract. The present paper deal with an unbalanced transportation problem, from coal transportation sector, is solved to demonstrate the application of Genetic Algorithm. Genetic Algorithms are unorthodox optimization search algorithms. It is a computerized search optimization algorithms based on the mechanics of natural genetics and natural selection. Using operations research techniques several feasible solutions are obtained. An efficient numerical solution has been found to suit the objective of the management by using Genetic Algorithm approach. It shows that an organization try to supply the commodities to all customers. This demonstrates the merit of Genetic Algorithm technique over the Transportation Problem, which is certainly an improvement in Transport Management.

Keywords: Genetic Algorithm, Transportation Problem, Demand level, Lexicographic Goal Programming.

1 Introduction

Transportation is directly related to industrialization, which is a symbol of developed country. In balanced/unbalanced Transportation Problem (T.P.), Decision Maker has to take a decision. Decision means to choose a better compromising course of action out of different available courses of action. The set of different actions can be obtained by successful application of OR-techniques.

One cannot think of industrialization without transportation which includes transportation of either public or goods/ores/raw materials etc. The priorities and demand may differ quantitatively and/or qualitatively. Apart from carrying persons from one place to other, the transportation problem is mainly concerned with the transportation of goods so that total transportation cost/time should be minimum. The minimization of transportation cost or transportation time leads to maximization of profit, under existing conditions.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 177–192.
springerlink.com

We are kind aware that India is facing acute power problem which is essential for development. Organizations are working in the field of production and distribution of thermal power whose production is based mainly upon coal. At the same time a huge amount of coal is required by steel manufacturing companies, bricks manufacturing companies, and many others. So demand for coal is very high but supply is less or limited due to various reasons. Thus in most of the cases we generally encounter with unbalanced transportation problems in which total supply are less than total demand.

The present paper deals with a sample problem taken from fourteen coal production sites (collieries) and six consumption sites (destinations) situated in Kuju / Charhi area of Central Coalfield Limited (A unit of Coal India Limited) in Hazaribag district of Jharkhand State. Presently the available methods for Transportation Problem did not provide to supply of commodity from every source/production site to every customers/consumption site. The purpose of this paper is to present a model so that the companies must have a good number of customers and should try to supply the commodities to all customers also companies running in loss may run in profit by making them a professional profitable public sector. This is possible by Genetic Algorithm Technique, a powerful technique to solve multi-objective decision-making problems. Genetic Algorithms perform directed random search through a given set of alternatives with the aim of finding the best alternative with respect to the given criteria of goodness. However the performance of any model will be depending on the availability of fund and local conditions.

2 Genetic Algorithm

Genetic Algorithm (GA), which focuses on optimizing general combinatorial problems; is a search technique used in computing to find true or approximate solutions to Optimization problems. GAs are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as (biological) inheritance, mutation, selection and crossover (recombination). GA is categorized as global optimization as well as global search heuristics. Here a candidate solution is represented by strings over a finite alphabet. Genetic Algorithm's are well suited for optimizing combinatorial problems [1, 2].

The development of a GA in the solution of a particular optimization problem involves two types of decisions. The first concerns the way in which the problem is to be modeled to fit into the GA framework and includes the definition of the space of feasible solutions, the form of the fitness function and the way in which individuals are to be represented. The second concerns the parameters of the GA itself and includes the proportion of the population to be produced as a result of reproduction, crossover and mutation; the selection procedure; the population size; the number of generations and other related decisions.

The basic structure and operation of a GA is as follows:-

GAs operates on a population of structures which are fixed length strings representing all possible solutions of a problem domain. Using such a representation, an initial population is randomly generated. For each structure (trial solution)

in the population, a fitness value is assigned. Each structure is then assigned a probability measure based on the fitness value which decides the contribution a parent solution makes to the new generation. This phase is referred to as the Reproduction Phase. Each of the offspring generated by the reproduction phase is then modified using Genetic Operators. The two operators used here are the Crossover Operator and the Mutation Operator. In the Crossover Operation, two individual strings are selected randomly for the population. A Crossover point is randomly selected to lie between the defining lengths of the string. The resulting substrings of the two parent strings are swapped resulting in two new strings. The Mutation Operator generates a new string by independently modifying the values at each loci of an existing string with a probability p_m. The parameter p_m is referred to as the probability of mutation.

3 Transportation Problem

General Transportation Problem (T.P.) with m Factory/production sites F_i and n destination D_j is given by the cost matrix $[C_{ij}]$, i = 1, 2, . . ., m, and j = 1, 2, . . ., n, together with production capacity (supply) a_i at F_i and demand b_j at D_j [3, 4]. The problem is said to be balanced if $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ (i.e. total supply = total demand) otherwise unbalanced.

Let the problem under consideration be unbalanced so that total demand exceeds total supply. The problem can be expressed mathematically as,

$$\text{Minimize } Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

Subject to,

$$\sum_{j=1}^{n} x_{ij} = a_i, \qquad i = 1, 2, \ldots, m.$$

$$\sum_{i=1}^{m} x_{ij} \geq b_j, \quad j = 1, 2, \ldots, n$$

$$\sum_{i=1}^{m} a_i \leq \sum_{j=1}^{n} b_j, \qquad x_{ij} \geq 0. \qquad \text{... (i)}$$

Where, x_{ij} is the amount of goods to be transported from source (production site) F_i to the destination (consumption site) D_j.

4 Formulation of Transportation Problem as Goal Programming Problem

Goal Programming (GP) is a highly flexible and practical methodology for modeling, solving and analyzing problems with multiple conflicting objectives. The key

aim of GP is to minimize deviations from the goal values. Fixing the relevant goal values is one of the most important steps of developing a GP model without which the formulation of the problem may lead to a wrong solution. Alternate acceptable solutions also exist. At different levels Decision Maker (DM) can always change the priority to suit the situations.

The GP formulations ordered the unwanted deviations into a number of priority levels, with the minimization of a deviation in a higher priority level being of infinitely more importance than any deviations in lower priority levels [5, 6]. This is known as Lexicographic or Pre–emptive GP. Ignizio [7, 8] gives an algorithm showing how a Lexicographic GP can be solved as a series of linear programmes. Lexicographic GP should be used when there exists a clear ordering amongst the decisions. A prioritized version of GP can be expressed as,

$$\text{Minimize } F(d)= P_1\Big(\sum_{r=k+1}^{k+m} d_r^{+}+\sum_{r=k+1}^{k+m} d_r^{-}\Big)+\sum_{r=2}^{k+1} P_r(d_{r-1}^{+}+d_{r-1}^{-})$$

Subject to,

$$f_r(x) + d_r^{-} - d_r^{+} = P_r, \qquad r = 1, 2, \ldots, k.$$

$$g_i(x) + d_i^{-} - d_i^{+} = P_i, \qquad i = k+1, \ldots, k+m.\ d_r^{+}, \quad d_r^{-}, d_i^{+}, d_i^{-}, x_j \geq 0,$$

$$j = 1, 2, \ldots, n.$$

with the conditions

$$d_r^{+}.\, d_r^{-} = 0 = d_i^{+}.\, d_i^{-}, \quad \forall \text{ r and i.} \qquad \ldots \text{(ii)}$$

$f_r(x)$ and $g_i(x)$ are linear functions of decision variables, the model (ii) can be solved by modified simplex method (Lee [9] and Ignizio [10]). The objective in model (ii) can be taken in a number of ways by changing the priority levels of goals.

Using model (ii), the goal programming version of model (i) can be written as,

$$\text{Minimize } F(d) = \sum_{j=1}^{n} P_j(d_j^{-})+d_{n+1}^{+}$$

Subject to,
Supply constraints,

$$\sum_{j=1}^{n} x_{ij} + d_j^{-} - d_j^{+} = a_i, \qquad i = 1, 2, \ldots, m;\ j = 1,2, \ldots, k.$$

Demand goals,

$$\sum_{i=1}^{m} x_{ij} + d_j^{-} - d_j^{+} = b_j^{*}, \qquad j = k+1, k+2, \ldots, k+n.$$

and, Budget goal,

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij} + d_{k+n+1}^{-} - d_{k+n+1}^{+} = B, \qquad \text{... (iii)}$$

where, $d_j^- \cdot d_j^+ = 0$, $\forall$ j; x_{ij}, d_j^-, $d_j^+ \geq 0$; and d_j^-, d_j^+, (j = 1, 2, . . ., n, n+1) are deviational variables, b_j^* represents certain percentage of b_j so that $b_j^* \leq b_j$ $\forall$ j = 1, 2, . . . , n, and B represents budget aspiration level as fixed by the DM.

5 Methodology

Developed tools of GA are used to obtain the optimize solution of Multi-objective Optimization Problem from Management and Industrial Engineering for Decision Maker. In the proposed technique, known as MI-LX-PM, the issue of exploration and exploitation makes a recombination and mutation operator dependent on the chosen selection operator for successful GA run [10, 11, 12].

MI-LX-PM (Laplace Crossover – LX and Power Mutation – PM are extended for integer decision variables) is efficient to solve integer and mixed integer constrained optimization problems. Here, a special truncation procedure for satisfaction of integer restriction on decision variables and a 'parameter free' penalty approach for constraint handling are used. The proposed steps and techniques are as follows:

Step 1st Real Coded Genetic Algorithm

The way the variables are coded is clearly essential for GAs efficiency. Here real – encoding of chromosomes is used in which decision variables are encoded as real numbers. GAs which make use of the real-encoding of chromosomes are termed as Real Coded Genetic Algorithms (RCGA). This representation seems quite natural to deal with problems having continuous search space. The use of real-parameter makes it possible to use large domains for variables.

Step 2nd Tournament Selection Technique

Genetic Algorithms use a selection technique to select individuals from the population to insert individual into mating pool. Individuals from the mating pool are used to generate new offspring, with the resulting offspring forming the basis of the next generation.

Goldberg and Deb ([13]) have shown that the Tournament Selection has better or equivalent convergence and computational time complexity properties when compared to any other reproduction operator that exists in the literature. So, in this algorithm, Tournament Selection operator is used as Reproduction Operator. In the Tournament Selection, tournaments are played between *k solutions* (*k* is tournament size) and the better solution is chosen and placed in the mating pool. *k* other solutions are picked again and another slot in the mating pool is filled with the better solution. If carried out systematically, each solution can be made to participate in exactly *k* tournaments. The best solution in a population will win all the *k-tournaments*, thereby making *k* copies of it in the new population. Using a

similar argument the worst solution will lose in all the *k-tournaments* and will be eliminated from the population. The user specifies the size of the tournament set as a percentage of the total population. Here, Tournament Selection Operator with tournament size three is used.

Step 3rd Laplace Crossover

In RCGAs, crossover has always been considered to be the fundamental search operator. Deep and Thakur ([14]) introduce a new parent centric real coded crossover operator, based on Laplace Distribution, known as Laplace Crossover (LX) to improve the performances of Real Coded Genetic Algorithms for function optimization. Using Laplace Crossover, two off springs $y^{(1)} = (y_1^{(1)}, y_2^{(1)}, \ldots, y_n^{(1)})$ and $y^{(2)} = (y_1^{(2)}, y_2^{(2)}, \ldots, y_n^{(2)})$ are generated from a pair of parents $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$ and $x^{(2)} = (x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)})$ in the following way.

First, two uniformly distributed random numbers, $u^i, u_i' \in [0,1]$ are generated. Then, a random number β_i is generated which follows the Laplace distribution by simply inverting the distribution function of Laplace distribution as follows:

$$\beta_i = \begin{cases} a - b\ \log_e(u_i),\ u_i' \le \frac{1}{2} \\ a + b\ \log_e(u_i),\ u_i' > \frac{1}{2} \end{cases}.$$

The offsprings are given by the equation,

$$y_i^{(1)} = x_i^{(1)} + \beta_i \left| x_i^{(1)} - x_i^{(2)} \right|,$$
$$y_i^{(2)} = x_i^{(2)} + \beta_i \left| x_i^{(1)} - x_i^{(2)} \right|.$$

This way the Laplace Crossover operator exhibits self adaptive behavior. If Laplace Crossover produces an offsprings which violates a box-constraint, i.e. $x_i < x_i^l$ or $x_i > x_i^u$ for some *i*, then x_i is assigned a random value in the interval $[x_i^l, x_i^u]$.

Step 4th Power Mutation

A mutation operator known as Power Mutation (PM), which is based on power distribution, is introduced for RCGA s ([15]). Its distribution function is given by, $f(x) = px^{p-1},\ 0 \le x \le 1$, and the density function is given by $F(x) = x^p,\ 0 \le x \le 1$, where *p* is the index of the distribution.

The PM is used to create a solution *y* in the vicinity of a parent solution $\bar{x}$ in the following manner. First a uniform random number *t* between 0 and 1 is created

and a random number *s* is created which follows the above mentioned distribution. Then following formula is used to create the muted solution,

$$y = \begin{cases} \bar{x} - s(\bar{x} - x^l), & if \ \ t < r, \\ \bar{x} + s(x^u - \bar{x}), & if \ t \geq r, \end{cases} \text{ where } t = \frac{x - x^l}{x^u - x^l} \text{ and } x^l \text{and } x^u$$

are lower and upper bounds of the decision variable and *r* is a uniformly distributed random number between 0 and 1. The strength of mutation is governed by the index of the mutation *(p)*. For small values of *p* less perturbance in the solution is expected and for large values of *p* more diversity is achieved. The probability of producing a mutated solution *y* on left (right) side of $\bar{x}$ is proportional to distance of $\bar{x}$ from x^l (x^u) and the muted solution is always feasible.

Algorithm of Genetic Algorithm:-

1) Input
   a. A population of a set of solutions (P),
   b. Population size (m),
   c. Number of iterations (N).
2) Output: Best possible solution (SB), and Cost of SB.
3) Repeat.
4) Sort solutions in P according to their costs.
5) Reproduction (P, s_1, s_2) selects two solutions s_1 and s_2 from P randomly. For exponential chance of selecting fittest solution generate a random number n_1 between [0, n-1] and then generate another random number n_2 between [0, n_1].
6) Perform S = Crossover (s_1, s_2). For this select randomly a number n* between [2, n-1]. Construct a solution s_1* by taking first n* blocks of s_1 and last (n-n*) blocks of s_2. Construct one another solution s_2* by taking first n* blocks of s_2 and the last (n-n*) blocks of s_1.
7) Find cost of two solutions.
8) Take the best from these two solutions from Crossover in P
9) Perform Mutation (e.g. after every five iteration). This is done by allocating an unallocated job for each solution in P.
10) Until (P converges or N iterations performed).
11) Set SB as first solution of P.

End Algorithm of Genetic Algorithm.

6 Sample Problem

Fourteen coal production sites with six user sites situated in C.C.L. Area (Unit of Coal India Ltd.) in Hazaribag of Jharkhand has been considered for our sample problem [16]. Let, X_{ij} = Amount of coal (lac metric ton) required to transport from i^{th} production site to j^{th} consumption site, i = 1, 2,..., 14 and j = 1, 2, ...,6. As per the data collected approximately, the information can be tabulated as Table 1:

Table 1.

Destination / Source / Source-point	01. KW	02. Giddi W	03. Rajrappa	04. NRS	05. TTPS	06. PTPS	Supply
01. Kuju	088	128	180	056	128	200	10
02. Ghato	032	220	280	128	208	300	05
03. Tapin N	048	272	300	180	280	340	07
04. Tapin S	064	172	272	152	240	312	08
05. KOGP	032	280	320	220	300	340	05
06. KOCP	032	320	320	208	288	360	09
07. Jh. New	040	300	340	208	312	368	06
08. Parej	024	260	348	208	388	340	07
09. Sayal	220	220	300	272	320	180	08
10. Pundi	088	192	200	080	192	288	07
11. Saunda D	220	216	268	208	300	060	10
12. Urimari	260	060	208	180	240	128	09
13. Giddi A	309	002	240	180	260	120	08
14. Sounda	240	040	260	192	280	064	06
Demand	15	15	12	25	20	20	**105** / **107**

Here we find that total supply = Σa_i = 105 lac metric ton and total demand = Σb_j = 107 lac metric ton, that is supply is less than demand. The problem is thus unbalanced.

Solution obtained by different methods (using software) for Transportation Problems e.g. North–West corner method, Least-Cost method, Vogel's method are summarized as follows:

North–West corner method:-

a. Value of variables x_{ij}: $x_{1,5} = 10$, $x_{2,5} = 5$, $x_{3,4} = 7$, $x_{4,4} = 5$, $x_{4,5} = 3$, $x_{5,1}=1$, $x_{5,3} = 4$, $x_{6,1}=7$, $x_{6,5} = 2$, $x_{7,4} = 6$, $x_{8,1} = 7$, $x_{9,3} = 4$, $x_{9,6} = 4$, $x_{10,4} = 7$, $x_{11,6}= 10$, $x_{12,2}= 7$, $x_{12,3}= 2$ $x_{13,2}= 8$, $x_{14,6}= 6$; {Dummy variable, $x_{15,3}= 2$}.

b. Objective Value: Rs.12, 904.00.

Least-Cost method:-

a) Value of variables x_{ij}: $x_{1,5} = 10$, $x_{2,5}=5$, $x_{3,4}=7$, $x_{4,4}=8$, $x_{5,1}= 5$, $x_{6,3} = 4$, $x_{6,5} = 5$, $x_{7,1} = 3$, $x_{7,4} = 3$, $x_{8,1} = 7$, $x_{9,3} = 4$, $x_{9,6} = 4$, $x_{10,4} = 7$, $x_{11,6} = 10$, $x_{12,2} = 7$, $x_{12,3} = 2$ $x_{13,2} = 8$, $x_{14,6} = 6$; {Dummy variable, $x_{15,3} = 2$}.

b) Objective Value: Rs.12, 904.00.

Vogel's method:-

(a) Value of variables x_{ij}: $x_{1,5} = 10$, $x_{2,5} = 5$, $x_{3,4}=7$, $x_{4,3}= 3$,$x_{4,4} = 5$, $x_{5,1} = 4$, $x_{5,3} =1$, $x_{6,1} =4$, $x_{6,5} =5$, $x_{7,4} =6$, $x_{7,4} =3$, $x_{8,1} =7$, $x_{9,3} =4$, $x_{9,6} =4$, $x_{10,4} =7$, $x_{11,6} =10$, $x_{12,2}=7$, $x_{12,3}=2$, $x_{13,2}=8$, $x_{14,6}=6$; {Dummy variable, $x_{15,3} = 2$}.

(b) Objective Value: Rs.12, 904.00.

Mathematical formulation of T.P. and solution obtained by using GA Technique:-

Mathematical formulation of above problem may be written as,

$$\begin{aligned}\text{Min } Z = {} & 088x_{1,1} + 128x_{1,2} + 180x_{1,3} + 056x_{1,4} + 128x_{1,5} + 200x_{1,6} + 032x_{2,1} + \\ & 220x_{2,2} + 280x_{2,3} + 128x_{2,4} + 208x_{2,5} + 300x_{2,6} + 048x_{3,1} + 272x_{3,2} + \\ & 300x_{3,3} + 180x_{3,4} + 280x_{3,5} + 340x_{3,6} + 064x_{4,1} + 172x_{4,2} + 272x_{4,3} + \\ & 152x_{4,4} + 240x_{4,5} + 312x_{4,6} + 032x_{5,1} + 280x_{5,2} + 320x_{5,3} + 220x_{5,4} + \\ & 300x_{5,5} + 340x_{5,6} + 032x_{6,1} + 320x_{6,2} + 320x_{6,3} + 208x_{6,4} + 288x_{6,5} + \\ & 360x_{6,6} + 040x_{7,1} + 300x_{7,2} + 340x_{7,3} + 208x_{7,4} + 312x_{7,5} + 368x_{7,6} + \\ & 024x_{8,1} + 260x_{8,2} + 348x_{8,3} + 208x_{8,4} + 388x_{8,5} + 340x_{8,6} + 220x_{9,1} + \\ & 220x_{9,2} + 300x_{9,3} + 272x_{9,4} + 320x_{9,5} + 180x_{9,6} + 088x_{10,1} + 192x_{10,2} + \\ & 200x_{10,3} + 080x_{10,4} + 192x_{10,5} + 288x_{10,6} + 220x_{11,1} + 216x_{11,2} + 268x_{11,3} + \\ & 208x_{11,4} + 300x_{11,5} + 060x_{11,6} + 260x_{12,1} + 060x_{12,2} + 208x_{12,3} + 180x_{12,4} + \\ & 240x_{12,5} + 128x_{12,6} + 300x_{13,1} + 002x_{13,2} + 240x_{13,3} + 180x_{13,4} + 260x_{13,5} + \\ & 120x_{13,6} + 240x_{14,1} + 040x_{14,2} + 260x_{14,3} + 192x_{14,4} + 280x_{14,5} + 064x_{14,6}.\end{aligned}$$

Subject to,

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} \leq 10$$
$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} \leq 05$$
$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} \leq 07$$
$$x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,6} \leq 08$$
$$x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,6} \leq 05$$
$$x_{6,1} + x_{6,2} + x_{6,3} + x_{6,4} + x_{6,5} + x_{6,6} \leq 09$$
$$x_{7,1} + x_{7,2} + x_{7,3} + x_{7,4} + x_{7,5} + x_{7,6} \leq 06$$
$$x_{8,1} + x_{8,2} + x_{8,3} + x_{8,4} + x_{8,5} + x_{8,6} \leq 07$$
$$x_{9,1} + x_{9,2} + x_{9,3} + x_{9,4} + x_{9,5} + x_{9,6} \leq 08$$

$$x_{10,1} + x_{10,2} + x_{10,3} + x_{10,4} + x_{10,5} + x_{10,6} \leq 07$$
$$x_{11,1} + x_{11,2} + x_{11,3} + x_{11,4} + x_{11,5} + x_{11,6} \leq 10$$
$$x_{12,1} + x_{12,2} + x_{12,3} + x_{12,4} + x_{12,5} + x_{12,6} \leq 09$$
$$x_{13,1} + x_{13,2} + x_{13,3} + x_{13,4} + x_{13,5} + x_{13,6} \leq 08$$
$$x_{14,1} + x_{14,2} + x_{14,3} + x_{14,4} + x_{14,5} + x_{14,6} \leq 06$$
$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} + x_{8,1} + x_{9,1} + x_{10,1} + x_{11,1} + x_{12,1} + x_{13,1} + x_{14,1} \geq 15$$
$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} + x_{6,2} + x_{7,2} + x_{8,2} + x_{9,2} + x_{10,2} + x_{11,2} + x_{12,2} + x_{13,2} + x_{14,2} \geq 15$$
$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} + x_{6,3} + x_{7,3} + x_{8,3} + x_{9,3} + x_{10,3} + x_{11,3} + x_{12,3} + x_{13,3} + x_{14,3} \geq 12$$
$$x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} + x_{6,4} + x_{7,4} + x_{8,4} + x_{9,4} + x_{10,4} + x_{11,4} + x_{12,4} + x_{13,4} + x_{14,4} \geq 25$$
$$x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} + x_{6,5} + x_{7,5} + x_{8,5} + x_{9,5} + x_{10,5} + x_{11,5} + x_{12,5} + x_{13,5} + x_{14,5} \geq 20$$
$$x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} + x_{6,6} + x_{7,6} + x_{8,6} + x_{9,6} + x_{10,6} + x_{11,6} + x_{12,6} + x_{13,6} + x_{14,6} \geq 20$$

where, $x_{ij} \geq 0$, i = 1, 2, ..., 14; j = 1, 2, ..., 6. ... (iv)

Solution obtained by using Genetic Algorithm Technique-

a) Objective Value: Rs.18,618.909981667013. ≈ Rs.18,618.91.
b) Crossover Probability: 0.8000.
c) Mutation Probability: 0.0001.
d) Value of variables x_{ij}:

$x_{1,1}$=1.41869556, $x_{1,2}$=2.79938128, $x_{1,3}$=1.13873383, $x_{1,4}$=0.29284409, $x_{1,5}$=1.71501670, $x_{1,6}$=0.36057845, $x_{2,1}$=0.65297580, $x_{2,2}$=0.04127890, $x_{2,3}$=0.16830337, $x_{2,4}$=1.74936827, $x_{2,5}$=0.53944459, $x_{2,6}$=1.56243490, $x_{3,1}$=1.86437378, $x_{3,2}$=0.76069579, $x_{3,3}$=0.50853819, $x_{3,4}$=0.49110544, $x_{3,5}$=0.79155924, $x_{3,6}$=2.12695992, $x_{4,1}$=0.88075749, $x_{4,2}$=1.30623048, $x_{4,3}$=1.38844905, $x_{4,4}$=1.60940541, $x_{4,5}$=0.80115520, $x_{4,6}$=0.52960272, $x_{5,1}$=0.63596981, $x_{5,2}$=0.54760036, $x_{5,3}$=1.01021899, $x_{5,4}$=1.05992640, $x_{5,5}$=0.98395596, $x_{5,6}$=0.60837914, $x_{6,1}$=0.55077030, $x_{6,2}$=0.80658721, $x_{6,3}$=1.23412565, $x_{6,4}$=0.87411499, $x_{6,5}$=3.18037858, $x_{6,6}$=1.14503750, $x_{7,1}$=0.86110934, $x_{7,2}$=0.83897026, $x_{7,3}$=0.29825203, $x_{7,4}$=1.44305360, $x_{7,5}$=1.50034035, $x_{7,6}$=0.94028400, $x_{8,1}$=0.74105452, $x_{8,2}$=1.66950575, $x_{8,3}$=0.69507665, $x_{8,4}$=0.46996134, $x_{8,5}$=0.63062440, $x_{8,6}$=1.60154761, $x_{9,1}$=0.56770727, $x_{9,2}$=3.61812218, $x_{9,3}$=1.02692210, $x_{9,4}$=0.34264968, $x_{9,5}$=0.62657657, $x_{9,6}$=0.89068298, $x_{10,1}$=2.12830442, $x_{10,2}$=0.30014077, $x_{10,3}$=1.01699623, $x_{10,4}$=1.22829430, $x_{10,5}$=1.04816440, $x_{10,6}$=0.68735710, $x_{11,1}$=1.62318497, $x_{11,2}$=0.98722849, $x_{11,3}$=1.38888093, $x_{11,4}$=0.75560671, $x_{11,5}$=1.28758097, $x_{11,6}$=3.68513199, $x_{12,1}$=1.72430851, $x_{12,2}$=0.30087553, $x_{12,3}$=1.58016175, $x_{12,4}$=0.63180436, $x_{12,5}$=2.00487897, $x_{12,6}$=0.67077166, $x_{13,1}$=0.68765705, $x_{13,2}$=2.26123113, $x_{13,3}$=0.69052304, $x_{13,4}$=1.93433436, $x_{13,5}$=0.40767783, $x_{13,6}$=0.88888020, $x_{14,1}$=0.66317503, $x_{14,2}$=0.45467059, $x_{14,3}$=1.08529863, $x_{14,4}$=0.40475719, $x_{14,5}$=2.09144415, $x_{14,6}$=0.38914215.

The graph of maximum fitness vs. number of generation shows the favorable quality of the results as effect of number of generation on the optimal solution-

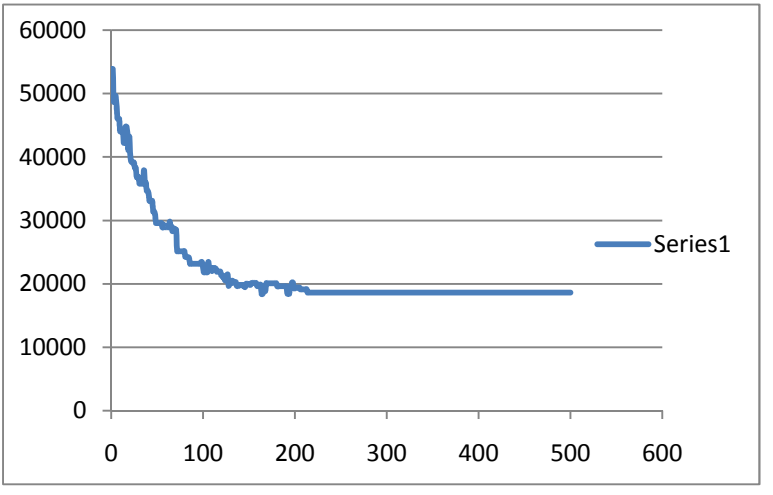


Fig. 1. Maximum fitness value vs. Number of Generation

Mathematical formulation as G.P. of T.P.:-

The GP version of the above model (iv) may be obtained as,

Minimize F(d)

Subject to,

supply goal,

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} \leq 10$$
$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} \leq 05$$
$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} \leq 07$$
$$x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} + x_{4,5} + x_{4,6} \leq 08$$
$$x_{5,1} + x_{5,2} + x_{5,3} + x_{5,4} + x_{5,5} + x_{5,6} \leq 05$$
$$x_{6,1} + x_{6,2} + x_{6,3} + x_{6,4} + x_{6,5} + x_{6,6} \leq 09$$
$$x_{7,1} + x_{7,2} + x_{7,3} + x_{7,4} + x_{7,5} + x_{7,6} \leq 06$$
$$x_{8,1} + x_{8,2} + x_{8,3} + x_{8,4} + x_{8,5} + x_{8,6} \leq 07$$
$$x_{9,1} + x_{9,2} + x_{9,3} + x_{9,4} + x_{9,5} + x_{9,6} \leq 08$$
$$x_{10,1} + x_{10,2} + x_{10,3} + x_{10,4} + x_{10,5} + x_{10,6} \leq 07$$
$$x_{11,1} + x_{11,2} + x_{11,3} + x_{11,4} + x_{11,5} + x_{11,6} \leq 10$$
$$x_{12,1} + x_{12,2} + x_{12,3} + x_{12,4} + x_{12,5} + x_{12,6} \leq 09$$
$$x_{13,1} + x_{13,2} + x_{13,3} + x_{13,4} + x_{13,5} + x_{13,6} \leq 08$$
$$x_{14,1} + x_{14,2} + x_{14,3} + x_{14,4} + x_{14,5} + x_{14,6} \leq 06$$

demand goal,

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} + x_{8,1} + x_{9,1} + x_{10,1} + x_{11,1} + x_{12,1} + x_{13,1} + x_{14,1} \geq 15$$

$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} + x_{6,2} + x_{7,2} + x_{8,2} + x_{9,2} + x_{10,2} + x_{11,2} + x_{12,2} + x_{13,2} + x_{14,2} \geq 15$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} + x_{6,3} + x_{7,3} + x_{8,3} + x_{9,3} + x_{10,3} + x_{11,3} + x_{12,3} + x_{13,3} + x_{14,3} \geq 12$$

$$x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} + x_{6,4} + x_{7,4} + x_{8,4} + x_{9,4} + x_{10,4} + x_{11,4} + x_{12,4} + x_{13,4} + x_{14,4} \geq 25$$

$$x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} + x_{6,5} + x_{7,5} + x_{8,5} + x_{9,5} + x_{10,5} + x_{11,5} + x_{12,5} + x_{13,5} + x_{14,5} \geq 20$$

$$x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} + x_{6,6} + x_{7,6} + x_{8,6} + x_{9,6} + x_{10,6} + x_{11,6} + x_{12,6} + x_{13,6} + x_{14,6} \geq 20$$

budget goal,

$$088x_{1,1} + 128x_{1,2} + 180x_{1,3} + 056x_{1,4} + 128x_{1,5} + 200x_{1,6} + 032x_{2,1} + 220x_{2,2} + 280x_{2,3} + 128x_{2,4} + 208x_{2,5} + 300x_{2,6} + 048x_{3,1} + 272x_{3,2} + 300x_{3,3} + 180x_{3,4} + 280x_{3,5} + 340x_{3,6} + 064x_{4,1} + 172x_{4,2} + 272x_{4,3} + 152x_{4,4} + 240x_{4,5} + 312x_{4,6} + 032x_{5,1} + 280x_{5,2} + 320x_{5,3} + 220x_{5,4} + 300x_{5,5} + 340x_{5,6} + 032x_{6,1} + 320x_{6,2} + 320x_{6,3} + 208x_{6,4} + 288x_{6,5} + 360x_{6,6} + 040x_{7,1} + 300x_{7,2} + 340x_{7,3} + 208x_{7,4} + 312x_{7,5} + 368x_{7,6} + 024x_{8,1} + 260x_{8,2} + 348x_{8,3} + 208x_{8,4} + 388x_{8,5} + 340x_{8,6} + 220x_{9,1} + 220x_{9,2} + 300x_{9,3} + 272x_{9,4} + 320x_{9,5} + 180x_{9,6} + 088x_{10,1} + 192x_{10,2} + 200x_{10,3} + 080x_{10,4} + 192x_{10,5} + 288x_{10,6} + 220x_{11,1} + 216x_{11,2} + 268x_{11,3} + 208x_{11,4} + 300x_{11,5} + 060x_{11,6} + 260x_{12,1} + 060x_{12,2} + 208x_{12,3} + 180x_{12,4} + 240x_{12,5} + 128x_{12,6} + 300x_{13,1} + 002x_{13,2} + 240x_{13,3} + 180x_{13,4} + 260x_{13,5} + 120x_{13,6} + 240x_{14,1} + 040x_{14,2} + 260x_{14,3} + 192x_{14,4} + 280x_{14,5} + 064x_{14,6} \leq 100.$$

absolute objective,

$$x_{ij} \geq 0,\ i = 1, 2, ..., 14;\ j = 1, 2, ..., 6. \qquad \text{... (v)}$$

Where F(d) is a function of prioritized goal deviations.

Here, an aspiration level of 100 lac rupees has been assigned to the budget goal. A number of alternate solutions may be obtained by changing priorities of goals.

Other negotiable solutions in different conditions for the model (v) can be obtained by reducing the demands by 75% and/or 50% that is with demand goals.

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} + x_{5,1} + x_{6,1} + x_{7,1} + x_{8,1} + x_{9,1} + x_{10,1} + x_{11,1} + x_{12,1} + x_{13,1} + x_{14,1} \geq 11.25 \text{ or } 7.5.$$

$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} + x_{5,2} + x_{6,2} + x_{7,2} + x_{8,2} + x_{9,2} + x_{10,2} + x_{11,2} + x_{12,2} + x_{13,2} + x_{14,2} \geq 11.25 \text{ or } 7.5.$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} + x_{6,3} + x_{7,3} + x_{8,3} + x_{9,3} + x_{10,3} + x_{11,3} + x_{12,3} + x_{13,3} + x_{14,3} \geq 09 \text{ or } 06.$$

$$x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} + x_{5,4} + x_{6,4} + x_{7,4} + x_{8,4} + x_{9,4} + x_{10,4} + x_{11,4} + x_{12,4} + x_{13,4} + x_{14,4} \geq 18.75 \text{ or } 12.5.$$

$$x_{1,5} + x_{2,5} + x_{3,5} + x_{4,5} + x_{5,5} + x_{6,5} + x_{7,5} + x_{8,5} + x_{9,5} + x_{10,5} + x_{11,5} + x_{12,5} + x_{13,5} + x_{14,5} \geq 15 \text{ or } 10.$$

$$x_{1,6} + x_{2,6} + x_{3,6} + x_{4,6} + x_{5,6} + x_{6,6} + x_{7,6} + x_{8,6} + x_{9,6} + x_{10,6} + x_{11,6} + x_{12,6} + x_{13,6} + x_{14,6} \geq 15 \text{ or } 10 \qquad \text{... (vi)}$$

Taking demand levels at 100% (total demand = 107.00), 75% (total demand = 80.25), 50% (total demand = 53.50) and different priority levels of all supply demand and budget goals, altogether six different situations occurred in each case in which a number of alternate solutions were found, which satisfy absolute objectives x_{ij}. Let S, B, D stand for supply, demand and budget goals respectively. The solutions obtained by solving (v) and (vi) can be tabulated as follows:

Table 2. Solution Table

Sr No	Demand Level	Priority Order	N0. of alternat solution	Best solution obtained	Total demand fulfilled out of 107/80.25/53.5 units
1	100%	S D B	11	$X_{1,5}$=10.000, $X_{2,5}$=5.000, $X_{3,4}$=7.000, $X_{4,4}$=5.000, $X_{4,5}$=3.000, $X_{5,1}$=1.000, $X_{5,3}$=4.000, $X_{6,1}$=7.000, $X_{6,5}$=2.000, $X_{7,4}$=6.000, $X_{8,1}$=7.000, $X_{9,3}$=4.000, $X_{9,6}$=4.000, $X_{10,4}$=7.000, $X_{11,6}$=10.000, $X_{12,2}$=7.000, $X_{12,3}$=2.000, $X_{13,2}$=8.000, $X_{14,6}$ = 6.000.	105.0000
		S B D	01	$X_{8,1}$ = 3.500, $X_{13,2}$ = 8.000.	11.5000
		D S B	11	$X_{1,5}$=10.000, $X_{2,5}$=5.000, $X_{3,4}$=7.000, $X_{4,4}$=5.000, $X_{4,5}$=3.000, $X_{5,1}$=1.000, $X_{5,3}$=4.000, $X_{6,1}$=7.000, $X_{6,5}$=2.000, $X_{7,4}$=6.000, $X_{8,1}$=7.000, $X_{9,3}$=4.000, $X_{9,6}$=4.000, $X_{10,4}$=7.000, $X_{11,6}$=10.000, $X_{12,2}$=5.000, $X_{12,3}$=4.000, $X_{13,2}$=10.000, $X_{14,6}$ =6.000.	**107.0000 (Acceptable) BudgetRs.13,204.00**
		D B S	01	$X_{1,3}$=12.000, $X_{1,4}$=25.000, $X_{1,5}$=20.000, $X_{8,1}$=15.000, $X_{11,6}$=20.000, $X_{13,2}$=15.000.	**107.0000 (Acceptable) Budget-Rs. 6,710.00**
		B S D	01	$X_{8,1}$=3.500, $X_{13,2}$ = 8.000.	11.5000
		B D S	01	$X_{8,1}$ = 2.9167, $X_{13,2}$ = 15.000.	17.9167
2	75%	B D S	01	$X_{8,1}$ = 3.2292, $X_{13,2}$ = 11.2500.	14.4792
		B S D	01	$X_{8,1}$ = 3.500, $X_{13,2}$ = 8.000.	11.5000
		D B S	01	$X_{1,3}$=9.000, $X_{1,4}$=18.750, $X_{1,5}$=15.000, $X_{8,1}$=11.250, $X_{11,6}$=15.000, $X_{13,2}$=11.250.	**80.2500 (Acceptable) Budget-Rs. 5,782.50**

Table 2. (*continued*)

		D S B	09	$X_{1,5}$=10.000, $X_{2,5}$=5.000, $X_{3,4}$=3.750, $X_{4,4}$=8.000, $X_{5,1}$=4.250, $X_{8,1}$=7.000, $X_{9,3}$=2.250, $X_{10,4}$=7.000, $X_{11,6}$=10.000, $X_{12,2}$=2.250, $X_{12,3}$=6.750, $X_{13,2}$=8.000, $X_{14,2}$=1.000, $X_{14,6}$ = 5.000.	**80.2500(Acceptable) Budget-Rs. 8,265.00**
		S B D	01	$X_{8,1}$ = 3.500, $X_{13,2}$ = 8.000.	11.5000
		S D B	09	$X_{1,5}$=10.000, $X_{2,5}$=5.000, $X_{3,4}$=3.750, $X_{4,4}$=8.000, $X_{5,1}$=4.250, $X_{8,1}$=7.000, $X_{9,3}$=2.250, $X_{10,4}$=7.000, $X_{11,6}$=10.000, $X_{12,2}$=2.250, $X_{12,3}$=6.750, $X_{13,2}$=8.000, $X_{14,2}$=1.000, $X_{14,6}$ = 5.000.	**80.2500 (Acceptable) Budget-Rs. 8,265.00**
3	50%	B D S	01	$X_{8,1}$ = 3.5417, $X_{13,2}$ = 7.5000.	11.0417
		B S D	01	$X_{8,1}$ = 3.5417, $X_{13,2}$ = 7.5000.	11.0417
		D B S	01	$X_{1,3}$=6.0000, $X_{1,4}$=12.5000, $X_{1,5}$=10.0000, $X_{8,1}$=7.5000, $X_{11,6}$=10.0000, $X_{13,2}$=7.5000.	**53.5000 (Acceptable) Budget-Rs. 3,855.00**
		D S B	03	$X_{1,5}$=10.0000, $X_{2,4}$=5.0000, $X_{4,4}$=0.5000, $X_{6,1}$=0.5000, $X_{8,1}$=7.0000, $X_{10,4}$=7.0000, $X_{11,6}$=10.0000, $X_{12,3}$=6.0000, $X_{13,2}$=7.5000.	**53.5000 (Acceptable) Budget-Rs. 4,603.00**
		S B D	01	$X_{8,1}$ = 3.5417, $X_{13,2}$ = 7.5000.	11.0417
		S D B	03	$X_{1,5}$=10.000, $X_{2,4}$=5.000, $X_{4,4}$=0.500, $X_{6,1}$=0.500, $X_{8,1}$=7.000, $X_{10,4}$=7.000, $X_{11,6}$=10.000, $X_{12,3}$=6.000, $X_{13,2}$=7.500.	**53.5000 (Acceptable) Budget-Rs. 4,603.00**

7 Discussion and Implication for Management

In general methods of Transportation Problem, one should concern mainly for its budget. After converting the problem in GP model, options explored for DM to provide preference in either demand or supply or budget as he wishes to fulfill the preferences specified by the management. It is observed from the obtained solution table that the solution with 100% demand level dominates over other two solutions from decision point of view. More interactive solutions can also be provided by changing demand goals and their priority levels. The solution obtained by GP formulation of TP is good over general methods and having open

opportunity for DM with budget, demand and supply. But, none of these methods supply the commodities to all customers.

In multiple objective decision problems we can not take a decision randomly because it may hamper the future prospects. That's why we required prioritization/ranking of the objectives according to their importance. Suppose, according to priority, the objectives (for supply as per demand) at first two priorities are satisfied but others are less or poorly satisfied. Then a question arises – Is this solution the best one? Of course not, from the management point of view, because so far supply of commodity to a number of customers is concerned, it is certainly not fair to satisfy one or two customers only leaving others waiting. Thus in the present scenario it is advisable that a company must have a good number of customers and should try to supply the commodities to all customers up to a justified level. The same idea is the outcome of the present paper, which follows the solution obtained on mathematical formulation of transportation problem using the GA technique.

8 Conclusion

Decision taken by DM is the best compromise one; even it requires support from the managerial team. And to get a good decision support we generally need help/advice of professionals. The used GA technique permits the DM to supply the every different type of commodities to all customers up to a justified level. This is certainly a development over transportation problems, which helps DM of a company as well as customers.

9 Future Prospect

Incorporating any developed learning strategy in the Genetic Algorithm Technique may be helpful for the improvement in the result.

Acknowledgment. The authors acknowledge to Prof. Darogi Singh, (Retd.), B.I.T., Sindri, Jharkhand.

References

[1] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Pearson Education, New Delhi (2004)
[2] Davis, L.: Handbook of Genetic Algorithms. Van Nostrand, Reinhold (1991)
[3] Prakash, S.: A Transportation problem with the objective to minimize total cost and duration of transport. Operation Research 18, 235–238 (1981)
[4] Lee, S.M., Moore, L.G.: Optimizing transportation problems with multiple objectives. AIIE Transaction 05(5), 333–338 (1973)
[5] Charnes, A., Cooper, W.W.: Management models and Industrial application of linear programming. John Wiley and Sons Inc., New York (1961)
[6] Charnes, A., Cooper, W.W.: Goal programming and multiple objective optimizations. European Journal of Operational Research 01, 307–322 (1977)

[7] Ignizio, J.P.: Generalized goal programming an over view. Computers and Operations Research 10, 277–289 (1983)
[8] Ignizio, J.P.: Goal programming and extensions. Lexington Books, D. C. Heath & Co., Massachusetts (1976)
[9] Lee, S.M.: Goal programming for Decision Analysis. Auerback, Philadelphia (1972)
[10] Eshelman, L.J., Schaffer, J.D.: Real-coded Genetic Algorithms and Interval Schemata. In: Whitley, D.L. (ed.) Foundation of Genetic Algorithms II, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
[11] Wright, A.H.: Genetic Algorithms for Real Parameter Optimization. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms I, pp. 205–218. Morgan Kaufmann, San Mateo (1991)
[12] Janikow, C.Z., Michalewicz, Z.: An Experimental Comparison of Binary and Floating Point Representation in Genetic Algorithms. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 31–36. Morgan Kaufmann, San Francisco (1991)
[13] Goldberg, D.E., Deb, K.: A Comparison of Selection Schemes Used in Genetic Algorithms. In: Foundation of Genetic Algorithms I, FOGA-I, vol. 01, pp. 69–93 (1991)
[14] Deep, K., Thakur, M.: A New Crossover Operator for Real Coded Genetic Algorithms. Applied Mathematics and Computation 188, 895–911 (2007)
[15] Deep, K., Thakur, M.: A New Mutation Operator for Real Coded Genetic Algorithms. Applied Mathematics and Computation 193, 211–230 (2007)
[16] Dubey, O.P., Singh, M.K., Dwivedi, R.K., Singh, S.N.: Interactive Decisions for Transport Management: Applications in the Coal Transportation Sector. The IUP Journal of Operations Management 10(02), 07–21 (2011)

A New Real Coded Genetic Algorithm Operator: Log Logistic Mutation

Kusum Deep, Shashi, and V.K. Katiyar

Department of Mathematics, Indian Institute of Technology Roorkee, India
{kusumfma,vktmafma}@iitr.ernet.in, shashibarak@gmail.com

Abstract. In this study, a new mutation operator for real coded genetic algorithms called the Log Logistic Mutation (LLM) is proposed. The performance of LLM is compared with existing real coded mutation operator namely Power Mutation (PM). LLM is used in conjunction with a well known crossover operator; Laplace Crossover (LX), to obtain a new generational real coded genetic algorithm called LX-LLM. LX-LLM is compared with the existing LX-PM. The performance of both the genetic algorithms is compared on the basis of success rate, average function evaluation, average error and computational time, and the supremacy of the proposed mutation operator is established.

Keywords: Real Coded Genetic Algorithm, Mutation Operator, Crossover Operator.

1 Introduction

A number of real life complex problems arising in the field of Engineering, Science, Industry and Finance turn out to be nonlinear optimization problems which may have constraints associated with them. For this reason, the demand for designing efficient, reliable and robust numerical optimization techniques is very high. There are a number of stochastic optimization techniques which are being used to solve complex optimization problems. Among these Genetic Algorithms (GA) are found to be very promising global optimizers. GAs are population based heuristics which mimic the Darwin's principal of "survival of fittest". The concept of GA was given by Holland [4]. A detailed implementation of GA could be found in Goldberg [1]. Although initial versions of GA uses binary numbers for encoding of chromosomes but their major drawbacks such as lack of precision and existence of "Hamming-cliff" problem impel to look for another type of encoding scheme.

To overcome these difficulties related to binary encoding of continuous parameter optimization problems, real-encoding of chromosomes is used. GAs which make use of real encoding of chromosomes are termed as Real Coded GA (RCGA).

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 193–200.
springerlink.com

Although a lot of effort has been put into the development of sophisticated real coded crossover operators to improve the performances of RCGAs for function optimization. But relatively less effort has been put into designing new mutation operators for RCGAs.

The most important aspect of mutation operator is that it prevents premature convergence of an algorithm. Mutation operator provides random diversity in the population as described in [10]. There are two issues which are of utmost importance for application of mutation operators. The first is the probability of mutation and the second is the strength of mutation. Detailed study about these can be found in [2,5,6]. For brief review of mutation operators see [7]. A novel mutation mechanism is also suggested, which is inspired by velocity update equation of PSO [3]. In [9] the effectiveness of a novel mutation operator based on immunity operation is demonstrated.

In an endeavor to define new operators for real coded genetic algorithms, in this paper, a new mutation operator called Log Logistic Mutation (LLM) is presented. Combining LLM with Laplace Crossover (LX) [8], a new generational RCGA called LX-LLM is designed. It is compared with existing RCGA, LX-PM. In order to establish the strength of Log Logistic mutation computational analysis is performed and discussed.

The paper is organized as follows: the proposed Log Logistic Mutation is defined in Section 2. In Section 3, new RCGAs based on Log Logistic Mutation is discussed. In Section 4, the experimental setup of all the GAs is explained. The computational results and discussion is given in Section 5. Finally in Section 6, the conclusions are drawn.

2 The Proposed Mutation Operator

The proposed mutation operator is based on Log-Logistic distribution and is represented as Log-Logistic mutation (LLM). Its density function is given by

$$f(x)=\begin{cases}\dfrac{\alpha(x/\beta)^{\alpha-1}}{\beta[1+(x/\beta)^{\alpha}]^2} & if\ x>0\\ 0 & otherwise\end{cases} \tag{1}$$

and distribution function is given by

$$F(x)=\begin{cases}\dfrac{1}{1+(x/\beta)^{-\alpha}} & if\ x>0\\ 0 & otherwise\end{cases} \tag{2}$$

where $\alpha>0$ is shape parameter and $\beta>0$ is scale parameter.

A mutated solution M is created in the vicinity of the solution P as follows:
Generate a uniformly distributed random number $r\in[0, 1]$,

if $r < T$; then

$$M = P - \lambda(P - L) \tag{3}$$

else

$$M = P + \lambda(U - P) \tag{4}$$

where L and U are the lower and upper bounds of decision variable, $T = (P - L)/(U - P)$ and λ is a random number following Log-Logistic distribution and is given by:

$$\lambda = \beta(h/1-h)^{1/\alpha} \tag{5}$$

where $h \in (0,1)$ is a uniformly distributed random number. The density function of Log-Logistic distribution for $\alpha = 1/2$ and $\alpha = 1/4$ is shown in Fig. 1. It is clear from the figure that for small values of α less disturbance in the solution is expected where as for large values of α more diversity is achieved. The parameter α is termed as mutation index as it controls the strength of mutation.

Result: *The mutated solution obtained from Log Logistic Mutation (LLM) will always be feasible.*

The proof of the result is given below:

Given: M is the mutated solution obtained from the parent P and $L \le P \le U$. $\lambda \in (0,1)$ is a random number following Log-Logistic distribution. Also either $M = P - \lambda(P - L)$ or $M = P + \lambda(U - P)$.

To Show: $L \le M \le U$.

Since $0 < \lambda < 1$.

$\Rightarrow 1 - \lambda > 0$ and $P - L \ge 0$

$$\Rightarrow (P - L)(1 - \lambda) \ge 0$$
$$\Rightarrow (P - L) - \lambda(P - L) \ge 0$$
$$\Rightarrow P - \lambda(P - L) \ge L$$
$$\Rightarrow M \ge L \tag{6}$$

Also $1 - \lambda > 0$ and $U - P \ge 0$

$$\Rightarrow (U - P)(1 - \lambda) \ge 0$$
$$\Rightarrow (U - P) - \lambda(U - P) \ge 0$$
$$\Rightarrow U - P \ge \lambda(U - P)$$
$$\Rightarrow U \ge P + \lambda(U - P)$$
$$\Rightarrow U \ge M \tag{7}$$

from equations (6) and (7) we get $L \le M \le U$.

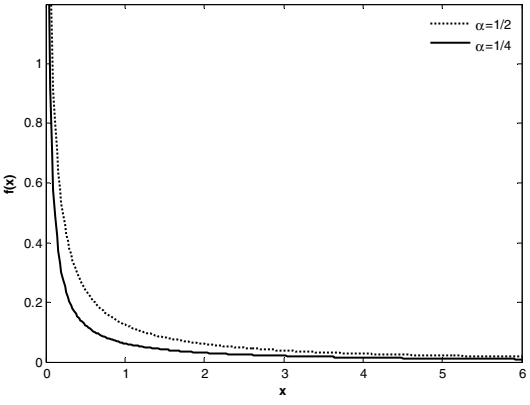


Fig. 1. Density function of Log Logistic distribution for $\beta = 1 and \alpha = 1/2, \alpha = 1/4$

3 The Proposed New RCGA

In this section a new RCGA namely LX-LLM is proposed, which combines the use of Laplace crossover with Log Logistic mutation. LX-LLM is compared with the existing LX-PM. Both the algorithms use tournament selection with tournament size three. In addition both the RCGAs are elite preserving with elitism size one. The elitism operator is put into effect to maintain fitness stability helping to increase the search performance of the proposed algorithm.

Table 1. List of Nonscalable benchmark functions

S.No	Function	Mathematical formula	Range	Min
1	Easom 2D	$Min\ f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	[-10, 10]	-1
2	Becker and Lago	$Min f(x) = (\lvert x_1\rvert - 5)^2 + (\lvert x_2\rvert - 5)^2$	[-10, 10]	0
3	Bohachevsky 1	$\mathbf{Min}\ f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	[-50, 50]	0
4	Eggcrate	$\mathbf{Min}\ f(x) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$	[-2 π , 2 π]	0
5	Periodic	$\mathbf{Min}\ f(x) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1\exp(-x_1^2 - x_2^2)$	[-10, 10]	0.9

Both the RCGAs are terminated if: either the prespecified maximum number of generations (3000) is reached or the best solution obtained by the algorithm lies within specified accuracy (0.01) of known optimum, whichever occurs earlier. The algorithm is tested on standard benchmark problems taken from literature

which includes: 5 nonscalable test problems (Table 1) and 5 scalable test problems of size 30 (Table 2) of different levels of complexity and multimodality. The pseudo code of LX-LLM is given below:

Algorithm

Step 1 (Initialization):

- ***Initialize population;***
- ***Set Generation=0;***

Step 2(Evaluation): Evaluate the fitness for each individual

Step 3(Termination check): Check the termination criteria If satisfied stop; else goto 4.

Step 4 (GA Operations)

- ***Select individuals according to selection algorithm to build a mating pool***
- ***Crossover the population in mating pool with given crossover probability***
- ***Mutate the current population with given mutation probability***

Step 5 (Replacement): Replace the old population with new population while retaining the best individual for next generation i.e. apply elitism with size 1

Step 6

- ***Evaluate the best fitness and find optimal individual***
- ***Increment generation; go to step 3.***

Table 2. List of Scalable benchmark functions

S.No	Function	Mathematical formula	Range	Min
1	Paviani	$\min_x f(x)=\sum_{i=1}^{n}[(\ln(x_i-2))^2+(\ln(10-x_i))^2]-\left(\prod_{i=1}^{n}x_i\right)^{0.2}$	$[2, 10]^{30}$	-997807.705158
2	Rastrigin	$\min_x f(x) = 10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$	$[-5.12, 5.12]^{30}$	0
3	Schwefel	$\min_x f(x) = -\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-500, 500]^{30}$	-12569.48
4	Sinusoidal	$\min_x f(x) =-\left[2.5\prod_{i=1}^{n}\sin\left(x_i-\frac{\pi}{6}\right)+\prod_{i=1}^{n}\sin\left(5\left(x_i-\frac{\pi}{6}\right)\right)\right]$	$[0, \pi]^{30}$	-3.5
5	Zakharov's	$\min_x f(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}\frac{i}{2}x_i\right)^2+\left(\sum_{i=1}^{n}\frac{i}{2}x_i\right)^4$	$[-10, 10]^{30}$	0

4 Experimental Setup

The experimental setup used for both the RCGAs viz. LX-LLM and LX-PM is given in this section.

Parameter Setting: Population size is taken as 10 times the number of variables. The mutation index for LLM is fixed at 2. The final parameter setting is presented in Table 3 where p_c and p_m represents the probabilities of crossover and mutation

respectively. Each GA is run 100 times with same initial populations while each run is initiated using a different set of initial population.

All the algorithms are implemented in C++ and the experiments are done on a Core Duo Processor with 1.66GHz speed and 1 GB RAM under WINXP platform.

Table 3. Parameter Setting for LX-LLM and LX-PM

Algorithm	*Nonscalable*		*Scalable*	
	p_c	p_m	p_c	p_m
LX-LLM	0.70	0.020	0.60	0.008
LX-PM	0.70	0.007	0.60	0.005

Performance Evaluation Criteria: A run in which the algorithm finds a solution satisfying $f_{\min} - f_{opt} \leq 0.01$, where $f_{\min}$ is the best solution found when the algorithm terminates and f_{opt} is the known global minimum of the problem, is considered to be successful.

For each method and problem the following are recorded:

- Success Rate (SR) = $\frac{Numberof\ successful\ runs}{Total\ numberof\ runs} \times 100$
- Average computational time (ACT) (in seconds).
- Average number of function evaluations (AFE).
- Average Error (AE) = $\frac{\sum_{n}(f_{\min} - f_{opt})}{n}$ where, n is the total number of runs.

5 Results and Discussion

The computational results of both the RCGAs are presented in this section. Table 4 summarizes the computational results for both the algorithms. As the execution time for nonscalable problems is very less and hence insignificant in most of the cases, so it is not included in the Table 4. From Table 4 it can be easily observed that the proposed mutation operator LLM outperforms PM in terms of various performance criteria and thus indicates the efficiency of proposed mutation operator for providing suitable diversity for obtaining the global optimal solution.

Table 4. Computational Results for Nonscalable and Scalable problems

		Nonscalable Problems				Scalable Problems			
Fun	Method	AE	SR	AFE	Fun	AE	SR	AFE	ACT
1	LX-PM	0.00625	100	378	1	0.00865	100	139364	3.43
	LX-LLM	0.00531	100	325		0.00601	83	117735	2.38
2	LX-PM	0.00612	100	4059	2	0.01154	77	883517	31.35
	LX-LLM	0.00556	100	282		0.00621	100	170659	4.51
3	LX-PM	0.00884	94	649	3	0.00756	100	128034	2.82
	LX-LLM	0.00521	100	528		0.00682	100	125105	2.73
4	LX-PM	0.01102	66	12718	4	0.00805	100	46486	1.06
	LX-LLM	0.00583	100	452		0.00741	100	41981	0.89
5	LX-PM	0.02115	44	1394	5	0.00874	100	336236	6.82
	LX-LLM	0.01331	64	883		0.00532	100	279067	6.47

6 Conclusion

In this paper a new real coded mutation operator called Log Logistic Mutation (LLM) is introduced. By combining LLM with already existing operator, namely LX a new generational RCGA called LX-LLM is proposed. For an analogical comparison it is compared with the existing RCGA based on Laplace crossover and Power mutation called LX-PM. Based on the numerical results it is clear that with respect to *reliability*, *efficiency* and *accuracy* measured in terms of success rate, function evaluation and average error respectively, **LX-LLM** performed better than LX-PM, though they both use same crossover operator, which clearly signifies the importance of mutation operators in the search process.

Thus LLM proved to be an efficient mutation operator when used with Laplace crossover. But will it work equally well with other crossover operators for RCGAs is the question whose answer is to be looked in future studies.

References

[1] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, New York (1989)

[2] Herrera, F., Lozano, M.: Two-Loop real coded genetic algorithms with adaptive control of mutation step sizes. Applied Intelligence 13, 187–204 (2002)

[3] Tsoulos, I.G.: Modifications of real code genetic algorithm for global optimization. Applied Mathematics and Computation 203, 598–607 (2008)

[4] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan press, Ann Arbor (1975)
[5] Smith, J.E., Fogarty, T.C.: Operators and parameter adaptation in genetic algorithms. Soft Computing 1, 81–87 (1997)
[6] Smith, J.E.: Modelling GAs with self adapting mutation rates. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 599–606. Morgan Kaufmann, San Francisco (2001)
[7] Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Applied Mathematics and Computation 193, 229–247 (2007)
[8] Deep, K., Thakur, M.: A new crossover operator for real coded genetic algorithms. Applied Mathematics and Computation 188, 895–911 (2007)
[9] Xing, L.N., Chen, Y.W., Yang, K.W.: A novel mutation operator based on the immunity operation. European Journal of Operational Research 197, 830–833 (2009)
[10] Spears, W.M.: Crossover or mutation? In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms, vol. 2, pp. 221–238. Morgan Kaufmann, San Mateo (1993)

Trust Management Model for Wireless Ad Hoc Networks

Brijesh Kumar Chaurasia[1,*] and Ranjeet Singh Tomar[2]

[1] ITM University, Gwalior, India
[2] Indian Institute of Information Technology,
Allahabad, India
bkchaurasia.itm@gmail.com, rs63@iiita.ac.in

Abstract. Wireless ad hoc networks consist of a set of nodes equipped with wireless interfaces. The mobile / static nodes dynamically create a wireless network without using any infrastructure. In the absence of infrastructure authentication and trust management is an issue for nodes in wireless ad hoc networks. The proposed trust management model allows nodes to evaluate the trust by taking into account certificate of other node(s) to overcome vulnerabilities. In this paper, we have addressed the trust management using certificate based approach. Simulation results show that the time taken by the scheme is significantly less and transmission over the shared wireless channel is in the order of milliseconds for different traffic conditions in wireless ad hoc networks. This proves the efficacy of the scheme and makes it attractive for MANET and VANET.

Keywords: MANET, Certificates, VANET.

1 Introduction

Wireless ad hoc network is a group of mobile/ static nodes. These nodes may dynamically create a wireless network without using any infrastructure. Wireless ad hoc networks can be categorized into infrastructure aided networks such as cellular networks, infrastructure less networks as mobile ad hoc networks, sensor ad hoc networks and hybrid networks such as vehicular ad hoc networks (VANET), worldwide interoperability for microwave access network. In wireless ad hoc networks, node includes laptop, computers, PDAs and wireless phones etc, have a limited transmission range. In trust management model, trust is a belief level that one entity can put on another entity for a specific action based on previous direct or indirect observations on behaviors of this entity [1], [2].

* Corresponding author.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 201–206.
springerlink.com

In mobile ad hoc networks (MANET) entities are mobile nodes. In MANETs, each node can communicate directly with only few nodes within the communication range and has to forward messages using the neighboring nodes until the messages arrive at the destination nodes. MANETs are especially useful in military and other applications such as emergency rescues [1]. However, due to nature of transmission in the shared wireless medium reaches all the nodes in the communication range, the nodes may misbehave selfishly for individual interests [3]. Authentication, privacy and trust management of entities are main issues of mobile ad hoc networks. In this paper, we propose trust management model to support evaluation of information sent by peer nodes and immediately detect and control false information or misbehaving nodes.

In vehicular ad hoc network, entity can be vehicles and road side units. In wireless networks, trust avoids the usage of central trusted authority and used to record feedback about the security evaluations of other entities in those networks [4]. In VANET, trust is a relationship between two vehicles such that one vehicle believes, expects, and accepts that the other trusted vehicle will act or intend to act beneficially [5], [6]. Existing trust management frameworks are categorized in two categories: trust management [3],[7],[8],[9] and repudiation based [10], [11]. Another classification of trust models are evidence based model and recommendation based model [12]. In evidence based model entities make a trust relationship based on evidence such as keys, certificate, timestamp etc. while recommendation based model establish a trust relationship based on unknown entities. The proposed model is used evidence based approach for trust management in wireless networks.

2 Proposed Trust Management Model

In this section, trust management protocol model is presented for wireless ad hoc networks i.e. MANET and VANET.

In VANET, a base station (RSU) continuously broadcasts its identity ID_{RSU} in its area. As a node $(vehicle)$ enters the area of a base station, it receives base station broadcast and determines the needs to associate itself and initiate certificate process. The base station confirms the authentication of the node itself or using any trusted authority. In case of VANET, central trusted authority (CTA) will issue a certificate after authentication through base station [13]. After receiving certificate, a node can establish trust on another node in the absence of infrastructure. Certification distribution in case of MANET can be in similar manner or by using base station. In the proposed trust management model [14], [15], the properties to the formation of trust in MANET are: subjective, context dependent, dynamic, asymmetric and time dependent. The main exchanges between node and base station are shown in Fig. 1. The whole process needs only one request and reply between node $(vehicle)$ and base station (RSU).

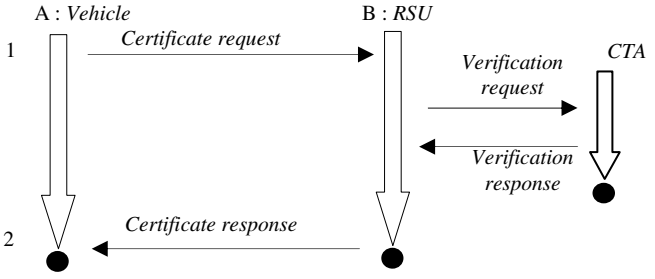


Fig. 1. Certificate Process- Main Messages

After receiving certificate, a node can establish trust on another node the basis of seeing the certificate. The structure of certificate is depicted by Fig.2. In trust management model, a node first sees the certificate identity of issuer authority and the time stamp. After certain threshold (validity), certificate will be expired. On the basis of fields of certificate structure, a node can establish the trust. The trust can be also established on the basis of signal strength, time of receiving the packets. In case of VANET it is easier as compared in MANET due to predictable topology. Certificate structure consists the three fields in proposed trust management model. Certificate identity is one byte and two byte other field is allocated for subject / context, which will depend on wireless ad hoc network application and requirement. Last two byte field is assigned for validity of certificate or time stamp. This certificate length is very small as compared of other existing trust model approaches. Packet transmission delay is presented in next section.

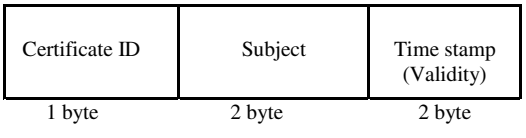


Fig. 2. Certificate Structure

3 Simulation and Results

In this section, the efficiency of the proposed trust model is verified through simulation using NCTUns [16] for VANET and NS-2 [17] for MANET. The computation platform was a desktop (CPU speed - 2.50 GHz, RAM - 2 GB RAM).

3.1 Simulation Setup and Results for VANET

Data packets were generated at a constant bit rate at *RSU* and the vehicle. Fig. 3 shows the average and maximum communication delay when number of vehicles in the *RSU* region varies from 5-40. The speed of vehicles was varied between 10-30 ms^{-1} with acceleration / deceleration = 3 ms^{-2} . Maximum delay time of certificate received is 70.4882 ms. and 104.0115 ms. at the speed of 10 m/s and 30 m/s respectively.

3.2 Simulation Setup and Results for MANET

The MANET topology is simulated in NS-2. It consists of a rectangular plane (1500m x 1500m) and 50 nodes are uniformly distributed over this plane. Data is uniform constant bit rate traffic with packet size of 512 bytes was used for the data traffic. The average speed of each of the nodes varies from 0- 20 ms^{-1}. Each simulation was run for a period of 300 seconds. Fig. 4 shows the packet delivery delay in MANET scenario using AODV and DSR protocol. Maximum time delay of certificate received is 570 ms. and 500 ms. using AODV and DSR routing protocol respectively. The proposed model takes very less time in terms of milliseconds in VANET as well as MANET.

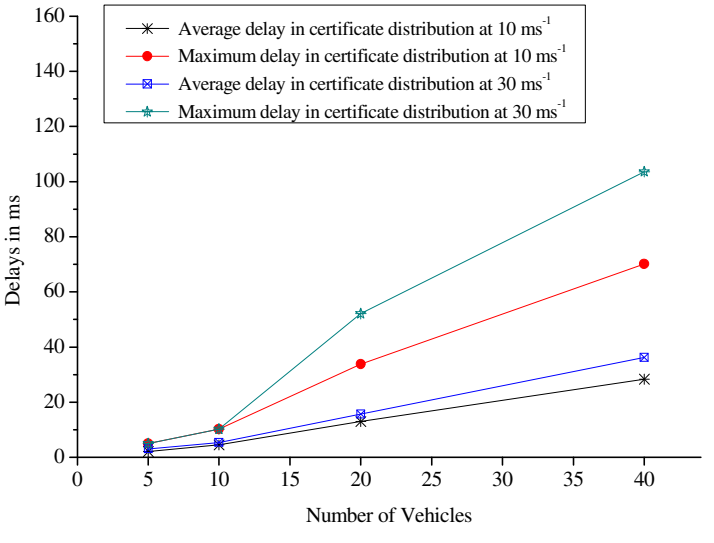


Fig. 3. Delays of Certificate distribution in VANET

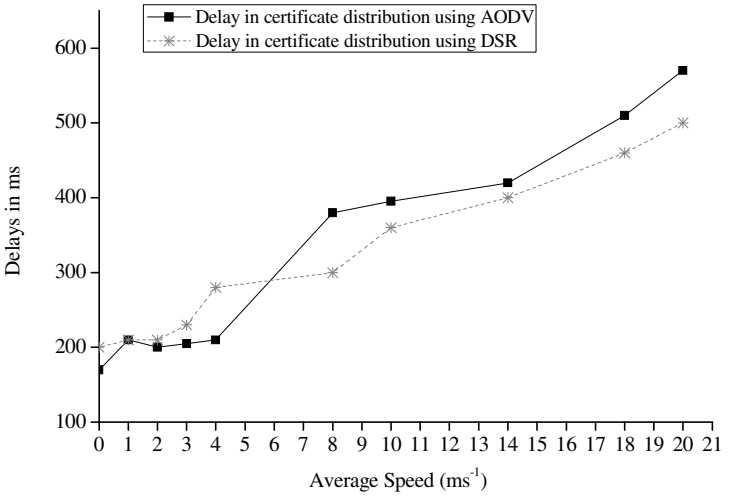


Fig. 4. Delays of Certificate distribution in MANET

4 Conclusion

In this paper, a trust management model is presented to establish trust for MANET and VANET. We design an evidence based trust model using certificate approach to handle ad hoc messages. Network simulation experiments shown that significant performance gains in terms of time delay using certificate based trust model is presented. The current approaches are not very sufficient for trust management of wireless mobile nodes. This research work also plans to contribute with location aware privacy preserving protocol implementing the trust model in the absence of infrastructure.

References

1. Li, J., Li, R., Kato, J.: Future Trust Management Framework for Mobile Ad Hoc Networks. IEEE Communications Magazine 46(4), 108–114 (2008)
2. Sun, Y.L., Han, Z., Yu, W., Liu, K.J.R.: A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks. In: Proceedings 25th IEEE International Conference on Computer Communications, Barcelona, Spain, pp. 1–13 (2006)
3. Chaurasia, B.K., Verma, S.: Maximising Anonymity of a Vehicle. International Journal of Autonomous and Adaptive Communications Systems (IJAACS), Special Issue on: Security, Trust, and Privacy in DTN and Vehicular Communications 3(2), 198–216 (2010)
4. Theodorakopoulos, G., Baras, J.: Trust evaluation in ad-hoc networks. In: Proceedings of ACM Workshop on Wireless Security, USA, pp. 1–10 (2004)

5. McKnight, D.H., Chervany, N.L.: The Meanings of Trust, Technical Report, University of Minnesota (1996)
6. Boukerche, A., Ren, Y.: A trust-based security system for ubiquitous and pervasive computing environments. Computer Communications 31, 4343–4351 (2008)
7. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: 6th Annual International Conference on Mobile Computing and Networking (MobiCom-2000), pp. 255–265 (2000)
8. Theodorakopoulos, G., Baras, S.: On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks. IEEE JSAC, Special Issue on Security in Wireless Ad Hoc Networks, 318–328 (2006)
9. Zouridaki, C.: A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs. In: Proc. 3rd ACM Wksp. Sec. Ad Hoc and Sensor Networks, pp. 1–10 (2005)
10. Buchegger, S., Le Boudec, J.-Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes — Fairness in Dynamic Ad-hoc NeTworks). In: Proc. ACM MobiHoc 2002, pp. 226–236 (2002)
11. Ding, Q., Li, X., Jiang, M., Zhou, X.: Repudiation Management in Vehicular Ad Hoc Networks. In: International Conference on Multimedia Technology (ICMT), pp. 1-5 (2010)
12. Li, H., Singhal, M.: Trust Management in Distributed Systems. Computer 40(2), 45–53 (2007)
13. Chaurasia, B.K., Verma, S.: Infrastructure based authentication in VANETs. International Journal of Multimedia and Ubiquitous Engineering 6(2), 41–54 (2011)
14. Bakar, A.A., Ismail, R., Jais, J., Manan, J.-L.A.: Forming Trust in Mobile Ad-Hoc Network. In: International Conference on Communications and Mobile Computing (CMC 2009), pp. 470–474 (2009)
15. Capra, L.: Engineering human trust in mobile system collaborations. In: SIGSOFT FSE, pp. 107–116 (2004)
16. NCTUNs: Network simulator and emulator, `http://nsl.csie.nctu.edu.tw/nctuns.html` (accessed on November 2008)
17. NS2: The Network Simulator ns-2, `http://isi.edu/nsnam/ns` (accessed on November 2008)

Dynamic Angle Calculation for Fast Routing in GPS Assisted MANETS

Reji Mathews, Amnesh Goel, Sukanya Ray, and Kamal Kant Ahirwar

Amity School of Engineering & Technology,
Amity University, Noida (U.P.), India
{contactreji,amneshgoel7,sukanyaray007,kamalkant25}@gmail.com

Abstract. The Ad Hoc set up of networked nodes has become a useful concept for setting up a temporary communication system among the participating communicating nodes. Such systems are variedly deployed in areas where there is no feasibility for establishing regular communication lines and wireless networking solutions. The effective routing among these systems becomes a challenge since reaching from one node to another may be achieved through the involvement of various intermediate nodes. The routing protocol should minimize the route calculation delay and provide best possible path from source to destination. The assistance from GPS receivers is used to calculate of best route and selects the most promising node at each step which can get the route to the final receiving node using its latitude and longitude as heuristics.

Keywords: MANET, GPS assisted, routing, angle, backtrack.

1 Introduction

There has been an increasing application of the MANETs in the geographical areas where the deployment of regular telecommunication networks have proved to be non-feasible. The areas include the drought and flood affected areas where the entire communication links and the wireless hot spots are damaged and needs a huge capital in terms of money and time to be restored the working state. Such situations need to be addressed with ad hoc setting up of the communication links especially for the rescue personnel team for coordination in their work process and for interfacing with the outside world to request assistance in terms of basic rescue goods like food packets, medical attention for the affected as well as media assistance. Hence such needs to be addressed on time effectively need a strong and reliable communication links to be established with high stability and less down time.

In the above mentioned situations, the most effective way to set up a communication system is to set up an ad hoc MANET. The MANET consists of number of nodes and each of these nodes has its own power supply and processing power. In such devices, there may not be a direct link between the main sending

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 207–218.
springerlink.com

node and the receiving node. In most of the cases, there may be a need for route calculation with multiple nodes acting as carriers of message between the main sending node and the receiver node. The need to select the next best node among those in the range of sender is an important decision to be made after processing its routing algorithms and assisting heuristics.

The routing algorithms should be highly efficient enough so as to avoid bottlenecks at any node in terms of traffic density and should succeed in selecting the best route to minimize delay and unwanted hoping around of packets infinitely among the nodes.

In this case we consider the situation where each node is assisted with a GPS [1][2] receiver on board which is capable of calculating the position of the device with respect to its Latitude and Longitude. The accuracy of the GPS devices in the current era has been extremely improvised by extensive research and optimized hardware specifications. Today's GPS devices have achieved the accuracy levels by limiting error margins to less than 3 meters.

2 Existing Approaches of Routing in MANETS

The current approaches of routing used in MANETS can be classified into couple of major categories. They include Reactive, Pro-Active and Hybrid Routing protocols.

Proactive approach can be considered to be one of the most straight forward approaches adopted in the simpler and less complex MANETS where the number of mobile node in network are less. The routing algorithm takes into consideration the pre-determined route or next hope attribute for each packet coming to it carrying the address of the destination node in a different network. The node decides upon the next hop address with the pre available information fed by the network administrator or generated best route details from the previous experiences and systems which has been imported into the current machine and stored in form of tables. The random nature of an ad hoc network and the prerequisite of quick response of nodes make the idea of proactive protocol more appropriate. As soon as any request arrives, the control layer checks whether the pre computed optimal route can fulfill such a request or not and if this pre computed optimal path suits well then waste of network resources when attempting to discover infeasible routes can be avoided. The main issue with this routing mechanism is the data communication delay if any of the pre-defined paths ends in a temporary or permanent failure. Though the protocol ensure the fast delivery of the packets to the destination without much overhead on the intelligent nodes, it does not place a provision for the service fault tolerance if a link is broken or a better promising route is found.

Flooding [3] plays a major role in this approach as all the nodes have a set of predefine connecting routes with them in form of table and they refer this table as and when communication required. Flooding is a strategy which floods each and every data packets into the network without doing any calculation for the destination ensuring the guaranteed transfer of data packet to destination node

from source but on the other hand, some overhead is associated with it namely network congestion, extra processing by each node and security. It is not advisable to give low priority to security part as data can be confidential; network congestion may lead to delay in data packet transfer and in such routing protocol each node needs to do some extra calculations. Hence, in such ad hoc network, flooding must be avoided as far as possible.

A very popular and commercially used table driven routing protocol is Destination Sequenced Distance Vector (DSDV) routing protocol [4]. The routing table information of every node is broadcasted periodically as well as when any node changes its position. The Time complexity [5] is O(d) and Communication complexity becomes O(n) as every node knows the route to all the other nodes in the system. DSDV needs effective methods to check upon network congestion arising out of the need for periodic updates and transmission regardless of the rate of change of network topology. This calls for need to limit the number of nodes that can connect to the network as the overhead complexity becomes $O(n^2)$. In DSDV the network size is preferably designed to be small otherwise which the overhead will be too large.[15]

Optimized Link State Routing (OLSR) [18] protocol inherits the stability of link state algorithm. OLSR protocol performs hop-by-hop routing in which each node in the network uses its most recent information to send a packet. So, even though a node is travelling, its packets can be successfully delivered to it, if its speed is such that its path could be traced by its neighbors. The optimization in the routing can be done mainly in two ways: Firstly, OLSR reduces the size of the control packet for a particular node by declaring only the subset of links with the node's neighbors who are its multipoint relay selectors, instead of all links in the network. Secondly, it reduces flooding of control traffic by using only the selected nodes, called multipoint relays to disseminate information in the network. Since only multipoint relays of a node can retransmit its broadcast messages, this protocol significantly reduces the number of retransmissions in the flooding procedure.

The Reactive routing [6] was designed to overcome the issue in the proactive routing. The routing algorithm works on the principle of ON-DEMAND service. The major difference between the previously discussed pro-active routing and reactive routing the dynamic policy adopted in reactive routing. The route is neither pre-defined nor stored in any of the internal routing tables. The node which has to deliver a packet to any destination node calls for a route request packet to be generated and circulated across the network. This creates the problem of flooding across the network leading to extra cost involved in processing the route in terms of time and power consumption of the nodes. In MANETS, the power conservation of any node is a major cause of concern since there is least infrastructure involved in the Ad Hoc systems and everything is set up on the requirement on a temporary basis. This involves the participation by highly sophisticated and intelligent nodes. The major constraints on any nodes of the MANET are the limited availability of the power which has to be traded off with the performance in terms of making complex calculation and data activity.

Considering widely used Reactive protocols, Ad-Hoc On demand Distance Vector (AODV) routing protocol and Dynamic Source Routing (DSR) protocol have proved to be efficient in routing packets in MANETs. A notable concern in DSR is the overhead due to potentially large route request packets which carry a bulk of routing information. The route reply of DSR is also large since they contain the address of every node in addition to the route. The memory overhead in DSR is also a concern as it needs to remember all the valid routes.

AODV requires symmetric links between nodes to communicate and fails to do so in the presence of asymmetric links. For both the routing protocols the Time complexity is O (2d) and the Communication complexity is of O (2n). [15]

Cluster-Based Routing Protocol (CBRP) [16] is another popular on-demand routing protocol, where the nodes are divided into clusters. When a node comes to in the network, it is in the undecided state. The first task of this node is starting the timer and to broadcast a HELLO message and when a cluster-head receives the HELLO message, it replies with triggered HELLO message. Then when the node receives the answer, it changes its current state to the member state, but if the node receives no message from any cluster-head, it makes itself a cluster-head, if and only if it has bidirectional link to one or more neighbor nodes else, when it has no link to any other node, it stays in the undecided state and repeats the procedure with sending a HELLO message again. Each node has a neighbor table where the node keeps the status of the link and current state of each neighbor in the neighbor table. A cluster head keeps information about its every member in the same cluster and it also has a cluster adjacency table, which provides information about the neighboring clusters.

The hybrid routing [7] [8] [9] is a good blend of the advantages of pro-active and reactive routing strategies. The technique was developed to overcome the disadvantages of the above discussed protocols. The new protocol incorporated the advantages of proactive and reactive routing techniques and minimized effects of the overhead arising in the calculation and processing of the unwanted packets avoiding the arrival of duplicate packets at each node.

Zone Routing Protocol (ZRP) [10][11] is a hybrid Ad-Hoc routing protocol which uses proactive procedure for each node's local neighbor and reactive procedure for the node's global communication. ZRP utilizes the IntrAzone Routing Protocol (IARP), IntErzone Routing Protocol (IERP) and Bordercast Resolution Protocol (BRP) [12] with a query control mechanism. The scope of IARP is limited by the routing zone radius: the hop counts that the IARP route updates are relayed. A subset of the routing zone nodes is the group of nodes whose distance to the central node is less than or equal to the zone radius. These nodes are known as Peripheral nodes. Larger the radius, more nodes it can hold but again resulting in large network traffic as every node in the zone maintains a routing table containing routes to every other nodes in the zone. The routing tables are updated periodically. IERP is the global reactive routing protocol of ZRP. Instead of broadcasting ZRP bordercasts and the query spreads throughout the network. Bordercasting can be achieved in two ways: "root directed bordercast"

and "distributed bordercast". In the former, the source node and the peripheral node form a multicast tree and append the forwarding message in the routing packet query. This results in routing overhead and this overhead increases when the zone radius increases, thus resulting in a noticeable performance lag in ZRP. In distributed broadcast each node is required to maintain an extended routing zone. This results in increasing the local routing information exchange [3]. One major limitation of ZRP is the maintenance of uniform zone radius. Each node participating in the Zone Routing Protocol needs to have the same zone radius value. Thus ZRP seems inappropriate in heterogeneous scenarios where each node have different radius value and different rate of mobility. [4]

Neighbor-Aware Multicast Routing Protocol (NAMP) [19] is a tree-based hybrid routing protocol, that uses neighborhood information. The routes in the network are built and maintained using the traditional request and reply messages or on-demand basis. This protocol uses neighbor information that is two-hops away for sending the packets to the receiver. If the receiver node is not within the range, it searches the receiver node using dominant pruning flooding method [20] and thus forms a multicast tree using the replies received along the reverse path. As NAMP targets to achieve minimized end-to-end delay of packets, it uses the tree based structure. There are mainly three operations in NAMP: Multicast tree creation, Multicast tree maintenance and Joining and leaving of nodes from the multicast group. All the nodes in the network keep neighborhood information of nodes upto two-hops away. This neighborhood information is maintained using a proactive mechanism. Periodic hello packet is used for this process. For the creation of the multicast tree, the source node sends a flood request packet to the destination node with data payload attached. This packet is flooded in the network using dominant pruning flooding method, which actually reduces the number of transmissions in the network for a particular flood request packet. During the forwarding process of the packet, each node selects a forwarder and creates a secondary forwarder list (SFL) which contains information about the nodes that were primarily considered as possible forwarders but ultimately were not selected for that purpose. Each intermediate node uses the chosen forwarder to forward the packet, but keeps the knowledge about other possible forwarders in SFL. The link failure recovery is one of the greatest advantages of NAMP.

3 Proposed Method

After studying the various existing protocols and current methodologies used in the state of the art, there arises a need for a routing mechanism in which the each nodes have less dependency on its predecessor nodes for making any decisions on the next hop node. The node having the packet to be forwarded is assumed that it knows the geographical location details of the destination node.

By using the details of the latitude and longitude of the destination node, the quick calculation of the distance to the target and the relative angle to any reference line can be easily determined by simple calculations involving the geographical coordinates of the current node, the destination node and the nodes which lie in the direct vicinity of the current decision making node.The main principle behind the working of this proposed protocol is based on the effective use of target vectors calculated with the help of the GPS coordinates combined with the effective use of back tracking algorithm to use the advantages of BFS searching technique. The inefficiency of the pro-active routing algorithm used in the many conventional MANET routing systems which often led to the failure of message delivery was taken into consideration in the current proposed method.

In case the selected node fails to deliver the message to the desired recipient, the backtracking is adopted in the worst case situation to take a step back to the previous node and select the next promising node available among the list of known neighbors. The policy of simple Breadth-First-Search technique is adopted at this step of routing.

The working of proposed routing protocol can be illustrated in the following section. Consider the following scenario given in the figure (fig 1) below.

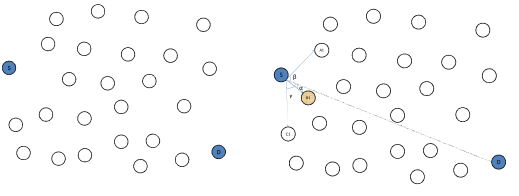


Fig. 1. A Scenario of Mobile Ad-Hoc network

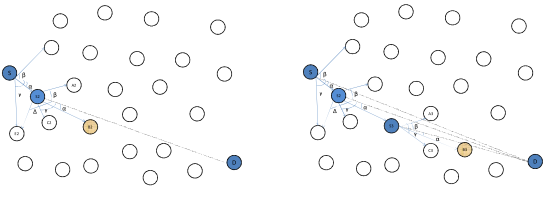


Fig. 2. Stage 2 where the minimum angle is selected and packet is passed

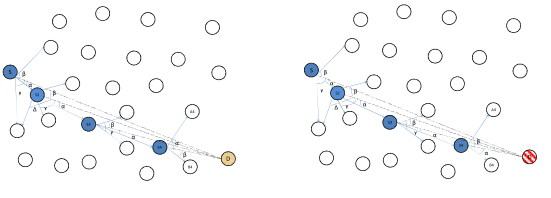


Fig. 3. Stage 3 where need for backtracking aroused

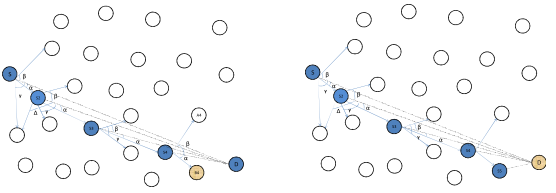


Fig. 4. Delivered after backtracking and choosing next option

4 Pseudo Code for Proposed Concepts

Begin

Set source node = S and destination node=D.

Initialize ConnectingTable[] with nodes direct connected to S.

Generate a direction vector between S and D to identify location of target node.

Calculate TargetAngle[] for all the nodes from ConnectingTable[] w.r.t. this direction vector.

Find the minimum angle from TargetAngle[].

Transfer data packet from S to node at least angle.

If transmission is successful then

S= node at least angle.

Else

Source = current node and choose next minimum angle from TargetAngle[].

End

If S=D then

End search

Else

Go to step 2.

End

End

The scenario has got 3 main components namely the source node denoted by S, the destination node denoted by D and the intermediate nodes lying on the path en-route the destination. Here we consider one node among the intermediate nodes represented by B.

The working of the proposed strategies takes place in the following steps.

a. The Source node (S) initially calculates the direction and distance vector to each adjacent nodes as well as the destination node using the knowledge of its GPS coordinates (assumed that the nodes are updated with the GPS coordinates of all other nodes in the ad hoc network using pre-existing algorithms used for sharing data among nodes in networks). The vectors S_i-A_i, S_i-B_i, S_i-C_i and S_i-D_i are generated in this step where i range from 1 to

k where k is the number of nodes through which the packet travels to get delivered to the final destination.

Now the node calculates the angle between the vectors which represents the location of the destination node and vectors connecting each of its adjacent nodes. In the above diagram, the calculation of the angles are done measuring the angle joining nodes B_i->S_i->D_i as ά, similarly A_i->S_i->D_i as β and Ci->Si->Di as γ.

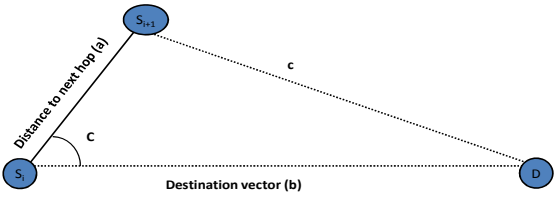


Fig. 5. Calculation of the Angle

The angle can be calculated by using the trigonometric Law of Cosine [17] formula $c^2 = a^2 + b^2 - 2ab\cos(C)$

On rearranging, $\cos(C) = (a^2+b^2-c^2)/2ab$

Where a, b and c are distance between the Geographical locations under consideration which is calculated using the Haversine Formula[13] and C is the angle which has to be calculated.

b. Now each of these angles calculated are sorted in the ascending order and the angle with the minimum value is selected and checked to get the corresponding immediate node with respect to which the angle was made. In the above scenario, the angle ά is considered and it's observed that the angle was formed by the destination vector S-D and the vector joining S and B.

c. The packet is immediately transferred to the node B and the source node is freed of any more calculations or tracking of the packet unless any delivery failure arises by all the successors of node B. In such situation, backtracking is performed.

d. Backtracking: Once the delivery attempt by all successors of node B is confirmed, the packet is now back tracked or returned to the source node S and in such case, it selects the next minimum angle greater than the angle ȧ which eventually led to a delivery failure state. From the figure (Fig 1) and calculation performed in step b, we chose the next node viz node A forming an angle $_\beta$ and which seems to be next available promising node.

The above steps are followed at the every intermediate node when the packet reaches each of them for next hop calculation. This protocol ensures that the overhead of calculation lies only with those nodes that currently hold the packet for the next relay. All other nodes are free to work upon the next incoming packet once its passes on its old packet to its successor.

5 Advantages of the New Approach

The main advantages of the proposed concepts can be studied thoroughly in comparison to the conventional protocols used in the systems which do not use the geographical location based information for decision making. The advantages of this protocol can be enlisted as follows:

5.1 The overhead is reduced to a considerable extend in terms of traffic, time and calculations at each nodes since there is no route request packets circulated at any point of time in the entire routing systems. This ensures faster delivery of the packets with considerable economy in terms of network latency.

5.2 Since there is no processing of the packets at other nodes outside the path of the packet route, a good amount of power conservation is achieved in the system. Only the nodes which appear to be the most promising ones and which seem to help in avoiding delivery failures are given the possession of the packets or bears the overhead of any calculation costs associated with the handling of packets.

5.3 Robustness and self-recovering capabilities of the MANET is highly enhanced using the proposed protocol. The processing logic at each node performs the back tracking of the packet routing control to the next promising available next-hop options at the previous node ensuring that the system itself derives the solution in case a dead end is reached or landing up in a state where the further processing of the forward path does not seem to be leading to any fruitful results.

5.4 At each step, the routing logic assures that the packet goes in the right direction where the destination node is situated. In majority of the situations, apart from the hop distance, at every hop the packet gets closer to the destination node in terms of its physical distance from the destination node.

5.5 The magnitude of the destination vector (distance between the line connecting source and destination) calculated from the source node can be optionally used to set a hop limit of the packets to limit or controlled the further unwanted processing by the nodes which seem to be leading to the destination but drifts far away from the destination after a fixed number of trials. This is very effective in situations where time is critical for packet delivery and the time costs involved in backtracking process can prove to be very expensive. The **Haversine** formula can be used to calculate the great-circle distance between two points when their GPS coordinates are available. The shortest distance over the earth's surface between the points (ignoring any hills, ridges, cliffs and other physical barriers) is given as

$$a = \sin^2(\Delta lat/2) + \cos(lat_1).\cos(lat_2).\sin^2(\Delta long/2)$$
$$c = 2.a.\tan2(\sqrt{a}, \sqrt{(1-a)})$$
$$\text{distance } (d) = R.c$$

Where R denotes the radius of the earth (approx 6371 kilometers) [13].

The following table shows the major comparative study of the existing protocol with the currently used protocols in Ad Hoc network routing [14].

Table 1. Comparison with other existing protocols

	DSDV	AODV	DSR	ZPR	Proposed
Time Complexity	O(d)	O(2d)	O(2d)	O(I)O(2d)	O(I)O(2d)
Communication Complexity	O(n)	O(2n)	O(2n)	O(Zn)O(V+n)	O(V+n)
Routing Structure	Flat	Flat	Flat	Flat	Flat
Routing Metric	SP	SP	SP	SP	SP
Flood for Route Discovery	Yes	Yes	Yes	Partially	Partially
Multipath possibilities	No	No	Yes	No	No
Periodic Message	Yes	No	No	Yes	May be
Loop Free	Yes	Yes	Yes	Yes	Yes
Multicast Capability	No	Yes	No	No	Yes
Source Routing	No	No	Yes	No	No
Functioning pro-actively	Yes	No	No	Yes (Locally)	Yes (Locally)
Functioning reactively	No	Yes	Yes	Yes (Globally)	Yes (Globally)

I: periodic update interval
n: Number of nodes in the network
SP: Shortest path
V: Number of nodes on the route reply path
ZN: Number of nodes in a zone
d: diameter of the network

6 Future Works

The proposed algorithm works on the assumed fact that each node in the system has the GPS coordinates of every other node it may need to communicate with. It should include a very effect technique which floods the GPS coordinates of each node across the ad hoc network keeping the network congestion to minimum possible levels. An effective algorithm to share the GPS coordinates among the participating node is currently under research.

The concept of tracking the moving node when the packet is already en-route from the source node is the inspiration taken from the guided missile technology which tracks the moving target by a set of predictive calculation based on its speed and sensing its heat signature. Similarly consider a situation when the packet which has already been sent from the source needs to be delivered. But if the mobile destination node has moved from its initial GPS coordinates, then the path

by which the packet should be sent should can be effectively calculated and predicted by the feed from other nodes which senses the moving node as it came near each of these nodes. A highly complex algorithm which can sense these inputs from the nodes which sensed the moving target during the course of its movement can be used to derive the equation of the trajectory of the moving target if it can be represented as a regular mathematical equation and hence the early calculation of the best path and next hop selection of nodes can be done in early stages to get the packet delivered to the destination with less hop counts during the course of movement of the target mobile node. This demands for a system wide distributed computing algorithms to be implemented with speed and accuracy in calculations.

References

[1] Oman, H.: Global Positioning System, Opportunities and Problems. IEEE AES Systems Magazine (July 1995)
[2] Wright, M., Stallings, D., Pu, D.: The Effectiveness of Global Positioning System Electronic Navigation. IEEE (March 3, 2003); ISBN 0-7803-7856
[3] Yen, Y.-S., Chang, H.-C., Chang, R.-S.: Routing with Adaptive Path and Limited Flooding for Mobile Ad Hoc Networks. IEEE, O-7803-9410-O/06/$20.OO © 2006
[4] Morshed, M.M., Rahman, M.H.: Performance Evaluation of DSDV and AODV Routing Protocols in Mobile Ad-hoc Networks
[5] Time Complexity available via www, http://en.wiktionary.org/wiki/time_complexity
[6] Ding, R., Yang, L.: A Reactive Geographic Routing Protocol for Wireless Sensor Networks. IEEE, ISBN 978-1-4244-7177-5/10/$26.00 © 2010
[7] Swapna Kumar, S., Nanda Kumar, M., Sheeba, V.S.: Implementation of Hybrid Ad-Hoc Routing Protocol. IEEE, doi:10.1109/ARTCom.2010.14, ISBN: 978-0-7695-4201-0/10 $26.00 © 2010
[8] Ajarmeh, I.A., El-Zayyat, K., Yu, J.: A Hybrid Routing Protocol for Mobile Ad hoc and Wireless Sensor Networks. IEEE, ISBN 978-1-4244-2108-4/08/$25.00 © 2008
[9] Cheng, H., Cao, J.: A design framework and taxonomy for Hybrid routing protocols in Mobile ad hoc networks. IEEE Communications Surveys & Tutorials (3rd Quarter 2008)
[10] Amouris, K.N., Papavassiliou, S., Li, M.: A Position-Based Multi-Zone Routing Protocol For Wide Area Mobile Ad-Hoc Networks. IEEE (1999), ISBN: 0-7803-5565-2/99/$10.000
[11] Pearlman, M.R.: Determining the Optimal Configuration for the Zone Routing Protocol. IEEE, ISBN: 0733–8716/99$10.00 ã 1999
[12] Sarkar, S., Bhaumik, B.B.: MEHRP: An Efficient Hybrid Routing Protocol for MANET using Direction Dependent Bordercasting
[13] Formula available via www, http://www.movable-type.co.uk/scripts/latlong.html
[14] Mbarushimana, C., Shahrabi, A.: Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks. IEEE, ISBN: 0-7695-2847-3/07 $20.00 © 2007

[15] Royer, E.M., Toh, C.-K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications (April 1999)
[16] Jiang, M., Li, J., Tay, Y.C.: Cluster Based Routing Protocol (CBRP). IETF Draft (August 1999), http://tools.ietf.org/html/draftietf-manet-cbrp-spec-01 (accessed February 21, 2008)
[17] Formula available via www, http://en.wikipedia.org/wiki/Law_of_cosines
[18] Jacquet, P., Mühlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized Link State Routing Protocol for Ad Hoc Networks. IEEE INMIC, 62–68 (2001)
[19] Pathan, A.-S.K., Alam, M.M., Monowar, M.M., Rabbi, M.F.: An Efficient Routing Protocol for Mobile Ad Hoc Networks with Neighbor Awareness and Multicasting. Proceedings of IEEE E-Tech., 97–100 (July 2004)
[20] Lim, H., Kim, C.: Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. In: Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 61–68 (2000)

Reliability Driven Soft Real-Time Fuzzy Task Scheduling in Distributed Computing Environment

P.K. Yadav[1], K. Bhatia[2], and Sagar Gulati[3]

[1] Central Building Research Institute, Roorkee (U.K.)
[2] Gurukula Kangri Vishwavidyalaya Hardwar (U.K.)
[3] Technology Education & Research Institute, Kurukshetra (Haryana)
prd_yadav@rediffmail.com, karamjitbhatia@yahoo.co.in, sagarg40@gmail.com

Abstract. All practical real-time scheduling algorithms in distributed systems present a trade-off between their computational complexity and performance. In real-time distributed systems, tasks have to be performed correctly, timely and most importantly in reliable manner. The research till date on task scheduling has primarily focused upon computation time, laxity, priority etc. Reliability of a task or task cluster is normally not considered, when it comes to task scheduling in distributed environment. Finding a reliable schedule in distributed systems with real-time constraints is shown to be NP-hard. The practical scheduling algorithms in real-time systems have not deterministic reliability. Deterministic reliable behavior is an important parameter for system robustness analysis. The intrinsic uncertainty in dynamic real-time systems increases the difficulties of scheduling problem. To alleviate these deficiencies, i have proposed a fuzzy scheduling approach to arrange real-time periodic and non-periodic tasks in systems considering reliability as the parameter for decision. In contrast, the approach balances task loads of the processors successfully while consider starvation prevention and fairness, which cause higher reliability tasks, have higher running probability. A simulation is conducted to evaluate the performance of the proposed approach. Experimental results have shown that the proposed fuzzy scheduler creates feasible schedules for homogeneous and heterogeneous tasks.

Keywords: Reliability, Failure rate, Failure density, Real-time distributed systems, Fuzzy Scheduling.

1 Introduction

Many applications namely avionics, traffic control, automated factory, and military systems require real-time communication and computation. In real-time systems, all tasks have specific parameters such as deadline, reliability, priority, etc. Modern embedded computing systems are becoming increasingly complex

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 219–226.
springerlink.com

[1]. Meanwhile, the traditional notions of best-effort and real-time processing have fractured into a spectrum of processing classes with different timeliness requirements including desktop multimedia, soft real-time, firm real-time, adaptive soft real-time, rate-based, and traditional hard real-time[2]. Many real-time systems are hard and missing deadline is catastrophic [3], whereas in soft real-time system occasional violation of deadline constraints may not result in a useless execution of the application or calamitous consequences, but decreases utilization [4]. A schedule, which is executing all real-time tasks within their deadlines and all the other constraints are met, is called a feasible schedule. Real-time scheduling can be classified in two categories, static and dynamic scheduling. A static real-time scheduling algorithm such as Rate Monotonic schedules all real-time tasks off-line using static parameters and requires complete knowledge about tasks and system parameters [5], while dynamic task scheduler calculates the feasible schedule on-line and allows tasks to be invoked dynamically. These algorithms use dynamic parameters such as deadline and laxity [6-8]. Scheduling in real-time system involves allocation of CPU and other resources to run corresponding tasks to meet certain timing constraints. Nonetheless, scheduling is more significant in real-time systems than non-real-time systems [9]. In real-time systems, tasks have to be performed correctly and in a timely fashion as well [10]. Tasks are classified as periodic and non-periodic [11]. The execution requests of a periodic task repeatedly occur at regular intervals. On the contrary, execution requests of a non-periodic task are unpredictable.

Nowadays, usage of real-time distributed systems is dramatically increasing, unfortunately, less is known about how to schedule them. Optimal scheduling of real-time tasks on distributed systems is known to be computationally intractable for large task sets. Any practical scheduling algorithm in distributed systems presents a trade-off between performance and computational complexity. The performance of a scheduling algorithm is measured in terms of additional processor required to be added at a schedule without deadline violations as compared to optimal algorithm.

In the presented paper, we focus on a real-time distributed system with different tasks reliability, failure rate and failure density attributes, and computes the performance & complexity of our proposed fuzzy scheduler with different input combinations and their effect on the system.

2 Fuzzy Inference Engine

Fuzzy logic[12], [13] is a superset of conventional Boolean logic and extends it to deal with new aspects such as partial truth and uncertainty. Fuzzy inference is the process of formulating the mapping from a given input set to an output using fuzzy logic. The basic elements of fuzzy logic are linguistic variables, fuzzy sets, and fuzzy rules [14]. The linguistic variables' values are words, specifically adjectives like "small," "little," "medium," "high," and so on. A fuzzy set is a collection of couples of elements. It generalizes the concept of a classical set,

allowing its elements to have a partial membership. The degree to which the generic element "x" belongs to the fuzzy set A (expressed by the linguistic statement x is A) is characterized by a membership function (MF), $f_A(x)$. The membership function of a fuzzy set corresponds to the indicator function of the classical sets. It can be expressed in the form of a curve that defines how each point in the input space is mapped to a membership value or a degree of truth between 0 and 1. The most common shape of a membership function is triangular, although trapezoidal and bell curves are also used. This operation normalizes all inputs to the same range and has a direct effect on system performance and accuracy.

A fuzzy set "*A"* defined within a finite interval called universe of discourse U as follows:

$$A = \{(x, f_A(x)), f_A(x): U \rightarrow [0,1]\}$$

U is the completely input range allowed for a given fuzzy linguistic variable. All fuzzy sets related to a given variable make up the term set, the set of labels within the linguistic variable described or, more properly, granulated. Fuzzy rules form the basis of fuzzy reasoning. They describe relationships among imprecise, qualitative, linguistic expressions of the system's input and output. Generally, these rules are natural language representations of human or expert knowledge and provide an easily understood knowledge representation scheme. There are two types of fuzzy inference models:

1. MAMDANI [15],
2. TSK OR SUGENO [16].

Interpreting an if-then rule involves two distinct parts: first evaluating the antecedent and then applying results to the consequent (known as implication) [17]. In the case of two-valued or binary logic, if-then rules do not present much difficulty. If the premise is true, then the conclusion is true, whereas with fuzzy approach, if the antecedent is true to some degree of membership, then the consequent is also true to that same degree.

Mamdani-type [15] inference expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. It is possible, and in many cases much more efficient, to use a single spike as the output's membership function rather than a distributed fuzzy set. This is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, Sugeno-type systems use weighted sum

of a few data points. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant.

3 The Proposed Model

As shown in Fig. 1, the major factors considered in our approach to determine the scheduling are failure density (FD), failure rate (FR) and reliability. The notion of failure density is used in the proposed approach to facilitate the number of failures during a given unit interval of time to the total number of processors in operation at the beginning.

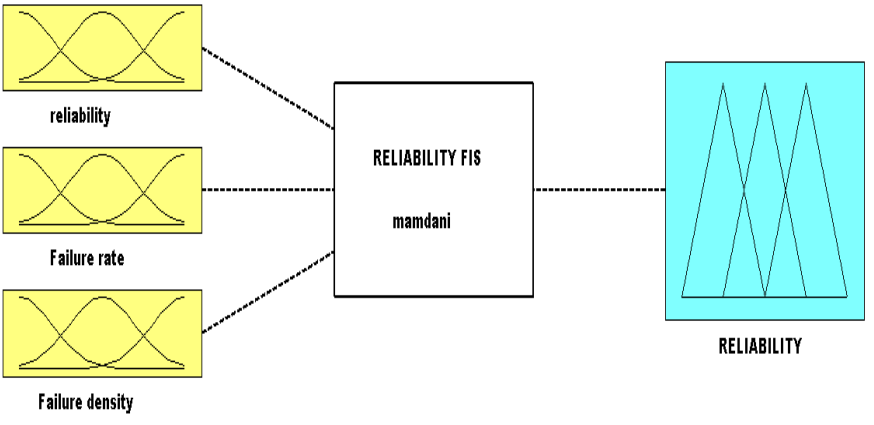


Fig. 1. Inference system Block Diagram

A task's reliability shows how reliable the task is! The inputs of these parameters are justified and represented as linguistic variables and fuzzy rules are then applied to those linguistic variables to compute the level value for deciding which task to select to schedule next.

Failure rate is another parameter, which could help in the judgment of scheduling fairness. We considered three triangular membership functions for task's reliability. "Good", "Medium" and "Poor" are these membership functions. The triangular membership functions for failure rate are "Low", "Medium" and "High", however failure density membership function considered three trapezoids. "High", "Medium" and "Low" are the name of these functions. For the $f_A(x)$ as the membership function, a large class of functions can be taken such as triangular, trapezoidal, Gaussian and bell function, however we selected trapezoidal and triangular for its usability in fuzzy dedicated hardware [18].The used membership functions for this model illustrated in Fig. 2 and 3.

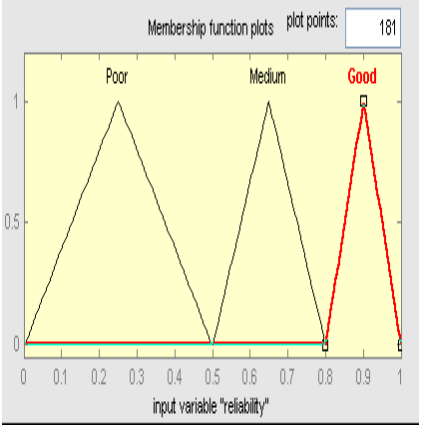


Fig. 2. Fuzzy set corresponding to Reliability

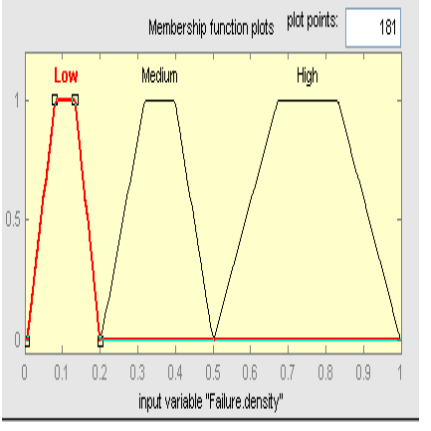


Fig. 3. Fuzzy set corresponding to Failure density

Fuzzy rules try to combine these parameters as they are connected in real worlds. Some of these rules are mentioned here:

- If (reliability is good) and (failure rate is low), then (RELIABILITY is optimal).
- If (failure rate is low) and (failure density is high), then (RELIABILITY is good).
- If (failure density is medium) and (reliability is good), then (RELIABILITY is average).

In fuzzy inference systems, the number of rules has a direct effect on its time complexity. Therefore, having fewer rules may result in a better system performance.

In our proposed algorithm as shown below, a newly arrived task will be added to the input queue. This queue consists of the remaining tasks from last cycle that has not yet been assigned.

Loop

For each processor in the distributed system, do the following:

1. *For each ready task, feed its reliability, failure rate and failure density into the inference engine. Consider the output of inference module as RELIABILITY of the task T.*

2. *Execute the task with highest RELIABILITY until a scheduling event occurs. (a running task finishes, a new task arrives)*

3. *Update the system states.*

End Loop

4 Results and Discussions

The methodology applied in the presented research gives the output as follows.

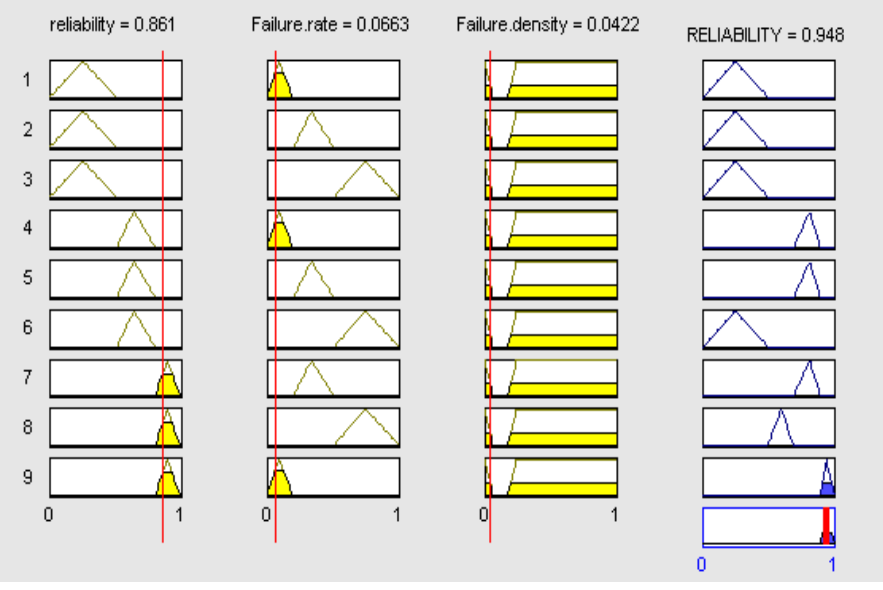


Fig. 4. Optimal Reliable solution corresponding to fuzzy sets

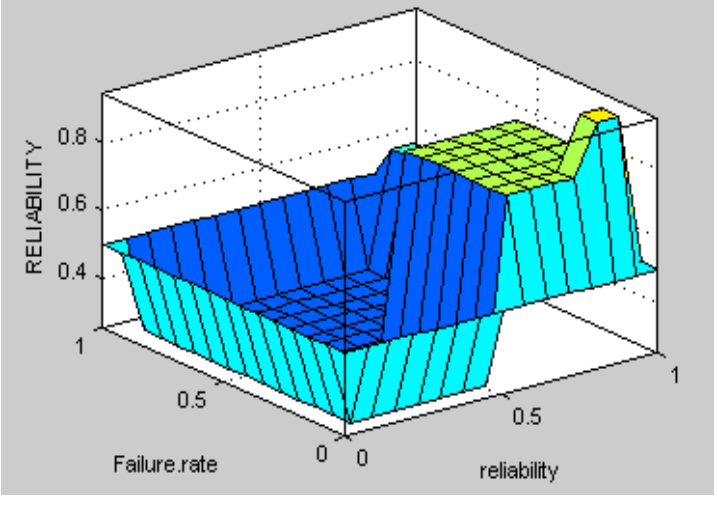


Fig. 5. The decision surface corresponding to inference rules

The proposed method is different from all the research methods presented in the past by various researchers as it is the very first time the scheduling policy has considered reliability as the parameter of judgment.

The decision surface corresponding to rules and membership functions is illustrated in Fig. 5

5 Conclusion and Future Work

The scheduler, which is proposed in this paper, has low complexity due to the simplicity of fuzzy inference engine. Consequently, its computation complexity and response time is constant and by increasing, the number of processors will not increase. This model is efficient when system has heterogeneous tasks with different constraints. In the future, I will conduct a deeper simulation and compare the results of fuzzy approach with the other algorithms.

References

[1] Gruian, F.: Energy-centric scheduling for real-time systems. Department of Computer Science. Ph.D dissertation: Lund University, p. 164 (2002)

[2] Petters, S.M.: Bounding the execution time of real-time tasks on modern processors. In: Proc. 7th Intl. Conf. Real-Time Computing Systems and Applications, Cheju Island, pp. 498–502 (2000)

[3] Taewoong, K., Heonshik, S., Naehyuck, C.: Scheduling algorithm for hard real-time communication in demand priority network. In: Proc. 10th Euromicro Workshop Real-Time Systems, Berlin, Germany, pp. 45–52 (1998)

[4] Lifeng, W., Haibin, Y.: Research on a soft real-time scheduling algorithm based on hybrid adaptive control architecture. In: Proc. American Control Conf., Lisbon, Portugal, vol. 5, pp. 4022–4027 (2003)

[5] Laplante, P.A.: The certainty of uncertainty in real-time systems. IEEE Instrum. Meas. Mag. 7, 44–50 (2004)

[6] Kreuzinger, J., Schulz, A., Pfeffer, M., Ungerer, T., Brinkschulte, U., Krakowski, C.: Real-time scheduling on multithreaded processors. In: Proc. 7th Intl. Conf. Real-Time Computing Systems and Applications, Cheju Island, South Korea, pp. 155–159 (2000)

[7] Thai, N.D.: Real-time scheduling in distributed systems. In: Proc. Intl. Conf. Parallel Computing in Electrical Engineering, Warsaw, Poland, pp. 165–170 (2002)

[8] Lin, C., Brandt, S.A.: Efficient soft real-time processing in an integrated system. In: Proc. 25th IEEE Real-Time Systems Symp. (2004)

[9] Buttazzo, G.C., Lipari, G., Caccamo, M., Abeni, L.: Elastic scheduling for flexible workload management. IEEE Trans. Comput. 51, 289–302 (2002)

[10] Lee, J., Tiao, A., Yen, J.: A fuzzy rule-based approach to real-time scheduling. In: Proc. 3rd IEEE Conf. Fuzzy Systems, IEEE World Congress Computational Intelligence, FL, vol. 2, pp. 1394–1399 (1994)

[11] Silly-Chetto, M.: Dynamic acceptance of aperiodic tasks with periodic tasks under resource sharing constraints. IEE Proc. Software 146, 120–127 (1999)

[12] Zadeh, L.A.: Fuzzy sets versus probability. Proc. IEEE 68, 421 (1980)
[13] Zadeh, L.A.: Fuzzy logic, neural networks, and soft computing. Commun. ACM 37, 77–84 (1994)
[14] Pedrycz, W., Gomide, F.: An introduction to fuzzy sets: analysis and design. The MIT Press (1998)
[15] Mamdani, E.H.: Application of fuzzy algorithms for the control of a dynamic plant. Proc. IEE 121, 1585–1588 (1974)
[16] Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst., Man, Cybern. 15, 116–132 (1985)
[17] Ascia, G., Catania, V.: A general purpose processor oriented to fuzzy reasoning. In: Proc. 10th IEEE International Conf. Fuzzy Systems, Melbourne, Australia, pp. 352–355 (2001)
[18] Youngdal, K., Hyung, L.-K.: An architecture of fuzzy logic controller with parallel defuzzification. In: Proc. Biennial Conf. of the North American Fuzzy Information Processing Society, Berkeley, CA, pp. 497–501 (1996)

Bacterial Foraging Optimization: A Survey

Vivek Agrawal, Harish Sharma, and Jagdish Chand Bansal

ABV-Indian Institute of Information Technology and Management, Gwalior
{vivek.ag86,jcbansal}@gmail.com,harish0107@rediffmail.com

Abstract. Often real world provides some complex optimization problems that can not be easily dealt with available mathematical optimization methods. If the user is not very conscious about the exact solution of the problem in hand then intelligence emerged from social behavior of social colony members may be used to solve these kind of problems. Based on this concept, Passino proposed an optimization technique known as the bacterial foraging optimization algorithm (BFOA). The foraging behavior of bacteria produces an intelligent social behavior, called as swarm intelligence. Social foraging behavior of Escherichia coli is studied by researchers and developed a new algorithm named Bacterial foraging optimization algorithm (BFOA). BFOA is a widely accepted optimization algorithm and currently it is a growing field of research for distributed optimization and control. Since its inception, a lot of research has been carried out to make BFOA more and more efficient and to apply BFOA for different types of problems. This paper presents a review of BFOA modifications and it application areas.

Keywords: Optimization, Escherichia coli Bacteria, Bacterial Foraging Optimization Algorithm, Swarm intelligence, Meta-heuristics.

1 Introduction

Nature inspired algorithms are based on social behaviour of various social insects to find near-optimal solutions to the difficult optimization problems. The common thing of all the population-based algorithms is that the possible solutions are modified by using the fitness of the objective function value. Therefore, the population is moved towards better solution areas of the search space. Two important classes of population-based optimization algorithms are evolutionary algorithms (11) and swarm intelligence-based algorithms (10). Swarm Intelligence is a meta-heuristic method in the field of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 227–242.
springerlink.com

artificial intelligence that is used to solve optimization problems. It is based on the collective behavior of social insects, flocks of birds, or schools of fish. However, a swarm can be considered as any collection of interacting agents or individuals. These animals can solve complex tasks without centralized control. Researchers have analyzed such behaviors and designed algorithms that can be used to solve combinatorial and numerical optimization problems in many science and engineering domains. Previous research (42; 16; 32) has shown that algorithms based on Swarm Intelligence have great potential. The algorithms that have emerged in recent years include Ant Colony Optimization (ACO) (42) based on the foraging behavior of ants, and Particle Swarm Optimization (PSO) (16) based on the behaviors of bird flocks and fish schools.

Bacteria Foraging Optimization Algorithm (BFOA), proposed by Passino (30), is a new algorithm in the category of swarm intelligence. The group foraging behaviour of E.coli bacteria in multi-optimal function optimization is the basis of the new algorithm. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. The communication of Individual bacteria is done by sending various signals. The foraging decisions of a bacteria are based on two factors one is called chemotaxis and another is mimicking chemotactic movement. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space. After invention of the BFOA, it has been applied to various Engineering and Science related optimization problems. Researchers are continuously modifying the BFOA by hybridization, inserting new phase and new control parameters to improve the performance. This algorithm has been already compared with GA, PSO and ABC algorithms for solving real world optimization problems.

Fig. 1 shows the top ten countries which are working on BFO. Fig. 2 shows the year wise publication details on BFO. The graph is plotted in

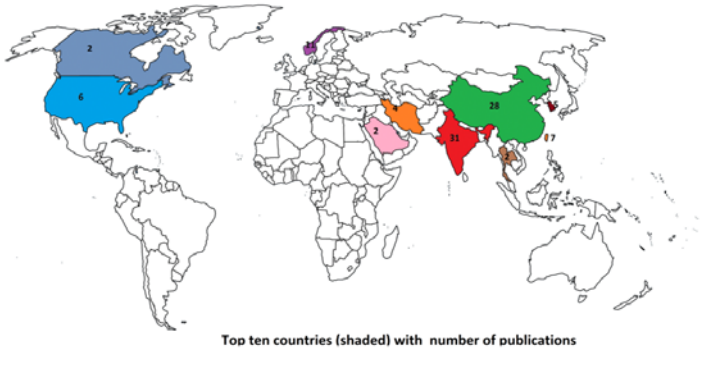


Fig. 1. World map showing BFO publication countries. (Source: *Scopus* Dated: 18 Sept. 2011)

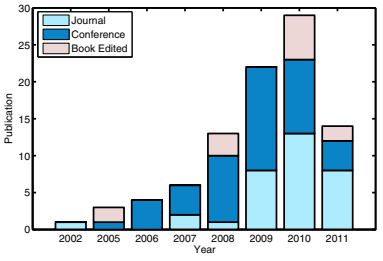


Fig. 2. Graph showing year wise publication details of BFO. (Source: *Scopus* Dated: 18 Sept. 2011)

stack way in which first part of stack shows the publication in journal,second part shows the publication in conference and third part shows publication in edited books. It is clear from this figure that in last three years publications of BFO research are highly increased. It shows the research on BFO is a very liked area now a days. Rest of the paper is organized as follows: Section 2 describes brief overview of behavior of bacteria foraging. In section 3, Bacterial Foraging Algorithm is explained. Some basic Improvements on Bacteria Foraging Optimization Algorithm are briefly reviewed in section 4. In section 5, applications of BFO is described. Finally, in section 6, paper is concluded.

2 Behavior of Bacteria

Escherichia coli is a single celled organism that lives in our gut. It is equipped with a set of rotary motors and diameter is 45 nm. Each motor drives a thin, helical, long filament. This filament extends several cell body lengths out into the external medium. The assemblage of motor and filament is named a flagellum. The motion of several flagella facilitates a cell to swim. The two basic operations performed by a bacterium at the time of foraging ([8]; [2]) are tumble or swim, which are performed with the help of flagella. When the flagella is rotated in the clockwise direction, each flagellum pulls on the cell, so each flagella moves independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place in search of nutrient gradient it tumbles frequently. When the flagella is rotated in the counterclockwise direction the bacterium swims at a very fast rate. In the above-mentioned algorithm bacteria search for nutrients in a manner to maximize energy, and the process in which bacteria moves by taking small steps while searching for nutrients is called CHEMOTAXIS. Bacteria also communicates with others by sending signals. In a friendly environment generally the bacteria moves for a longer distance. Fig. 3 depicts how clockwise and counter clockwise movement of a bacterium take place in a nutrient solution. E.coli bacteria is cylindrical organism with hemispherical endcaps. The cell which weighs only 1 picogram , is about 70% water. Some strains are flagellated while

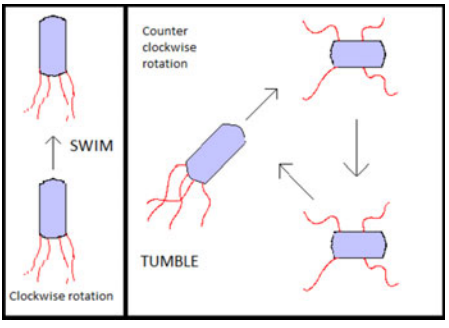


Fig. 3. Swim and tumble of a bacterium ([8])

others are not. When E. coli grows, it first gets longer in length and then divides himself from the middle. In sense it is immortal, because the mother cell is replaced by two daughters, essentially identical to the daughters of the previous generation.If well fed and held at the temperature of the human gut (37^oC), E. coli can replicate and synthesize everything it needs to make a new replica of itself in about 20-22 minutes. Thus, if we start at noon today with one cell (and lots of food), by noon tomorrow there will be $2^{72} = 4.7 \times 1021$ cells enough to pack a cube 17 meters on a side. Researchers analyzed the social behaviour of bacteria and simulates it to develop a new nature inspired algorithm named Bacteria Foraging Optimization algorithm. Following are the basic steps of bacteria foraging that are used in simulating the algorithm.

Chemotaxis: This simulates the movement of E.coli through swimming and tumbling via flagella.The bacteria alternate between these two modes of operation for the entire lifetime. Movement of bacteria is represented by $\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$.

Swarming: A group of E.coli cells arrange themselves in a riding ring by moving up the nutrient gradient. The cells when stimulated by high level of succinate, release an attractant aspertate, which helps them to aggregate into groups and thus move as concentric patterns of swarm.

$$
\begin{aligned}
J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}(\theta, \theta^i(j,k,l)) \\
&= \sum_{i=1}^{S}[-d_{attractant}\exp(-w_{attractant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \\
&\quad + \sum_{i=1}^{S}[h_{repellant}\exp(-w_{repellant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)]
\end{aligned}
$$

where $\theta = [\theta_1, \theta_2, ...\theta_p]^T$ is a point in the p-dimensional search domain. $d_{attractant}, w_{attractant}, h_{repellant}, w_{repellant}$ are different coefficients that should be chosen properly.

Reproduction: If well fed and held at a temperature of human body(37 degree celsius), E.coli can replicate and synthesize everythingto make a new clone of itself in about 20-22 minutes. Starting with one cell, in 24 hours there will be 2^{72} cells. Which is enough to pack a cube 17 metres on side.

Elimination and dispersal: If there is a sudden rise in temperature then all the bacteria in a region are killed or a group is dispersed in a new location.

3 Bacteria Foraging Optimization Algorithm

BFOA is based on the four principal mechanisms observed in a real bacterial system: chemotaxis, swarming, reproduction, and elimination-dispersal. In this section, a pseudo-code for BFOA is explained step by step.

Following are the various abbreviation used in the pseudo-code.

j: index for the chemotactic step,
k: index for the reproduction step,
l: index of the elimination-dispersal event,
p: Dimension of the search space,
S: Total number of bacteria in the population,
N_c : The number of chemotactic steps,
N_s: The swimming length,
N_{re} : The number of reproduction steps,
N_{ed} : The number of elimination-dispersal events,
P_{ed} : Elimination-dispersal probability,
$C_(i)$: The size of the step taken in the random direction specified by the tumble.
$P(j,k,l) = \theta^i(j,k,l)|i = 1,2,...,S$ represent the position of each member in the population of the S bacteria at the j^{th} chemotactic step, k^{th} reproduction step, and l^{th} elimination-dispersal event.
$J(i,j,k,l)$ denote the cost at the location of the i^{th} bacterium $\theta^i(j,k,l) \in \Re^p$.

BFOA ALGORITHM(8)
[1] Initialize parameters $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i), (i = 1,2,...S), \theta^i$.
[2] Elimination-dispersal loop: $l = l + 1$
[3] Reproduction loop: $k = k + 1$
[4] Chemotaxis loop: $j = j + 1$
(a) For $i = 1,2,...S$
take a chemotactic step for bacterium i as follows.
(b) Compute fitness function, $J(i,j,k,l)$.

Let, $J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$ (i.e. add on the cell-to cell attractant - repellant profile to simulate the swarming behavior).
(c) Let $J_{last} = J(i,j,k,l)$ to save this value since we may find a better cost via a run.
(d) Tumble: generate a random vector $\triangle(i) \in \Re^p$ with each element $\triangle_m(i)$, $m = 1, 2, ...p$, a random number on [-1, 1].
(e) Move: Let $\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$.
This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.
(f) Compute $J(i,j+1,k,l)$ and let

$$J(i,j+1,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$$

(g) Swim
Let $m = 0$ (counter for swim length).
while $(m < N_s)$ (if have not climbed down too long). **do**
  Let$m = m + 1$.
  if $J(i,j+1,k,l) < J_{last}$(if doing better) **then**
    let $J_{last} = J(i,j+1,k,l)$ and let
    $\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$
    And use this $\theta^i(j+1,k,l)$ to compute the new $J(i,j+1,k,l)$ as we did in [f]
  else
    let $m = N_s$
  end if
end while
(h) Go to next bacterium $(i+1)$ if $i \neq S$ (i.e., go to [b] to process the next bacterium).
[5] IF$j < N_c$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.
[6] Reproduction:

(a) For the given k and l, and for each $i = 1, 2, ..., S$, let
  $J^i_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l)$
  be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health).
(b) The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[7] If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

[8] Elimination-dispersal:
for $i = 1, 2..., S$ with probability P_{ed} **do**
Eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain.
if $l < N_{ed}$ **then**
go to step 2;
else
end.
end if
end for

Flow chart of the BFOA algorithm is shown in Fig. 4. This flow diagram presents working of BFOA step by step.

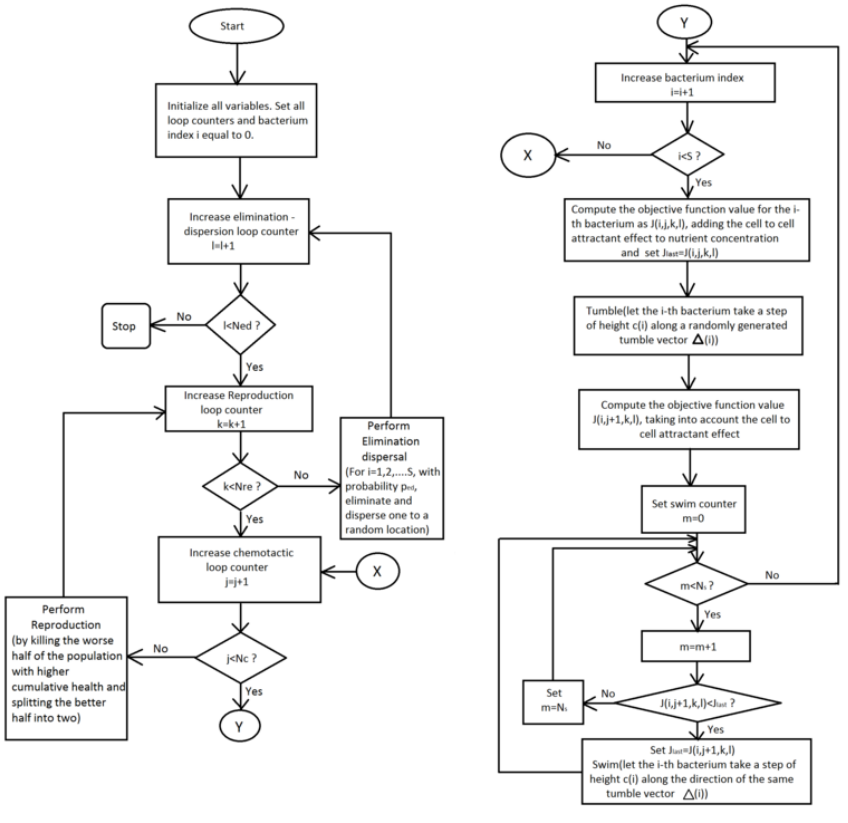


Fig. 4. Flow chart of BFO Algorithm (8)

4 Brief Review of Modifications in BFO

Often real world provides some complex optimization problems that can not be easily dealt with available mathematical optimization methods. If the user is not very conscious about the exact solution of the problem in hand then intelligence emerged from social behavior of social colony members may be used to solve these kind of problems. Bacteria are in the category of social insects. The foraging behavior of bacteria produces an intelligent social behavior, called as swarm intelligence. This swarm intelligence is simulated and an intelligent search algorithm namely, Bacteria Foraging Optimization (BFO) algorithm. Since its inception, a lot of research has been carried out to make BFO more and more efficient and to apply BFO for different types of problems. In order to get rid of the drawbacks of basic BFO, researchers have improved BFO in many ways. Following subsections contains a brief review of modification on BFOA.

4.1 Hybridization of BFO with Other Population Based Algorithms

In 2010 Panikhom, S. et al. (29) hybridized Tabu search with BFO and applied it to analyze stability of linear and nonlinear systems. In 2010 Manuaba, I.B.G. et al. (25) used combination of bacterial foraging technique with Particle swarm optimization. The proposed method was applied to PID controller tuning and was compared to another method. In same year Praveena, P. et al. (31) combined the Particle swarm optimization and varying chemotactic step-size bacterial foraging optimization algorithms for dynamic economic dispatch with non-smooth fuel cost functions. Further Liu, X. et al. (41) generated a bacterial foraging global optimization algorithm based on the particle swarm optimization. The aim of new hybrid algorithm was to improve the efficiency, accuracy and overcome the drawbacks of weak ability to perceive the environment and vulnerable to perception of local extreme in the optimization process of bacterial foraging optimization (BFO) algorithm. In 2010 Zang, T. et al. (43) hybridized Bacterial foraging optimization algorithm with particle swarm optimization to generate distribution network reconfiguration. Distribution network reconfiguration for loss minimization is a complex, large-scale combinatorial optimization problem. In 2010 Hooshmand, R.A. and Ezatabadipour, M. used bacterial foraging oriented by particle swarm optimization algorithm Corrective action planning considering FACTS allocation and optimal load shedding.

In 2009 Shao, L. and Chen, Y. (35) hybridized Bacterial foraging optimization algorithm and tabu search for motif discovery. Extracting motifs in the sea of DNA sequences is an intricate task but have great significance. In the

same year Bakwad, K.M. et al. (1) cascaded Bacterial foraging optimization technique with adaptive filter to enhance peak signal to noise ratio from single image. They proposed a new approach to enhance peak signal to noise ratio of highly corrupted image affected by impulse noise. The proposed technique is implemented using an adaptive median filter and the bacterial foraging optimization (BFO) technique. In 2009 Olesen, J.R. et al. (27) used particle swarm optimization and bacterial foraging for Auto-clustering. Further, Kou, P.G. et al. (28) generated Optimal PID governor tuning of hydraulic turbine generators with bacterial foraging particle swarm optimization algorithm. To improve the quality of PID parameters of the turbine governor, bacterial foraging optimization (BFO) algorithm was introduced. Considering the slow convergence of BFO algorithm and the good convergence of particle swarm optimization (PSO) algorithm, a novel method named BFO- PSO algorithm was proposed. In 2009 Majhi, R. et al. (24) hybridized adaptive bacterial foraging optimization (ABFO) and BFO based techniques for Efficient prediction of stock market indices. The structure used in these forecasting models is a simple linear combiner. The connecting weights of the adaptive linear combiner based models were optimized using ABFO and BFO by minimizing its mean square error (MSE).

In 2007 Kim, D.H. et al. (19) hybridized genetic algorithm and bacterial foraging approach for global optimization. It used four test functions and the performance of the algorithm was studied with an emphasis on mutation, crossover, variation of step sizes, chemotactic steps, and the lifetime of the bacteria. The proposed algorithm was then used to tune a PID controller of an automatic voltage regulator (AVR). Simulation results clearly illustrated that the proposed approach was very efficient and could easily be extended for other global optimization problems. In the same year Kim, D.H. et al. (17) hybridized genetic algorithm and bacterial foraging approach for robust tuning of PID controller with disturbance rejection. To design disturbance rejection tuning, disturbance rejection conditions based on H^{∞} are illustrated and the performance of response is computed for the designed PID controller as the integral of time weighted squared error.

4.2 Inserting New Control Parameters/Phase in BFO

Yanhai Chen and Weixing Lin (6) proposed modification in the chemotactic step $C(i)$ of BFOA. In BFOA, the size of the chemotactic step $C(i)$ has remained constant, although it is a random number for each bacterium when initialize the population. At the early stage of BFO algorithm, in order to get close to the optima as soon as possible,it takes sufficiently large step, and slows down to find global optima more precisely at the later period of algorithm. Usually, the former way is course search and the latter is considered as elaborate search which can acquire a much better effect in both

convergence rate and precision. Hence, an adaptive step-size method is proposed described as follows:

$$C(i) = C_{max}(i) - \frac{C_{max}(i) - C_{min}(i)}{N_c} \cdot j$$

Where $C_{max}(i)$ and $C_{min}(i)$ are maximum and minimum of the step size of the i^{th} bacterium, respectively. N_c denotes the lifetime of bacterium, namely the total chemotaxis steps, and j is index of current chemotaxis step. They also proposed modification in the algorithmsś search scope. It is well known that algorithm efficiency is affected by search scope. They introduced a method to narrow the search scope dynamically in order to provide a large range at the early stage of the algorithm and decreasing scope at the late stage of the algorithm, which will help to speed up the convergence and improve the algorithm accuracy. Concrete work is as follows:

$$\theta_{max}(j) = \theta_{gbest} + \frac{R}{2^j}$$

$$\theta_{min}(j) = \theta_{gbest} - \frac{R}{2^j}$$

where $[\theta_{min}(j), \theta_{max}(j)]$ is the current searching scope at the j^{th} chemotaxis step, and R is the sphere of activity of the bacteria swarm. After the updation of θ_{gbest} and chemotaxis step j, the searching scope decreases gradually so that there is coarse search at the early period of the algorithm and the fine search at the late phase.

Hanning et al. (4) introduced Self-Adaptation in Bacterial Foraging Optimization (SA - BFO) Algorithm. In the SA-BFO algorithm, they introduced an individual run-length unit to the i^{th} bacterium of the colony and each bacterium by using the current status of its own can only modify the search behavior of itself . In this way, the position as well as the run-length unit of each bacterium undergoes evolution. In SA-BFO evolution process, each bacterium displays alternatively two distinct search states:

(1) **Exploration state**, during which the bacterium employs a large run-length unit to explore the previously unscanned regions in the search space as fast as possible.
(2) **Exploitation state**, in this state the bacterium uses a small movement-length unit to exploit the promising regions slowly in nearby locations.

Each bacterium in the colony permanently maintains an appropriate balance between Exploration and Exploitation states by varying its own run-length unit adaptively. This balance is achieved with the help of two decision

indicators: first is fitness improvement and second is no improvement registered lately.

The criteria that determine the adjustment of individual run-length unit and the entrance into one of the states are the following:

Criterion-1: if a new promising domain is discovered by the bacterium , the run-length unit of this bacterium is adapted to another smaller one. Here discovers a new promising domain means this bacterium register a fitness improvement from the last generation to the current beyond a certain precision . Following Criterion-1, the bacteriums behavior will self-adapt into Exploitation state.

Criterion-2: if the bacteriums current fitness is unchanged for some consecutive generations then this bacteriums run-length unit is augment and it enters into Exploration state. This situation means that the bacterium searches on an un-promising domain or the domain has nothing new where this bacterium can focuses its search.

In 2006, (9) dos Santos Coelho, L. and da Costa Silveira, C. modified BFOA for present speed variable using uniform, Gaussian, and Cauchy probability distribution functions for the movement of bacteria.

In 2009, Chen, H. et al. (5) presented a variation on the original BFO algorithm, namely the Cooperative Bacterial Foraging Optimization (CBFO), which significantly improve the original BFO in solving complex optimization problems. This significant improvement was achieved by applying two cooperative approaches to the original BFO, namely the serial heterogeneous cooperation on the implicit space decomposition level and the serial heterogeneous cooperation on the hybrid space decomposition level. In 2010, Niu, B. et al. (26) introduced a linear variation of chemotaxis step in the basic BFO for finding the optimal portfolios.

5 Applications of BFO

BFO is a newly developed algorithm in the field of Bio-Inspired computing. It has been successfully applied in the fields of computers, science, maths as well as medicine. Fig. 5 shows various fields categorize by subject on which ABC has been successfully applied. In this Figure CS stands for Computer Science, ENG for Engineering, MT for Mathematics, DS for Decision Sciences, BGM for Biochemistry Genetics and Molecular Biology, PA for Physics and Astronomy, MS for Materials Science, MD for Medicine, SS for Social Sciences and CM for Chemistry. Table 1 shows a brief review of BFOA application in various filed of engineering optimization problems.

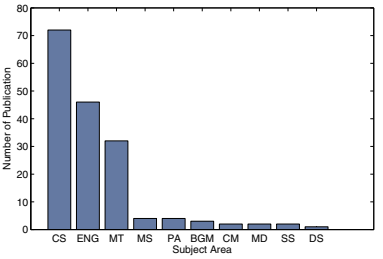


Fig. 5. Graph showing subject area wise publication of BFO. (Source: Scopus Dated: 18 Sept. 2011)

Table 1. Summary of Applications of BFO to Engineering Optimization Problems

Year	Application and Reference	Year	Application and Reference
Computer Engineering Optimization Problems			
2011	Bus scheduling(12)	2011	Image segmentation (34)
2011	Data clustering (39)	2010	The Cooperative Optimization (15)
Electrical Engineering Optimization Problems			
2010	Global function Optimization (41)	2010	Fuzzy PID controller design using self adaptive bacterial foraging optimization (36)
2010	Constrained optimization (40)	2010	Neural network for short-term load forecasting (43)
2009	Fractional order PID (22)	2009	Auto-clustering (27)
2009	Prediction of stock market (24)	2009	Non-linear model identification (21)
2008	Economic load dispatch (33)	2007	Robotic manipulator system (9)
2006	Adaptive channel equalizer (23)	2005	Robust tuning (18)
2005	Adaptive tuning of PID controller (20)		
Electrical and Electronics Engineering Optimization Problems			
2009	Dynamic economic dispatch (38)	2009	Motif discovery (35)
2009	Enhance peak signal (1)	2008	Economic load dispatch (34)
2008	Resonant frequency (13)		
Miscellaneous Optimization Problems			
2011	Placement of capacitors and dispersed generators (14)	2010	Dynamic economic dispatch (31)
2010	Adaptive Channel Equalizer (37)	2010	Second-generation current conveyors (3)
2010	Distribution network reconfiguration (43)	2008	Option model calibration (7)

6 Conclusion

This paper attempted to provide an overall survey of Bacterial Foraging Optimization algorithm and its variants. First, the social behaviour of Bacteria is explained step by step and next the simulation of behaviour for Bacterial Foraging Optimization algorithms is shown. Next it provided an extensive review of the modifications of Bacterial Foraging Optimization based on hybridization, fine tuning of control parameters and induction of new phase/control parameter. After it, a brief overview of various most significant engineering and science applications of Bacterial Foraging Optimization and its modified variants are tabulated. The content of the paper indicates the fact that Bacterial Foraging Optimization algorithm will continue to remain a vibrant and active field of multi-disciplinary research in the years to come.

Acknowledgement. First and Second author acknowledge ABV- Indian Institute of Information Technology and Management Gwalior, India for providing research grant to carry out this work.

References

[1] Bakwad, K.M., Pattnaik, S.S., Sohi, B.S., Devi, S., Panigrahi, B.K., Gollapudi, S.V.: Bacterial foraging optimization technique cascaded with adaptive filter to enhance peak signal to noise ratio from single image. IETE Journal of Research 55(4), 173 (2009)
[2] Berg, H.C.: Motile behavior of bacteria. Physics Today 53(1), 24–30 (2000)
[3] Chatterjee, A., Fakhfakh, M., Siarry, P.: Design of second-generation current conveyors employing bacterial foraging optimization. Microelectronics Journal (2010)
[4] Chen, H., Zhu, Y., Hu, K.: Self-adaptation in bacterial foraging optimization algorithm, vol. 1, pp. 1026–1031 (2008)
[5] Chen, H., Zhu, Y., Hu, K.: Cooperative bacterial foraging optimization. Discrete Dynamics in Nature and Society, 1–17 (2009)
[6] Chen, Y., Lin, W.: An improved bacterial foraging optimization, pp. 2057–2062 (2009)
[7] Dang, J., Brabazon, A., O'Neill, M., Edelman, D.: Option model calibration using a bacterial foraging optimization algorithm, pp. 113–122 (2008)
[8] Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. Foundations of Computational Intelligence 3, 23–55 (2009)
[9] dos Santos Coelho, L., da Costa Silveira, C.: Improved bacterial foraging strategy for controller optimization applied to robotic manipulator system. In: 2006 IEEE Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, pp. 1276–1281. IEEE (2006)
[10] Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm intelligence. Elsevier (2001)
[11] Fogel, D.B.: Introduction to evolutionary computation. Evolutionary Computation: Basic Algorithms and Operators 1, 1 (2000)

[12] Pang, H.-L., Wang, D.-W., Gao, Z.-W.: Adaptive bacterial foraging optimization and its application for bus scheduling. Journal of System Simulation 23(6), 1151–1155 (2011)
[13] Gollapudi, S.V.R.S., Pattnaik, S.S., Bajpai, O.P., Devi, S., Vidya Sagar, C., Pradyumna, P.K., Bakwad, K.M.: Bacterial foraging optimization technique to calculate resonant frequency of rectangular microstrip antenna. International Journal of RF and Microwave Computer-Aided Engineering 18(4), 383–388 (2008)
[14] Hooshmand, R.A., Mohkami, H.: New optimal placement of capacitors and dispersed generators using bacterial foraging oriented by particle swarm optimization algorithm in distribution systems. Electrical Engineering (Archiv fur Elektrotechnik), 1–11
[15] Hui, C., Yang, L.: Cbfo: The cooperative optimization of bacterial foraging. In: 2010 International Conference on Computer Application and System Modeling (ICCASM), vol. 2, pp. V2-106–V2-109. IEEE (2010)
[16] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
[17] Abraham, A., Kim, D.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization and robust tuning of pid controller. SCI, vol. 75, pp. 171–199 (2007)
[18] Kim, D.H.: Robust Tuning of Embedded Intelligent PID Controller for Induction Motor Using Bacterial Foraging Based Optimization. In: Wu, Z., Chen, C., Guo, M., Bu, J. (eds.) ICESS 2004. LNCS, vol. 3605, pp. 137–142. Springer, Heidelberg (2005)
[19] Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences 177(18), 3918–3937 (2007)
[20] Kim, D.H., Cho, J.H.: Adaptive Tuning of PID Controller for Multivariable System Using Bacterial Foraging Based Optimization. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 231–235. Springer, Heidelberg (2005)
[21] Lin, W.X., Liu, P.X., Li, W.L., Chen, Y.H., Ou, C.: Application of bacterial foraging optimization in a non-linear model identification. Journal of System Simulation 24(10), 3100–3104 (2009)
[22] Luo, Y., Li, J.: The controlling parameters tuning and its application of fractional order pid bacterial foraging-based oriented by particle swarm optimization, vol. 1, pp. 4–7 (2009)
[23] Majhi, B., Panda, G., Choubey, A.: On the development of a new adaptive channel equalizer using bacterial foraging optimization technique. In: 2006 Annual IEEE India Conference, pp. 1–6. IEEE (2006)
[24] Majhi, R., Panda, G., Majhi, B., Sahoo, G.: Efficient prediction of stock market indices using adaptive bacterial foraging optimization (abfo) and bfo based techniques. Expert Systems with Applications 36(6), 10097–10104 (2009)
[25] Abdillah, M., Soeprijanto, A., Mauridhi, H.P., Manuaba, I.B.G.: Coordination of pid based power system stabilizer and avr using combination bacterial foraging techique particle swarm optimization. In: 4th International Conference on Modeling, Simulation and Applied Optimization (2011)

[26] Niu, B., Xiao, H., Tan, L., Li, L., Rao, J.: Modified Bacterial Foraging Optimizer for Liquidity Risk Portfolio Optimization. In: Li, K., Li, X., Ma, S., Irwin, G.W. (eds.) LSMS 2010. Communications in Computer and Information Science, vol. 98, pp. 16–22. Springer, Heidelberg (2010)
[27] Olesen, J.R., Hernandez, J.C., Zeng, Y.: Auto-Clustering Using Particle Swarm Optimization and Bacterial Foraging. In: Cao, L., Gorodetsky, V., Liu, J., Weiss, G., Yu, P.S. (eds.) ADMI 2009. LNCS, vol. 5680, pp. 69–83. Springer, Heidelberg (2009)
[28] Kou, P.-G., Zhou, J.-Z., He, Y.-Y., Xiang, X.-Q., Li, C.-S.: Optimal pid governor tuning of hydraulic turbine generators with bacterial foraging particle swarm optimization algorithm. In: Proceedings of the CSEE, vol. 26 (2009)
[29] Sarasiri, N., Sujitjorn, S., Panikhom, S.: Hybrid bacterial foraging and tabu search optimization (btso) algorithms for lyapunov's stability analysis of nonlinear systems. International Journal of Mathematics and Computers in Simulation 4(3), 81–89 (2010)
[30] Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002)
[31] Praveena, P., Vaisakh, K., Rama Mohana Rao, S.: Particle Swarm Optimization and Varying Chemotactic Step-Size Bacterial Foraging Optimization Algorithms Based Dynamic Economic Dispatch with Non-Smooth Fuel Cost Functions. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 727–738. Springer, Heidelberg (2010)
[32] Price, K.V., Storn, R.M., Lampinen, J.: Differential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)
[33] Saber, A.Y., Venayagamoorthy, G.K.: Economic load dispatch using bacterial foraging technique with particle swarm optimization biased evolution. In: Swarm Intelligence Symposium, SIS 2008, pp. 1–8. IEEE (2008)
[34] Sathya, P.D., Kayalvizhi, R.: Image segmentation using minimum cross entropy and bacterial foraging optimization algorithm
[35] Shao, L., Chen, Y.: Bacterial foraging optimization algorithm integrating tabu search for motif discovery. In: 2009 IEEE International Conference on Bioinformatics and Biomedicine, pp. 415–418. IEEE (2009)
[36] Su, T.J., Chen, L.W., Yu, C.J., Cheng, J.C.: Fuzzy pid controller design using self adaptive bacterial foraging optimization. In: Proceedings of SICE Annual Conference 2010, pp. 2604–2607. IEEE (2010)
[37] Su, T.J., Cheng, J.C., Yu, C.J.: An adaptive channel equalizer using self-adaptation bacterial foraging optimization. Optics Communications 283(20), 3911–3916 (2010)
[38] Vaisakh, K., Praveena, P., Rao, S.R.M.: Pso-dv and bacterial foraging optimization based dynamic economic dispatch with non-smooth cost functions. In: 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 135–139. IEEE (2009)
[39] Wan, M., Li, L., Xiao, J., Wang, C., Yang, Y.: Data clustering using bacterial foraging optimization. Journal of Intelligent Information Systems, 1–21
[40] Wang, Q., Gao, X.Z., Wang, C.: An adaptive bacterial foraging algorithm for constrained optimization
[41] XiaoLong, L., RongJun, L., Ping, Y.: A bacterial foraging global optimization algorithm based on the particle swarm optimization, vol. 2, pp. 22–27

[42] Yagmahan, B., Yenisey, M.M.: Ant colony optimization for multi-objective flow shop scheduling problem. Computers & Industrial Engineering 54(3), 411–420 (2008)

[43] Zang, T., He, Z., Ye, D.: Bacterial Foraging Optimization Algorithm with Particle Swarm Optimization Strategy for Distribution Network Reconfiguration. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 365–372. Springer, Heidelberg (2010)

Fuzzy Logic Controller and Neural Network Controller as a Power System Regulator Implemented on GUI

Parita D. Giri[1] and Satish K. Shah[2]

[1] Electrical Engg. Dept., S.V.I.T., Vasad, Gujarat, India
[2] Electrical Engg. Dept., Faculty of Technology & Engineering, M.S.U., Baroda, Gujarat, India,
{bittu_goswami,satishkshah_2005}@yahoo.com

Abstract. This paper presents design of Fuzzy Logic Controller (FLC) and Neural Network Controller (NNC) as a Regulator for effective voltage control over a Simple and a Stabilized regulator in order to maintain stability and enhance the closed-loop performance of a power system using a fast computing user friendly Graphical User Interface(GUI). The gains and tuning parameters are kept almost same in Simple, Stabilized, Fuzzy Logic and Neural Network Regulator. The step responses are interfaced on a common GUI page, the performance of Fuzzy Logic Regulator in comparison to the conventional fixed gain regulators proves better but Neural Network Controller has the best results.

Keywords: Fuzzy Logic, Neural Network, Voltage Regulator, Steady state error, GUI.

1 Introduction

In the power system, the excitation system maintains the effective voltage control for reactive power flow and enhancement of the system stability. It must be able to respond quickly to a disturbance enhancing the transient stability and the small signal stability. Three principal control systems directly affect a synchronous generator: the amplifier, exciter and generator control. The voltage regulator is the intelligence of the system and controls the output of the exciter so that the generated voltage and reactive power change in the desired way. In most modern systems the automatic voltage regulator is to hold the terminal voltage magnitude of a synchronous generator at a specified level. Low –frequency oscillations are a common problem in large power systems. A power system regulator can provide a supplementary control signal to the excitation system and/or the speed governor system of the electric generating unit to damp these oscillations. Because power systems are highly nonlinear systems, with configurations and parameters that

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 243–256.
springerlink.com

change with time, the power system stabilizer design based on the linearized model of the power systems cannot guarantee its performance in a practical operating environment. To improve the performance of a nonlinear system numerous techniques have been proposed for their design, such as genetic algorithms, neural networks, fuzzy and many other nonlinear control techniques. The application of fuzzy logic control appears to be most suitable one whenever a well-defined control objective cannot be specified, the system to be controlled is a complex one, or its exact mathematical model is not available. In this paper the fuzzy logic regulator simulation results clearly demonstrate the superiority of the Fuzzy Logic Regulator in comparison to the Simple Regulator and the Stabilizer. But neural network proves good even when the exact input-output relations are not known, here best results are obtained with the Neural Network Regulator.

2 Model of Power System Selected for Analysis

***Amplifier model*:** The amplifier model is represented by a gain K_A and a time constant T_A ; the transfer function is,

$$\frac{V_R}{V_E} = \frac{K_A}{1 + T_A S} \tag{1}$$

Typical values of K_A are in the range of 10 to 400. The Amplifier time constant T_A is very small ranging from 0.02 to 0.1 s.

***Exciter model*:** The transfer function of a modern exciter may be represented by a gain K_E and a single time constant T_E

$$\frac{V_F}{V_R} = \frac{K_E}{1 + T_E S} \tag{2}$$

Typical values of KE are in the range of 10 to 400. The time constant T_E is in the range of 0.5 to 1.0 s.

Generator model: In the linearized model, the transfer function relating the generator terminal voltage field voltage can be represented by a gain K_G and a time constant T_G.

$$\frac{V_t}{V_F} = \frac{K_G}{1 + T_G S} \tag{3}$$

Sensor/Stabilizer model: The sensor is modelled by a simple first-order transfer function, given by

$$\frac{V_S}{V_t} = \frac{K_R}{1 + T_R S} \tag{4}$$

Block diagram of power system regulator

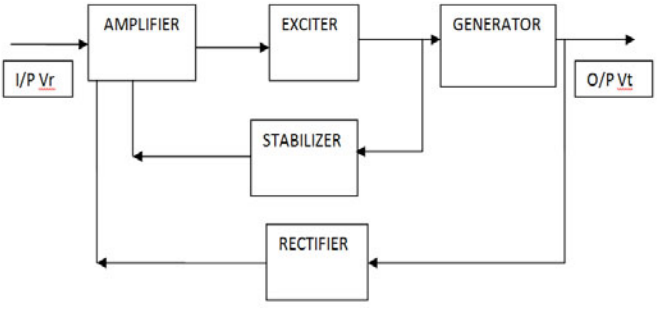


Fig. 1.

The gains and time constants selected for all the regulating systems are shown below:

Table 1. Gains and Time constants

Specifications	Linearized	Stabilized	FLC	NNC
K_A	10	10	1	1
T_A	0.05	0.05	0.05	0.05
K_E	1	1	1	1
T_E	0.5	0.5	0.5	0.5
K_G	1	1	1	1
T_G	1	1	1	1
K_R	-	0.1	-	-
T_R	-	0.02	-	-

The Matlab simulation obtained from FLCGUI for linearized and stabilized regulators are shown in Fig. 2 & Fig. 3.

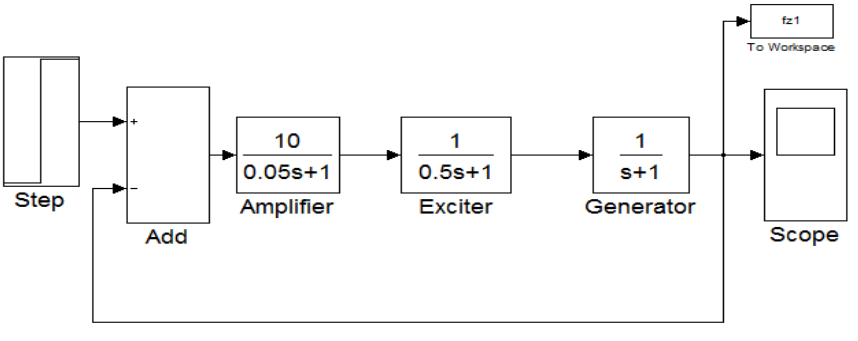


Fig. 2.

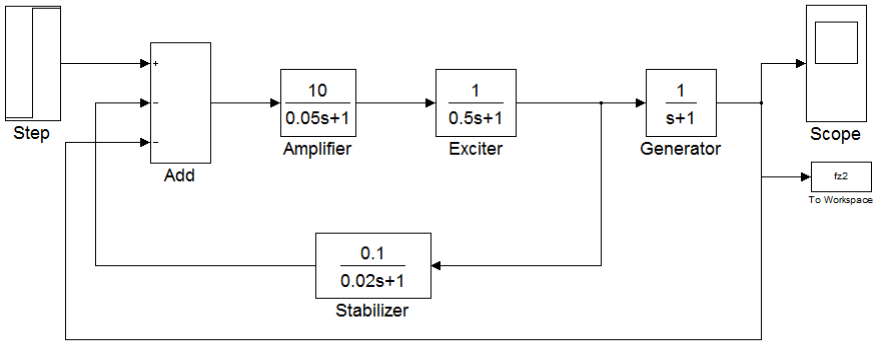


Fig. 3.

3 FLC Structure

Expressing some systems in the form of a mathematical model is very difficult. Fuzzy logic is a logical system, which is an extension of multi valued logic. Fuzzy Logic is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with un sharp boundaries in which membership is a matter of degree. In effect, much of FL may be viewed as a methodology for computing with words rather than numbers. Fuzzy logic provides a simple tool to interpret this experience into reality. Fuzzy logic controllers are rule-based controllers. The structure of the FLC resembles that of a knowledge based controller except that the FLC utilizes the principles of fuzzy set theory in its data representation and its logic.

-Fuzzification – the function is to read, measure, and scale the control variable and to transform the measured numerical values to the corresponding linguistic fuzzy variables with appropriate membership values

-Knowledge base - this includes the definitions of the fuzzy membership functions defined for each control variables and the necessary rules that specify the control goals using linguistic variables of a fuzzy if-then rule

-Inference mechanism – it should be capable of simulating human decision making and influencing the control actions based on fuzzy logic;

-Defuzzification module – which converts the inferred decision from the linguistic variables back into the numerical values.

4 FLC Design

The design process of an FLC may split into the five steps described as:

a. Selection of the control variables

The selection of control variables depends on the nature of the controlled system and the desired output. Here the output error (e) and the rate of derivative of the output (de) are used as controller inputs.

b. Membership function definition

Each of the FLC input signal and output signal, fuzzy variables ($Xj=\{e,\ de,u\}$), has the real line R as the universe of discourse. In practice, the universe of discourse is restricted to a comparatively small interval [*Xminj, Xmaxj*]. The universe of discourse of each fuzzy variables can be quantized into a number of overlapping fuzzy sets .The number of fuzzy sets for each fuzzy variables varies according to the application. The reasonable number is an odd number (3,5,7…). Increasing the number of fuzzy sets results in a corresponding increase in the number of rules. Membership functions can be of a variety of shapes, the most usual being triangular, trapezoidal, singleton or an exponential. The development of the control system based on fuzzy logic involves the following steps:

- Fuzzification
- Data base building;
- Rule base formation
- Inference machine elaboration;
- Deffuzification strategy.

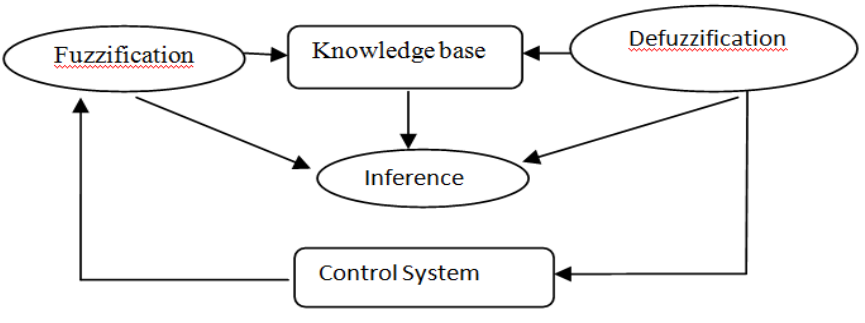


Fig. 4.

The inputs of FL Regulator are defined as the voltage error e (k) and change of error de (k) . The fuzzy controller ran with the input and output normalized universe [-1, 1]. Fuzzy sets are defined for each input and output variable. There are three fuzzy levels (N - negative, Z - zero, P –positive). The membership functions for input and output variables are triangular. The min - max method inference engine is used; the defuzzify method used in this FLC is centroid. The complete set of control rules is shown in table 2.

Table 2. Fuzzy Rule Base

e/de	N	Z	P
N	N	Z	Z
Z	Z	Z	P
P	Z	P	P

The fuzzy logic input and output variables with triangular membership functions are shown in Fig.5, Fig. 6 & Fig.7.

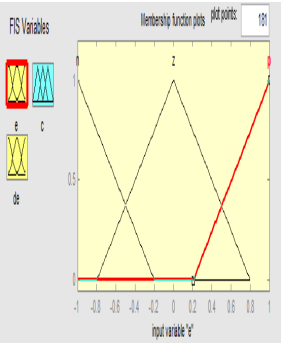


Fig. 5.

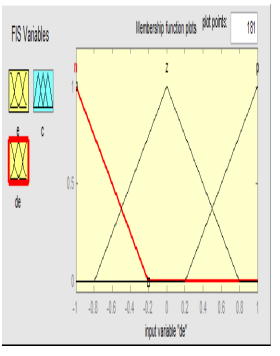


Fig. 6.

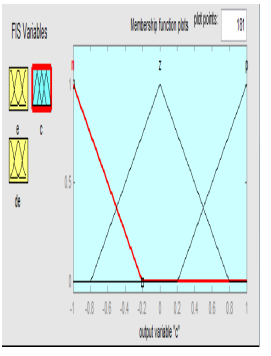


Fig. 7.

Each of the 9 control rules shown in fig. 8 represents the desired controller response to a particular situation. The block diagram presented in fig. 9 shows a FL Regulator in the Matlab simulation (ANFIS edit) and in fig. 10 the simulation of the surface control is presented.

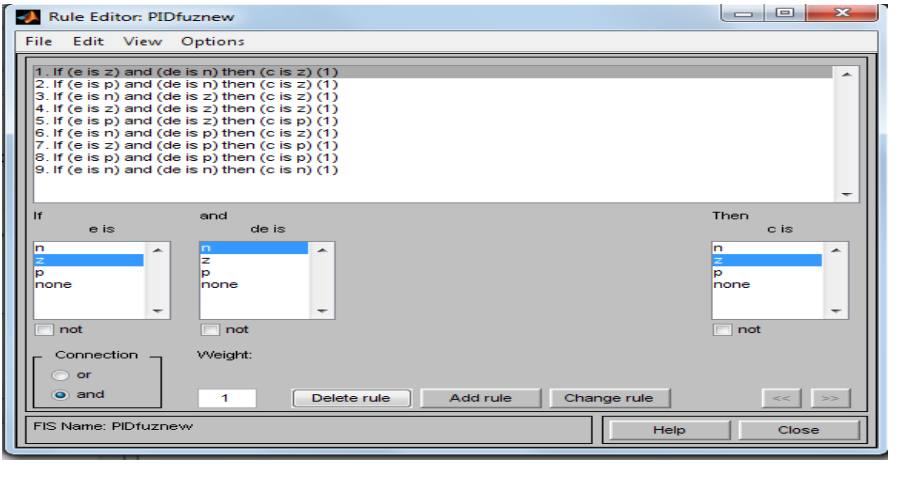


Fig. 8.

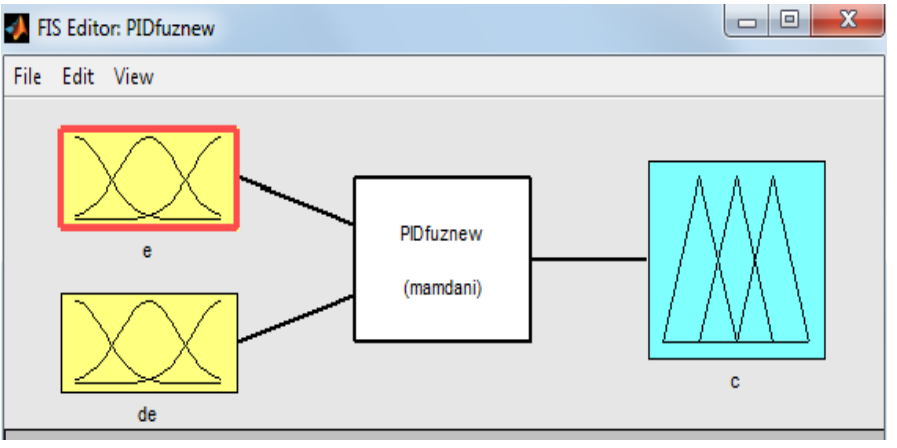


Fig. 9.

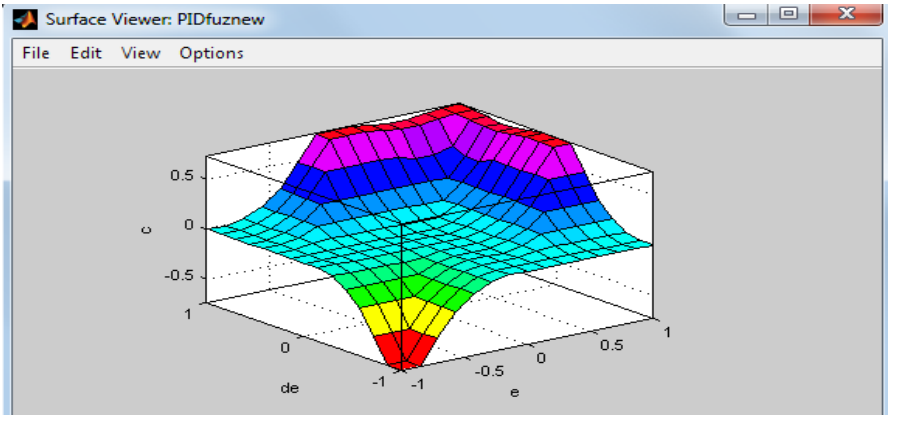


Fig. 10.

The simulation of FL regulator driving the linearized regulator is shown in Fig.11.

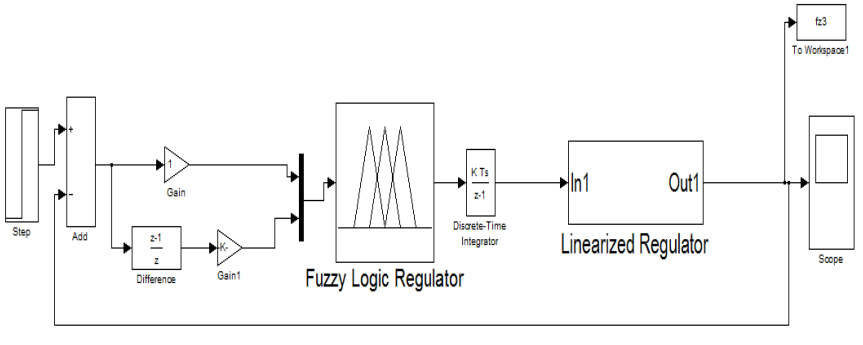


Fig. 11.

5 Neural Networks

Artificial Neural Network is a system loosely modelled on the human brain. It is a parallel distributed processing neuro computing natural intelligent system based on machine learning algorithms. It is a module that simulates the connection of multiple layers of simple processing elements called neurons in a sophisticated software. Each neuron is linked to certain of its neighbours with varying coefficients of connectivity that represents the strength of these connections. Learning is accomplished by adjusting these strengths to the overall network to output the appropriate results.

Backpropogation algorithm

Standard back propagation is a gradient descent algorithm, as is the Widrow - Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. In its first phase the input data is sent from input to output layer and in second phase the calculated error is propagated backwardly to input layer in order to change the weights of hidden layers. One iteration of this algorithm can be written as:

$$X_{n+1} = X_n + \alpha_n \times g_n \ldots\ldots\ldots\ldots \quad \ldots (11)$$

X_n : Vector of current weights and biases, g_n : current gradient, α_n : Learning rate

The neural network training data is obtained by running the simple PID Controller shown in Fig. 12. The variables p3 and t3 are stored in work space that are called back by neural network programme through the GUI. The network uses back propagation algorithm. The feed forward neural network is initialized with a performance goal of 0.00001 and 1,00,0000 epochs with the 'tansig ' transfer functions and 'trainlm' back propagation with 'learngdm' gradient descent algorithm. The performance plot and the validation check of the trained neural network is shown by Fig. 13. For the input value of 0.1212, the NNC gives an output of 0.9971.Thus it is trained for the improved step response. The trained NN is utilized as a Controller to Lineralized Regulator as shown in Fig.14.

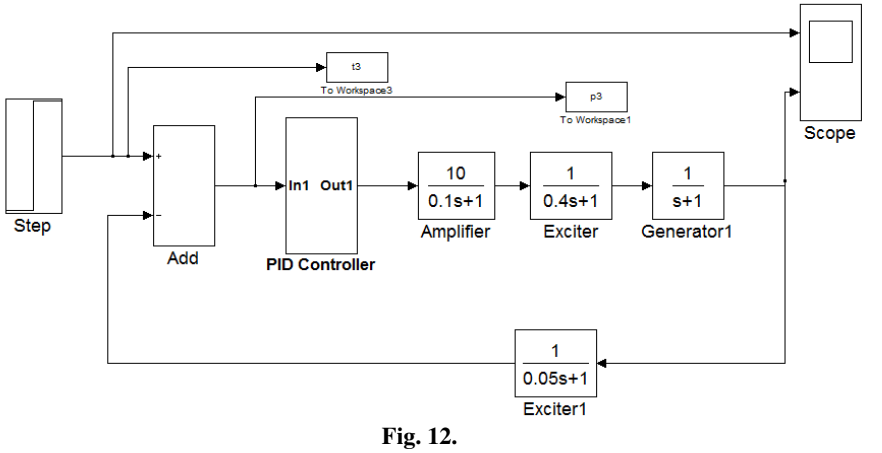


Fig. 12.

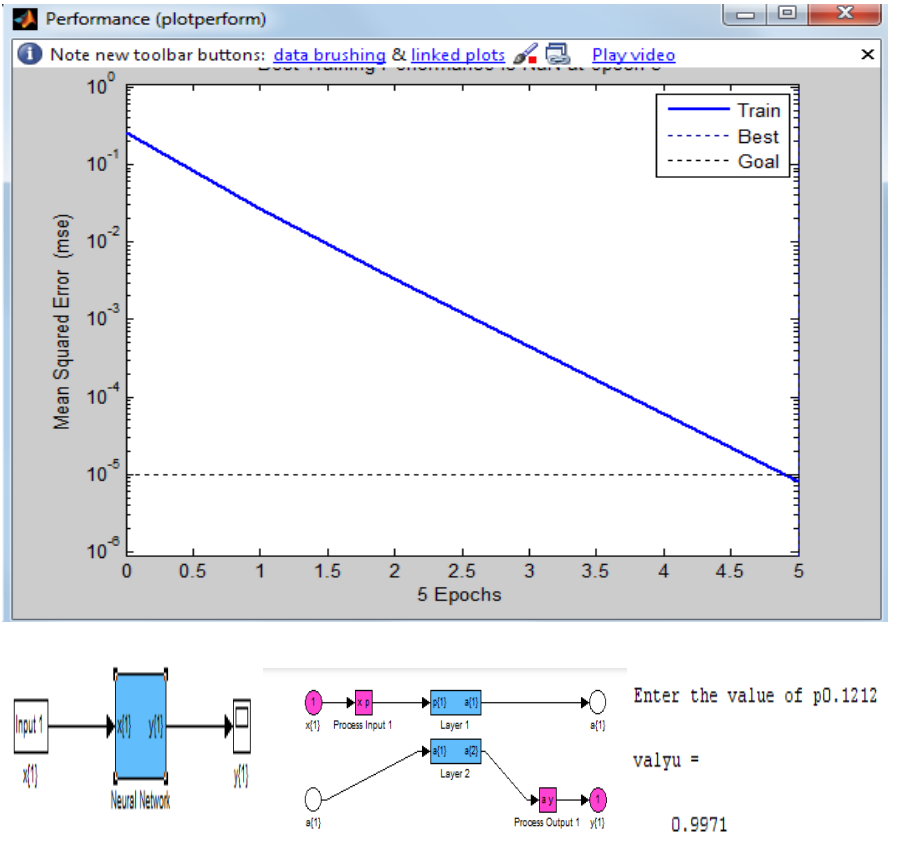

Performance (plotperform)
Note new toolbar buttons: data brushing & linked plots Play video
Train
Best
Goal
Mean Squared Error (mse)
5 Epochs
Input 1
x{1}
Neural Network
y{1}
Process Input 1
Layer 1
Layer 2
Process Output 1
Enter the value of p0.1212
valyu =
0.9971

Fig. 13.

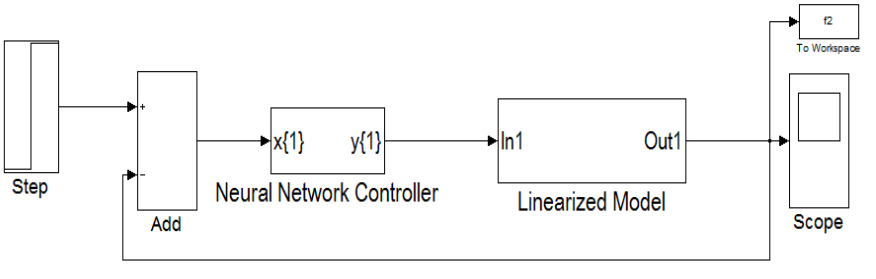

Step
Add
x{1}
y{1}
Neural Network Controller
In1
Out1
Linearized Model
f2
To Workspace
Scope

Fig. 14.

6 Graphical User Interface

GUIDE is the MATLAB Graphical User Interface, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of designing and building GUIs. We can use the GUIDE tools to

-Lay out the GUI

Using the GUIDE Layout Editor, we can lay out a GUI easily by clicking and dragging GUI components -- such as panels, buttons, text fields, sliders, menus, and so on -- into the layout area.

-Program the GUI

GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for all the GUI call backs -- the commands that are executed when a user clicks a GUI component. Using the M-file editor, we can add code to the call backs to perform the functions we want them to.

7 Matlab Implementation

The GUI is created to simulate the liberalized regulator and stabilized regulator. The GUI uses the control rules shown in Table2. And it simulates the response of Fuzzy Logic Regulator. The trained neural network is called back in GUI and the neural network response is simulated. The GUI also enables us to view the block diagram of fig.1 (named block diagram), simulation models by selecting model1, model2, model3 and model4 pushbuttons. Also step responses of all the 4 regulators cab be selected from popup menu. Thus the entire work can be controlled and monitored by FLCNNGUI shown in Fig.11.Following call backs are generated for the execution of FLCNNGUI:

```
FLCNNGUI
FLCGUI_OpeningFcn
FLCGUI_OutputFcn
p1_Callback
p2_Callback
p3_Callback
p4_Callback
p5_Callback
p6_Callback
p7_Callback
p8_Callback
popupmenu1_Callback
popupmenu1_CreateFcn
pushbutton10_Callback
pushbutton9_Callback
```

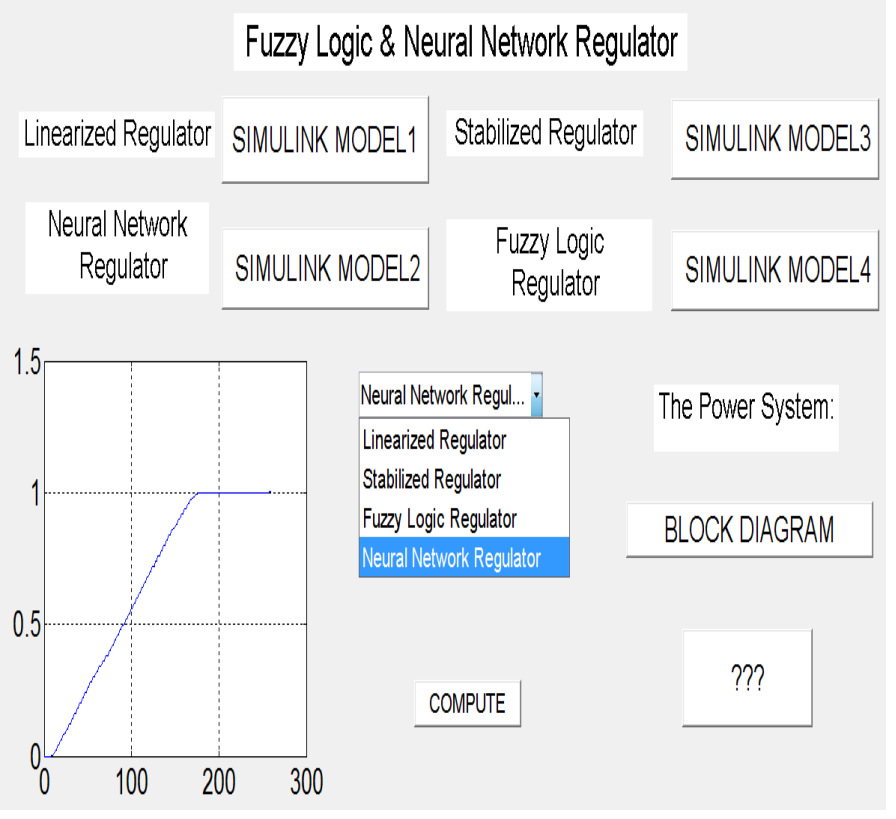


Fig. 15 Final GUI

8 Conclusion

The simulation results show that the linearized model has many oscillations, peak overshoot is high and steady state error is large. By using the stabilized regulator the overshoot can be controlled but still a large steady state error appears. The fuzzy logic Regulator requires exact input-output relations for the formation of appropriate rule base. The FLC regulator with approximated rule base has no overshoot and has zero steady state error. But the settling time is little bit higher. While the neural network works even with the partial information. The Neural Network Regulator has zero overshoot and zero steady state error with least settling time. So for the fast acting systems that can't tolerate oscillations, the FLC and NNC are the best regulators.

Table 3. Comparisons of Controllers

Specifications	Linearized	Stabilized	FLC	NNC
T_p	High	Lower	Much Lower	Lowest
T_r	High	Lower	Much Lower	Lowest
T_s	-	-	Much Lower	Zero
E_{ss}	-	-	Zero	Zero

9 Results

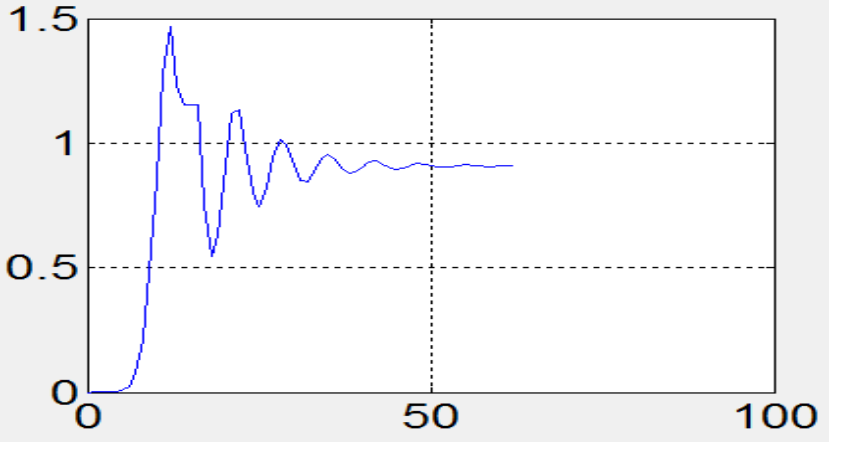


Fig. 16. Lineralized Regulator

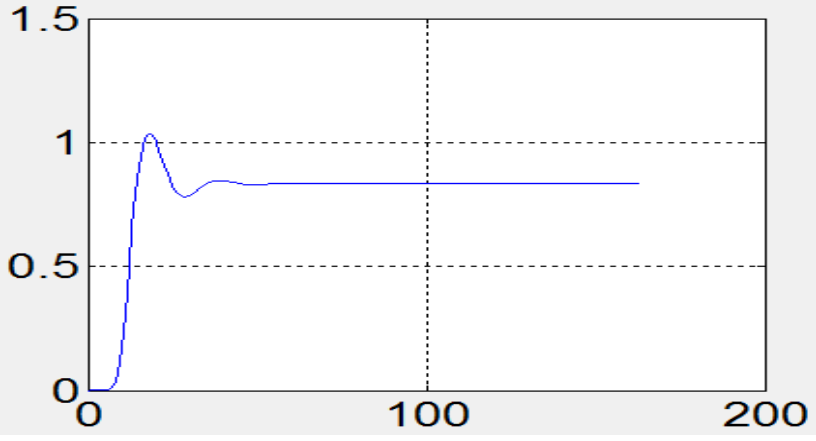


Fig. 17. Stabilized Regulator

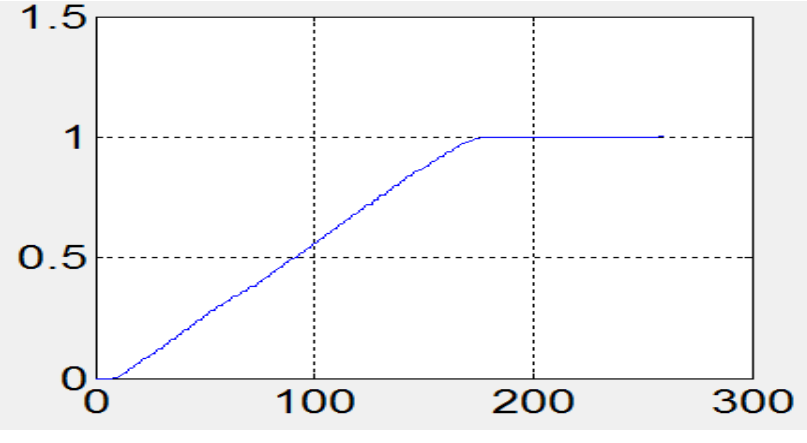


Fig. 18. Fuzzy Logic Regulator

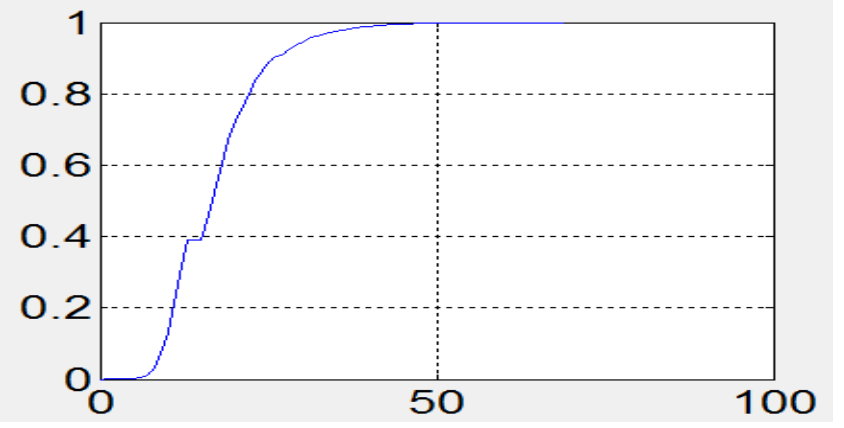


Fig. 19. Neural Network Regulator

References

[1] Kunder, P.: Power System Stability and Control. McGraw, Hill, New York (2001)
[2] Larsen, E.V., Swann, D.A.: Applying Power System Stabilizers Part I-III. IEEE Transactions on Power Apparatus and Systems PAS-100(6), 3017–3041 (1981)
[3] Hadisaadat: Power System Analysis. Tata McGraw, Hill (2002)
[4] Matlab 2008- Control system toolbox, Fuzzy Logic Toolbox, Neural Network Toolbox
[5] Ghoshal, S.P.: Optimization of PID gains by particle swarm optimization in fuzzy based automatic generation control. Electr. Power Syst. Res. 72203–72212 (2004)
[6] Gaing, Z.L.: A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans. Energy Convers. 19, 384–391 (2004)

Power Quality Improvement of Distribution System by Optimal Placement of Distributed Generators Using GA and NN

S. Chandrashekhar Reddy[1] and P.V.N. Prasad[2]

[1] Assoc.Professor, Dept. of EEE,
Christu Jyoti Institute of Technology & Science,
Jangaon, Andhra Pradesh
chandureddy6@yahoo.com, chandrashekharreddyphd@gmail.com
[2] Professor, Dept. of Elec. Engg.,
Univ. College of Engg.,
Osmania University, Hyderabad
polaki@rediffmail.com

Abstract. Distributed generator (DG) is now commonly used in distribution system to reduce total power loss and to improve the power quality of the network. The major task of connecting DG units is to identify their optimal placement in the system and to evaluate the amount of power to be generated in the DG. By considering this objective, a hybrid technique using Genetic algorithm and Neural-network is proposed in this paper. By placing DGs at optimal locations and by evaluating generating power based on the load requirement, the total power loss in the system can be minimized without affecting the voltage stability of the buses. Due to reduction of total power loss in the system and improvement of bus voltages, the power quality of the system increases. The results show the improved performance of proposed method for different number of DGs connected in the system.

Keywords: Distributed generator (DG), Genetic algorithm (GA), Neural network (NN), Newton-Raphson method.

1 Introduction

In the field of electric power system infrastructure and market, the need and importance of Distributed Generation has grown to a large extent in the recent years. DG is a small-scale power generation that is normally connected to the distribution system. In order to reduce the cost of service, the DGs use several modular technologies that are located throughout a utility's service area [1]. Distributed generation is an approach, which minimizes the amount of power loss that occur during transmission by generating the power very close to load centre.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 257–267.
springerlink.com

Some of the major advantages of accommodating DG units in distribution level are peak load shaving, enhanced system security and reliability, improved voltage stability, grid strengthening, reduction in the on-peak operating cost, reduction of network loss etc [2]. The main motive of applying DGs is to increase rational use of energy, deregulation or competition policy, diversification of energy sources, availability of modular generating plant, ease of finding locations for smaller generators, shorter construction time and lower capital costs for smaller plants, and proximity of the generation plant to heavy loads, which reduces the transmission costs [3].

Several technologies are used for DG sources such as photo voltaic cells, wind generation, combustion engines, fuel cells and other types of generation from the resources that are available in the geographical area [4]. Generally, DG units are integrated in the existing distribution system and studies have been done for finding the best location and size of DGs to produce utmost benefits [5]. The main goals that are considered while identifying the optimal DG location and size are the minimization of transmission loss, maximization of supply reliability, maximization of profit of the distribution companies etc [6]. Due to substantial costs, the DGs should be allocated properly with optimal size for enhancing the system performance in order to minimize the system loss as well as to improve the voltage profile while maintaining the system stability [7]. The effect of placing DG on network indices will differ based on its type and location and load at the connection point [8].

In this paper, the optimal placement of DG units and amount of power generated by these units are computed using two Genetic algorithms and Neural-network in three stages. In the first and third stages, GA is used and in the second stage NN is used. By using this method, the total power loss in the system can be reduced and thus the quality and reliability of the system can be increased. The rest of the paper is organized as follows: Section 2 describes the proposed technique with sufficient mathematical models and illustrations; Section 3 discusses the implementation and results and Section 4 concludes the paper. Some of the recent research related to optimal location and size of distributed generator are as follows.

Amanifar *et al*. [9] have proposed a Particle Swarm Optimization (PSO) algorithm for finding the optimal placement and size of DGs and the proposed method was tested on a 15- bus test system. Their aim was to reduce the total cost of the system, real power loss and the number of DGs to be connected. Jahani *et al.* [10] have presented a comparison between combination of heuristic search and PSO optimization methods and Modified Shuffled Frog Leaping Algorithm (MSFLA) for optimal placement and sizing of distributed generation in a distribution network. The objective function contains modified voltage profile and power losses and the proposed method was tested on the IEEE 33-bus system. Injeti *et al*. [11] have proposed a methodology for optimal planning and operation of active distribution networks regarding the location and sizing of DG units on 12-bus, 33-bus and 69-bus radial distribution networks. In order to reduce the real power losses and to enhance the voltage profile, Lalitha *et al*. [12] have presented a Fuzzy and PSO technique for the installation of Distributed Generators in radial distribution systems.

2 Optimal Placement of DG Units Using GA and NN

The optimal placement of DG units on the buses of distribution network at minimum power loss and the amount of power to be generated by these units is the main objective. By fixing DGs at suitable locations and generating power based on the load requirement, the quality of power of the system can be increased. By considering the above problem, a new method is proposed for optimal placement of DGs and computing the amount of power to be generated by DGs using Genetic algorithm and Neural-network. The proposed method consists of three stages. In the first stage, Genetic algorithm is used to generate the training dataset for various number of DGs to be connected in the system; in the second stage Neural network is used to identify the best location for the given number of DGs to be connected in the system and in the third stage Genetic algorithm is used to compute the amount of power to be generated by the DGs to reduce the power losses. Initial step for identification of optimal locations of DGs is computation of power flow between the buses.

Power flow between the buses is calculated using Newton-Raphson method. It is the commonly used method for load flow calculation because it has the advantage of lesser number of iterations, when compared to other methods. The real and reactive power flow in the buses mainly depend on voltage and angle values and are computed using the Eq. (1) and Eq. (2).

$$P_i = \sum_{k=1}^{N} V_i \times V_k \left(G_{ik} \times \cos\theta_{ik} + B_{ik} \times \sin\theta_{ik} \right) \tag{1}$$

$$Q_i = \sum_{k=1}^{N} V_i \times V_k \left(G_{ik} \times \sin\theta_{ik} - B_{ik} \times \cos\theta_{ik} \right) \tag{2}$$

where, N is the total number of buses, V_i & V_k are the voltage at i^{th} bus and k^{th} bus respectively, θ_{ik} is the angle between i^{th} bus and k^{th} bus and G_{ik} & B_{ik} are the values of conductance and susceptance respectively. After calculating the real and reactive power flow between the buses, the total power losses in the system can be computed using Eq. (3).

$$P_{loss} = \sum_{i,j=1}^{N} \mathrm{Re}\,al\left[Conj\left(\left(V_m(i) \right) \times \left(V_m(j) \right) \right) \times Y_{ij} \times B \right] \tag{3}$$

where, V_m is the voltage magnitude, Y_{ij} *i*s the Y-bus matrix and B is the base MVA.

The Y-bus matrix mainly depends on the values of resistance and reactance. After computing the total power loss in the system, the suitable locations needed to be identified for fixing the DGs is obtained using GA and NN and the amount of power to be generated by DG is computed by GA. The initial process of identifying the buses which satisfy the active and reactive power constraints is discussed using the following Genetic algorithm. The process that takes place in the proposed method consists of: (i) Generating training dataset using GA.

(ii) Neural-network to identify optimal placement of DGs (iii) Computing power to be generated by DG. This process is discussed in the following sections.

2.1 First Stage: Generating Training Dataset Using GA

The various possible optimal placements of DGs in the system is identified using Genetic algorithm which consists of five sub-stages, namely, generation of initial chromosomes, fitness function, crossover operation, mutation operation and termination.

The first step in Genetic algorithm operation is initializing the chromosome. Initially the number of input and output variables and the range of each variable are identified. In the present method, there are one input and three output genes. The input gene is the total number of DGs to be connected in the system and the output gene is the possible locations of connecting DGs and their corresponding real and reactive powers. The real and reactive powers are randomly generated within a certain power limits. The constraints of active & reactive powers of DG units are shown in Table 1.

Table 1 Active and Reactive Power Constraints

Constraints	Maximum	Minimum
Active Power	10.0 MW	0.0 MW
Reactive Power	5.0 MVAR	0.0 MVAR

Fitness function used in the proposed method is total power loss. The fitness function is calculated for the entire chromosomes generated in the above stage and the initial chromosome is ordered based on reduction of total power loss.

The crossover operation is done based on the crossover rate. By applying crossover between two parent chromosomes, a new chromosome is generated which is the combination of two parent chromosomes. This crossover operation is applied for all the chromosomes, so that one can obtain a new set of chromosomes. Then the fitness function using Eq. 3 is applied for these chromosomes and arranged based on lowest total power loss.

The mutation is done based on the mutation rate by randomly selecting the genes in the chromosome to obtain a new set of chromosomes. In this stage, the best chromosome is selected based on the fitness function. The above process is repeated until it reaches the maximum number of iterations.

After completing the process, a best set of chromosome is obtained based on reduction in power loss. The best set of chromosome obtained is the optimal placement for connecting one DG to the system.

The above process is repeated by increasing the total number of DGs to be connected as two & three so the output genes become six & nine respectively. From this Genetic algorithm, best set of chromosomes for placing one, two and three number of DGs in the system can be obtained. After generating training dataset, the next step is to train the Neural-network using the above generated training dataset.

2.2 Second Stage: NN to Identify Optimal Placement of DGs

The Neural-network is used for identifying the optimal placement of DGs for the given number of DGs to be connected. NN is used to select the best location from the few locations computed in the first stage, which satisfy the active and reactive power constraints. Usually, Neural-network consists of two stages, namely raining stage and testing stage. In the training stage, the network is trained based on the training dataset and in the testing stage, if the number of DGs to be connected as input, it gives the best location of installation of DGs in the system.

Neural network training consists of three layers, namely input layer, hidden layer and output layer. Here input layer consists of *x* layers, hidden layer consist of *n* layers and output layer consists of *3x* layers. The configuration of Neural-network used in the proposed method is shown in Fig 1. In the proposed method, the network is trained based on back propagation algorithm.

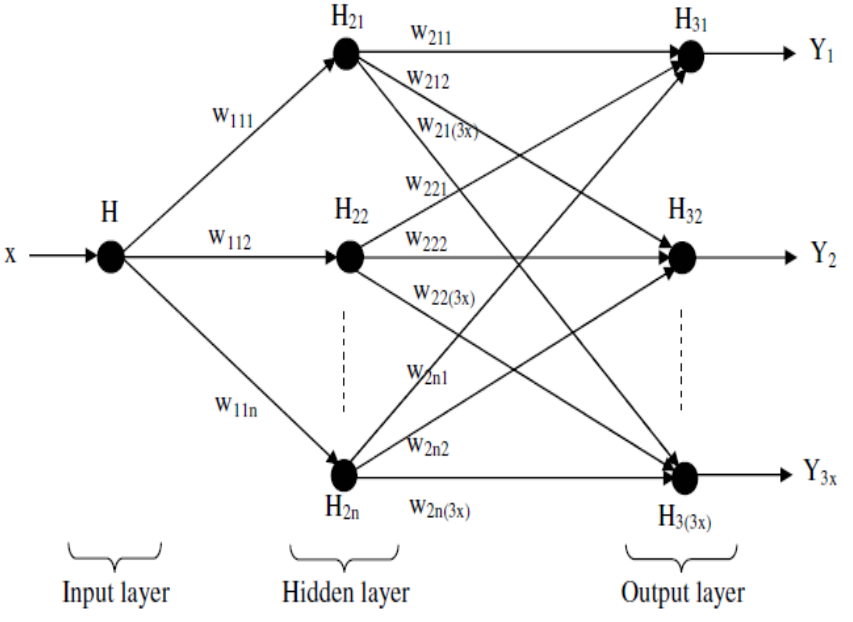


Fig. 1. Neural Network

The steps for training the Neural-network are: (i) Initialize the input weight of each neuron (ii) Apply the training dataset to the network. Here, *x* is the input to the network and Y_1, Y_2……Y_{3x} are the outputs of the network. Eq. (4), Eq. (5) and Eq. (6) represent the activation functions performed in the input and output layers.

(iii) Adjust the weights of all neurons. (iv) Determine the optimal placement of DGs to be connected.

$$Y_1 = \sum_{r=1}^{n} W_{2r1} Y_1(r) \tag{4}$$

$$Y_2 = \sum_{r=1}^{n} W_{2r2} Y_2(r) \tag{5}$$

$$Y_{(3x)} = \sum_{r=1}^{n} W_{2r(3x)} Y_{(3x)}(r) \tag{6}$$

$$\text{where, } Y(r) = \frac{1}{1+\exp(-W_{11r} \cdot x)} \tag{7}$$

In the testing stage, if the total number of DGs to be connected in the system is given as input, it gives the output as optimal placement of DGs and their real and reactive powers. The next process is to identify the power to be generated by the corresponding generator to reduce the total power loss in the system.

2.3 Third Stage: Computing DG Power generation Using GA

The amount of power to be generated by each generator is computed using GA in this stage, after obtaining optimal location of DGs in the second stage. The initial chromosome is taken from the output of neural network and the real and reactive power variables are changed. The real and reactive power values are randomly generated within a certain power limit for *N* chromosomes. Then for this power values, the total power loss in the system is computed in the evaluation stage and the chromosomes are ordered based on the total power loss.

Then crossover operation is applied to the above set of chromosomes with a crossover rate and so a new set of chromosomes are obtained and for that newly generated chromosome, fitness function is computed and arranged based on the total power loss. After applying crossover operation, then mutation operation is applied with a mutation rate. After completion of mutation, the termination process is executed. In the termination process, the real and reactive power with lowest total power loss is taken as the best power values to be generated by the DGs. By fixing DGs in the load bus with power generation, the total power loss reduces and the bus voltages remain stable, indicating improvement of power quality of the system.

3 Results and Discussion

The proposed method is implemented in MATLAB 7.11 for IEEE 30 bus system [13] and is shown in Fig 2. In the test system, bus 1 is considered as the slack bus and the base MVA of the system is 100 MVA. Buses 2, 13, 22, 23 and 27 are generator buses and remaining all other buses are load buses [13]. Table 2 shows the active & reactive powers, power factor and power loss after the installation of DG units.

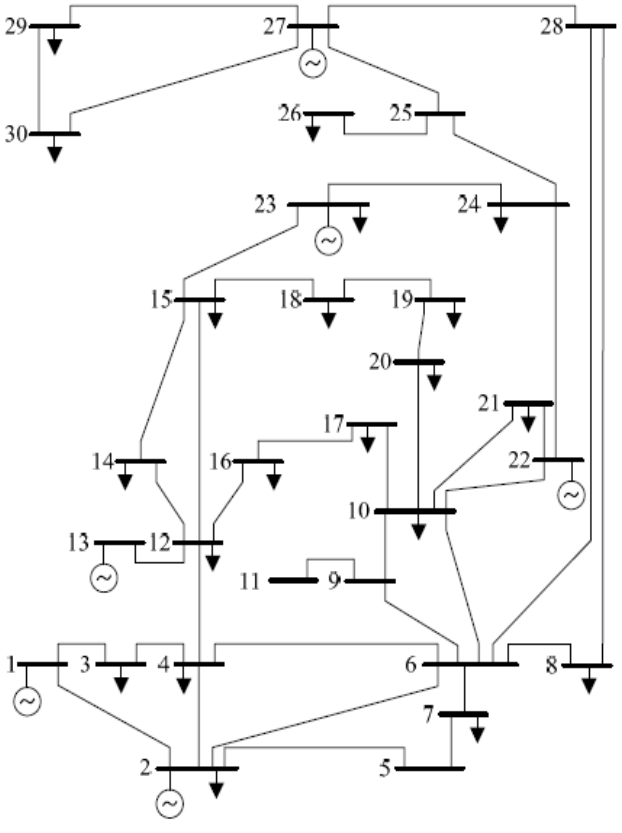


Fig. 2. IEEE 30 Bus System

Table 2. Power Loss with Installation of DG Sets

No. of DGs	DG Bus No.	Active Power of DG (MW)	Reactive Power of DG (MVAR)	Power Factor of DG	Total Power Loss (MW)	% Reduction of Power Loss
Zero	-	-	-	-	10.809	-
One	5	9.9505	1.4230	0.990	9.3717	13.3%
Two	7	9.8020	1.1669	0.993	8.6876	19.6%
	21	9.6635	4.1939	0.922		
Three	18	9.1255	3.2840	0.950		
	26	8.4990	0.5604	0.998	8.2197	24.0%
	30	9.9800	3.1268	0.954		

From Table 2, it is clear that the power loss in the system reduces after connecting DGs in the system. The power loss in the system without connecting DG is 10.809 MW. But after connecting one DG, the total power loss is reduced by 13.3%. In this case, the best location obtained from the present method is bus number 5 and power generated by that DG is 9.9505 MW. After connecting two DGs in the system, the optimal locations obtained are bus numbers 7 & 21 and the power generated by those DGs are 9.802 MW & 9.6635 MW respectively. The total power loss in the system is reduced to 8.6876 MW i.e., reduced by 19.6%. When three DGs are connected in the system, the best placement obtained are bus numbers 18, 26 & 30 and the power generated by DGs are 9.1255 MW, 8.499 MW & 9.98 MW respectively, and the total power loss is reduced to 8.2197 MW. Thus, there is 24% reduction of power loss with three DGs when compared to without DG in the system. All the DGs are found to be operating at good power factors between 0.922 and 0.998. Table 3 shows the per unit voltage profiles at different buses for without DG connection and with three different cases of DG connections.

Table 3. Per Unit Voltage Profiles at Different Buses

Bus No.	No DG	One DG	Two DGs	Three DGs	Bus No.	No DG	One DG	Two DGs	Three DGs
1	1.0600	1.0600	1.0600	1.0600	16	1.0485	1.0486	1.0524	1.0526
2	1.0400	1.0430	1.0430	1.0430	17	1.0421	1.0422	1.0473	1.0470
3	1.0284	1.0285	1.0301	1.0310	18	1.0322	1.0323	1.0375	1.0497
4	1.0204	1.0260	1.0225	1.0236	19	1.0290	1.0291	1.0345	1.0426
5	1.0100	1.0100	1.0100	1.0100	20	1.0327	1.0328	1.0382	1.0442
6	1.0170	1.0178	1.0196	1.0206	21	1.0334	1.0335	1.0434	1.0410
7	1.0068	1.0069	1.0102	1.0085	22	1.0453	1.0455	1.0515	1.0530
8	1.0100	1.0100	1.0100	1.0100	23	1.0336	1.0337	1.0428	1.0408
9	1.0541	1.0542	1.0575	1.0576	24	1.0315	1.0316	1.0383	1.0430
10	1.0469	1.0471	1.0527	1.0522	25	1.0292	1.0293	1.0335	1.0533
11	1.0820	1.0820	1.0820	1.0820	26	1.0174	1.0119	1.0161	1.0589
12	1.0625	1.0625	1.0647	1.0654	27	1.0362	1.0363	1.0389	1.0578
13	1.0710	1.0710	1.0710	1.0710	28	1.0174	1.0174	1.0191	1.0230
14	1.0478	1.0478	1.0512	1.0526	29	1.0160	1.0167	1.0194	1.0531
15	1.0429	1.0430	1.0478	1.0507	30	1.0053	1.0054	1.0081	1.0573

From Table 3, it can be seen that after connecting DGs in the system, the voltages in the buses remained stable well within the margins. The maximum and minimum voltages are 1.082 per unit and 1.01 per unit respectively. Compared with the voltage of the slack bus (1.06 per unit), the upper and lower percentage variations of bus voltages are 5.1% and -2.1% respectively for all the three cases of DG installations. These variations are shown in Fig. 3. Also there are no

changes in the voltages at the buses 1, 5, 8, 11 and 13. Hence, after connecting DG units in the system, the total power loss is reduced considerably and bus voltages remained stable within permissible limits.

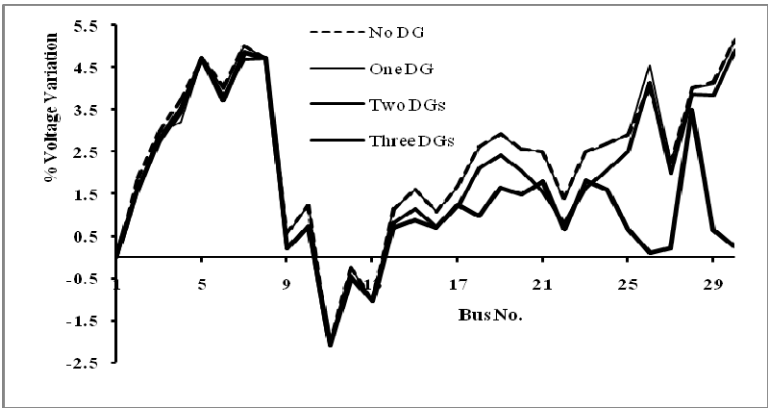


Fig. 3. Percentage Variations of Bus Voltages Compared with Slack Bus Voltage

In this method, 50 iterations have been performed to identify the amount of power to be generated by DGs and also to evaluate the total power loss. The variation of power loss reduction with number of iterations for the installation of one DG, two DGs & three DGs are shown in Fig. 4.

4 Conclusion

In this paper, optimal locations of DG units were computed for IEEE 30 bus system using GA and NN methods at minimum power loss. In the first stage GA is used to find the locations which satisfy the active and reactive power constraints. In the second stage NN is used to obtain the best location of DGs at minimum power loss and in the third stage generation capacities of DGs were evaluated using GA. The optimal locations for DG installations are bus 5 for one DG, buses 7 & 21 for two DGs and buses 18, 26 & 30 for three DGs. The total power loss without connecting a DG was 10.809 MW. When one DG is connected in bus 5 with 9.9505 MW power generation, the total power loss was 9.3717 MW. With two DG installations connected to buses 7 & 21 generating 9.802 MW and 9.6635 MW respectively; the total power loss was reduced to 8.6876 MW. After connecting three DGs in buses 18, 26 & 30 with respective power generations of 9.1255 MW, 8.499 MW & 9.98 MW, the total power loss was reduced further to 8.2197 MW. Thus there is a considerable reduction of power loss from 13% to 24% (from 1.4373 MW to 2.5893 MW) with the installation of DG units when compared with the system without DG. In all the above three cases, the voltage profile of the buses remained stable within tolerable limits and the power factor of the DGs

is maintained at more than 0.92. From the results, it can be concluded that the proposed method can be used effectively to find the optimal location of buses for installing DG sets in the system network to minimize total power loss maintaining the bus voltages within permissible margins. Thus there is considerable improvement in power quality and reliability.

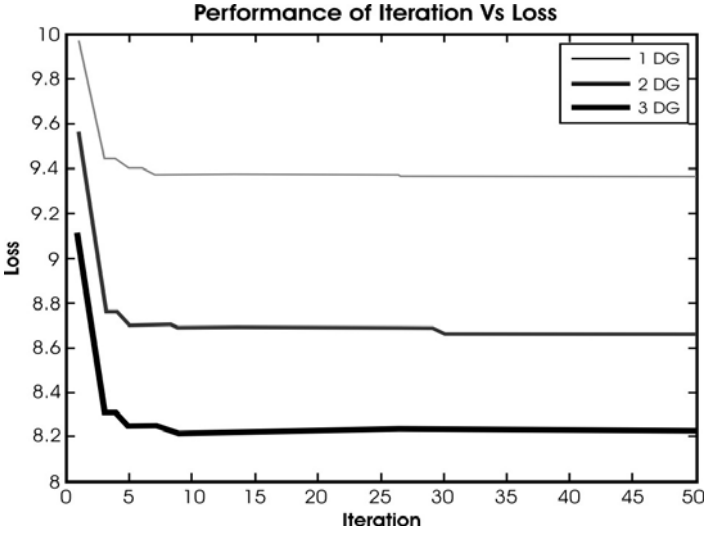


Fig. 4. Loss reduction vs. Iterations

References

[1] Varikuti, R., Damodar Reddy, M.: Optimal Placement of DG Units using Fuzzy and Real Coded Genetic Algorithm. Journal of Theoretical and Applied Information Technology 7(2), 145–151 (2009)

[2] Krueasuk, W., Ongsakul, W.: Optimal Placement of Distributed Generation using Particle Swarm Optimization. In: Proceedings of Power Engineering Conference in Australasian Universities, Australia (2006)

[3] Kotb, M.F., Shebl, K.M., El Khazendar, M., El Husseiny, A.: Genetic Algorithm for Optimum Siting and Sizing of Distributed Generation. In: Proceedings of 14th International Middle East Power Systems Conference, Egypt, pp. 433–440 (2010)

[4] Jain, N., Singh, S.N., Srivastava, S.C.: Particle Swarm Optimization Based Method for Optimal Siting and Sizing of Multiple Distributed Generators. In: Proceedings of 16th National Power Systems Conference, pp. 669–674. Osmania University, Hyderabad (2010)

[5] Hajizadeh, A., Hajizadeh, E.: PSO-Based Planning of Distribution Systems with Distributed Generations. International Journal of Electrical and Electronics Engineering 2(1), 33–38 (2008)

[6] Singh, D., Singh, D., Verma, K.S.: GA based Optimal Sizing & Placement of Distributed Generation for Loss Minimization. International Journal of Electrical and Computer Engineering 2(8), 556–562 (2007)
[7] Al-Rubayi, R.H., Alrawi, A.M.: Optimal Size and Location of Distributed Generators using Intelligent Techniques. Eng. & Tech. Journal 28(23), 6623–6633 (2010)
[8] Akorede, M.F., Hizam, H., Aris, I., Ab Kadir, M.Z.A.: A Review of Strategies for Optimal Placement of Distributed Generation in Power Distribution Systems. Research Journal of Applied Sciences 5(2), 137–145 (2010)
[9] Amanifar, O., Hamedani Golshan, M.E.: Optimal Distributed Generation Placement and Sizing for Loss and THD Reduction and Voltage Profile Improvement in Distribution Systems using Particle Swarm Optimization and Sensitivity Analysis. International Journal on Technical and Physical Problems of Engineering 3(2), 47–53 (2011)
[10] Jahani, R., Shafighi Malekshah, A., Chahkandi Nejad, H., Araskalaei, A.H.: Applying a New Advanced Intelligent Algorithm for Optimal Distributed Generation Location and Sizing in Radial Distribution Systems. Australian Journal of Basic and Applied Sciences 5(5), 642–649 (2011)
[11] Kumar Injeti, S., Kumar, N.P.: Optimal Planning of Distributed Generation for Improved Voltage Stability and Loss Reduction. International Journal of Computer Applications 15(1), 40–46 (2011)
[12] Padma Lalitha, M., Veera Reddy, V.C., Usha, V., Sivarami Reddy, N.: Application of Fuzzy and PSO for DG Placement for Minimum Loss in Radial Distribution System. ARPN Journal of Engineering and Applied Sciences 5(4), 30–37 (2010)
[13] http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.html

Evaluation of Proactive Fisheye Ad Hoc Source Routing Protocol for Various Battery Models in VANET Using Qualnet

Manish Sharma and Gurpadam Singh

[1] Department of Physics, Govt. College, Dhaliara, H.P. India
[2] Deparment of E & C Engg., B.C.E.T. Gurdaspur, Punjab, India
manikambli@rediffmail.com

Abstract. The modern world is VANET world where speed and Quality of Service (QoS) are the real characteristics. In VANET so many functions like gaming, internet etc. has been added leading to fast CPU clock speed hence more battery consumption. There is an increase in demand for wireless sensors and portable electronic devices in VANET. Hence wireless devices are becoming ubiquitous; batteries are used to power these devices. However, batteries are not durable and have to be replaced periodically. Choosing a right battery model is the key of above problem up to certain extent. In this paper we studied proactive Fisheye source Ad hoc routing protocol taking in to consideration various VANET parameters like speed, altitude etc. in real traffic scenario. The Proactive Fisheye source routing protocol is compared for battery models Duracell AA(MX-1500), Duracell AAA(MN-2400), Duracell AAA(MX-2400), Duracell C-MN(MN-1400) standard using Qualnet as a Simulation tool. Since Energy conservation is main focus area now days. Hence performance of the protocols with various battery models along with FIFO and residual battery parameters counts and helps to make a right selection of battery model. Varying parameters of VANET shows that in the real traffic scenarios Duracell AA (MX-1500) and Duracell AAA (MN-2400) performs more efficiently for energy conservation.

Keywords: VANET, Ad hoc Routing, FIFO, battery models, Qualnet.

1 Introduction

Vehicular Ad hoc Network (VANET) is a new communication paradigm that enables the communication between vehicles moving at high speeds. There is an increase in demand for wireless sensors and portable electronic devices in VANET. Hence wireless devices are becoming ubiquitous; batteries are used to power these devices. However, batteries are not durable and have to be replaced periodically .It has been found in the last decade so many functions like gaming, internet etc has been added leading to fast CPU clock speed hence more battery consumption. It is known that power consumption is proportional to frequency of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 269–278.
springerlink.com

operation. In order to improve QoS and energy conservation in fast moving vehicles various light weight routing protocols needed to be studied in Physical and data link layer. So that Right selection of the protocol can be made. There are mainly three types of routing protocols, Proactive [1], Reactive [2], Hybrid [3].The fisheye state routing protocol studied here is proactive in nature. Proactive routing protocol continuously updates the routing table, thus generating sustained routing overhead. These protocols are having different criteria for designing and classifying routing protocols for wireless ad hoc network. The Mobile Ad hoc Network (MANET) working group of the Internet Engineering Task Force (IETF) [4] develops standards for routing in dynamic networks of both mobile and static nodes. The protocols in focus now days are Hybrid protocols and others [7]. Its use in the context of VANET's along with reactive and proactive has always been area under investigation. Routing protocols are always challenging in the fast moving nodes as their performance degrades and such type of network is difficult to manage as fast handoff, signal quality, Interference maximizes along with other geographical factors. It must be noted that the battery capacity is tripled only in the last few years whereas amount of data transfer is increased from few Kbps to Several tens of Mbps. Hence designing of right scenario, Choosing energy efficient protocol and selection of right battery model is the need of the hour.

In this work, the feasibility, the performance, and the limits of ad hoc communication using the Fisheye routing protocol is evaluated as per battery models Duracell AA (MX-1500), DuracellAAA (MN-2400), DuracellAAA (MX-2400), Duracell C-MN(MN-1400)[11],and Potentials for optimizing the deployed transport and routing Protocols is investigated. Special care is taken in to provide Realistic scenarios of both road traffic and network usage. This is accomplished by simulating a scenario with the help of simulation tool Qualnet [6]. A micro simulation environment for road traffic supplied vehicle movement information, which was then fed in to an event-driven network simulation that configured and managed a VANET model based on this mobility data. The protocols and their various parameters of the transport, network, data link, and physical layers were provided by well-tested implementations for the networks simulation tool, while VANET mobility is performed by our own implementation. The Qualnet(6) is used as a simulation tool because of its wide applications and potential to give results nearly similar to that of real world.

2 Ad Hoc Routing Protocols

Routing protocol is a standard that controls how nodes decide how to route the incoming packets between devices in a wireless domain & further Distinguished in many types. There are mainly three types of routing protocols. Ad-hoc on demand vector distance vector (AODV), Dynamic MANET On demand (DYMO) and Dynamic source routing (DSR) are the examples of reactive routing protocols whereas Optimized Link State Routing (OLSR) and Fisheye state routing (FSR) are the examples of proactive routing protocols. Hybrid routing protocols is the combination of both proactive and reactive routing protocols, Temporary Ordered Routing Algorithm (TORA), Zone Routing Protocol (ZRP), Hazy Sighted Link

State (HSLS) and Orderone Routing Protocol (OOPR) are its examples. In our work the chosen protocol is Fisheye State routing Protocol and its Queuing parameter is also considered.

2.1 Fisheye State Routing

Fisheye State Routing (FSR)[1] is built on top of GSR. The novelty of FSR is that it uses a special structure of the network called the ‘‘fisheye.’ ’This protocol reduces the amount of traffic for transmitting the update messages. The basic Idea is that each update message does not contain information about all nodes. Instead, it contains update information about the nearer nodes more frequently than that of the farther nodes. Hence, each node can have accurate and exact Information about its own neighboring nodes. The following example explains the fish eye state routing protocol. In FSR, the network is viewed as a fish eye by each participating node. An example of this special structure is shown in Fig.1 Here; the scope of fish eye is defined as the set of nodes that can be reached within a given number of hops from a particular centre node. In the figure, we have shown three scopes with one, two, and three hops. The centre node has the Most accurate information about all nodes in the white circle and soon. Each Circle contains the nodes of a particular hop from a centre node. The advantage Of FSR is that, even if a node does not have accurate information about a destination, as the packet moves closer to the destination, more correct information about the route to the destination becomes available.

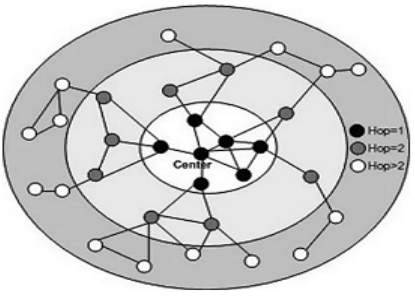


Fig. 1. Fisheye Structure

2.2 FIFO (First-In-First-Out)

A queue in which the first item in is the first item out is called FIFO [5]. In this a packet waits in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than average processing rate, the queue will up and new packets will be discarded. Note that a FIFO queue holds the packet .if the traffic consists of fixed size packet (e.g. cells in ATM networks) ,the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable-length packets, the fixed output rate must be based on the number of bytes or Bits.

3 Battery Models

Since batteries continue to power an increasing number of electronic systems, their life becomes a primary design consideration. Figure 2 illustrates a widening battery gap between trends in processor power consumption [11] and improvements in battery capacity [12]. Bridging this gap is a challenge that system designers must face for the foreseeable future. The need to improve battery life is in large part has driven the research and development of low power design techniques for electronic circuits and systems [13, 14]. Low power design techniques were successful in reducing the energy drawn from the battery and improving battery life. By understanding both the source of energy and the system that consumes it, the battery life can be maximized. Basically the zinc/potassium hydroxide/manganese dioxide cells, commonly called alkaline[15] or alkaline-manganese dioxide cells, have a higher energy output than zinc-carbon (Leclanche) cells. Other significant advantages are longer shelf life, better leakage resistance, and superior low temperature performance. In comparison to the zinc-carbon cell, the alkaline cell delivers up to ten times the ampere-hour capacity at high and continuous drain conditions, with its performance at low temperatures also being superior to other conventional aqueous electrolyte primary cells. Its more effective, secure seal provides excellent resistance to leakage and corrosion.

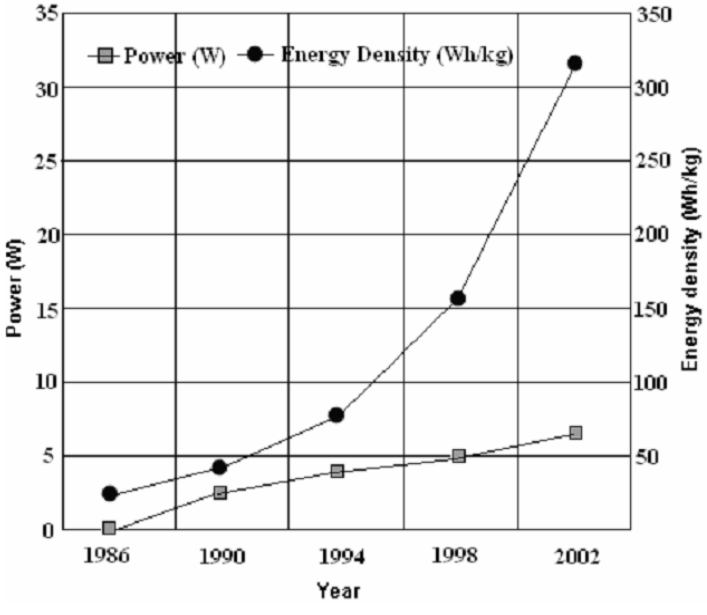


Fig. 2. A widening battery gap between trends in processor power consumption

The use of an alkaline electrolyte, electrolytic ally prepared manganese dioxide, and a more reactive zinc powder contributes to a higher initial cost than zinc-carbon cells. However, due to the longer service life, the alkaline cell is actually more cost-effective based upon cost-per-hour usage, particularly with high drains and continuous discharge. The high-grade, energy-rich materials composing the anode and cathode, in conjunction with the more conductive alkaline electrolyte, produce more energy than could be stored in standard zinc carbon cell sizes The product information and test data included in this section represent Duracell's newest alkaline battery products. Note that these battery models have some common battery parameters.

3.1 Duracell AA(MX-1500)

Table 1. Parameters of Duracell AA (MX-1500)

Parameters	Values
Nominal Voltage	1.5 V
Operating Voltage	1.6 - 0.75V
Impedance	81 m-ohm @ 1kHz
Typical Weight	24 gm (0.8 oz.)
Typical Volume	8.4 cm 3 (0.5 in.3)
Storage Temperature Range	-20°C to 35°C
Operating Temperature Range	-20°C to 54°C
Terminals	Flat
ANSI	15 A
IEC	LR6

3.2 Duracell AAA (MN-2400)

Table 2. Parameters of Duracell AAA (MN-2400)

Parameters	Values
Nominal Voltage	1.5 V
Operating Voltage	1.6 - 0.75V
Impedance	114 m-ohm @ 1kHz
Typical Weight	11 gm (0.4 oz.)
Typical Volume	3.5 cm 3 (0.2 in.3)
Storage Temperature Range	-20°C to 35°C
Operating Temperature Range	-20°C to 54°C
Terminals	Flat
ANSI	24 A
IEC	LR03

3.3 Duracell AAA (MX-2400)

Table 3. Parameters of Duracell AAA (MX-2400)

Parameters	Values
Nominal Voltage	1.5 V
Operating Voltage	1.6 - 0.75V
Impedance	114 m-ohm @ 1kHz
Typical Weight	11 gm (0.4 oz.)
Typical Volume	3.5 cm 3 (0.2 in.3)
Storage Temperature Range	-20°C to 35°C
Operating Temperature Range	-20°C to 54°C
Terminals	Flat
ANSI	24 A
IEC	LR03

3.4 Duracell C-MN (MN-1400)

Table 4. Parameters of Duracell C-MN (MN-1400)

Parameters	Values
Nominal Voltage	1.5 V
Operating Voltage	1.6 - 0.75V
Impedance	136 m-ohm @ 1kHz
Typical Weight	139 gm (0.4 oz.)
Typical Volume	3.5 cm 3 (0.2 in.3)
Storage Temperature Range	-20°C to 35°C
Operating Temperature Range	-20°C to 54°C
Terminals	Flat
ANSI	13 A
IEC	LR20

4 Simulation Tool

The adopted methodology for the results of this research work is based on simulations near to the real time packages before any actual implementation. Hence many parameters of VANET are taken.

Qualnet(6) is a fast, scalable and hi-fidelity network modelling software. It enables very efficient and cost-effective development of new network technologies. Qualnet is network modeling software that predicts performance of networking protocols and networks through simulation and emulation. Using emulation and simulation allows to reproduce the unfavourable conditions of networks in a controllable and repeatable lab setting. Qualnet enables users to Design new

protocol models, Optimize new and existing models, Design large wired and wireless networks using pre-configured or user-designed models, Analyze the performance of networks and perform what-if analysis to optimize them. Qualet (6) is the preferable simulator for ease of operation. So, we found Qualet be the best choice to implement our scenarios as we do not need every feature possible, just those for the token passing and message routing. Qualet is a commercial simulator that grew out of GloMoSim, which was developed at the University of California, Los Angeles, UCLA, and is distributed by Scalable Network Technologies [6]. The Qualet simulator is C++ based. All protocols are implemented in a series of C++ files and are called by the simulation kernel. QualNet comes with a java based graphical user interface (GUI).

Table 5. Simulation Parameters

Parameters	Values
Simulator	Qualnet Version 5.o.1
Terrain Size	1500 x 1500
Simulation time	3000s
No. Of Nodes	20
Mobility	Random Way Point, Pause time= 0s
Speed of Vehicles	Min.=5m/s Max.=20m/s
Routing Protocols	Fisheye Source Routing
Medium Access protocol	802.11b
Tx Power	150dbm
Data size	512 bytes
Data Interval	250ms
No. of sessions	5
Altitude	1500m
Weather mobility	100ms
Battery models	Duracell AA(MX-1500), Duracell AAA(MN-2400), Duracell AAA(MX-2400), Duracell C-MN(MN-1400).

5 Designing of Scenario

The scenario is designed in such a way that it undertakes the real traffic conditions using high speed mobility. We have chosen 20 fast moving vehicles in the region of 1500X1500 with the random way point mobility model. There is also well defined path for some of the vehicles, so that real traffic conditions can also be taken care of. It also shows wireless node connectivity of few vehicles using CBR application. The area for simulation is Hilly area with altitude of 1500 meters. Weather

mobility intervals is 100ms.Pathloss model is two ray with max prop distance of 100m.Here the data size is 512 bytes and transmission power is 150dbm. The speed of the vehicles varies from 5m/s to 20m/s. The data intervals choosen here is 250ms and the number of sessions is five.

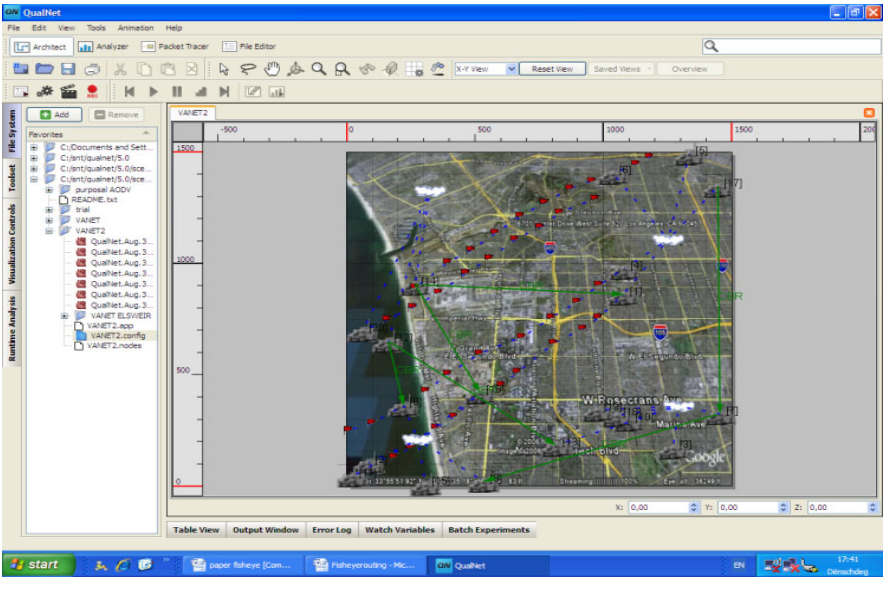


Fig. 3. Qualnet VANET Scenario

6 Results and Discussions

The simulation results as shown in Fig. 4 and 5 brings out some important characteristic differences between the various Battery models. It has been found that the residual battery capacity for Duracell AA (MX-1500) and Duracell AAA (MN-2400) is quite high as compared to other two. This is because they offers low impedance of 81 m-ohm @ 1kHz and 250 m-ohm @ 1kHz as compared to 114 m-ohm @ 1kHz and 136 m-ohm @ 1 kHz for Duracell AAA (MX-2400) and C-MN(MN-1400).Moreover ANSI values are 15A and 24A for Duracell AA(MX-1500) and Duracell AAA (MN-2400) as compared to 13A and 24 A for both Duracell C-MN (MN-1400), Duracell AAA (MX-2400).The Typical weight values also mend for consideration. The FIFO Peak Queue size also shows the better performance of Duracell AA(MX-1500) and Duracell AAA (MN-2400).Moreover the weight differences are also considerable for performances.

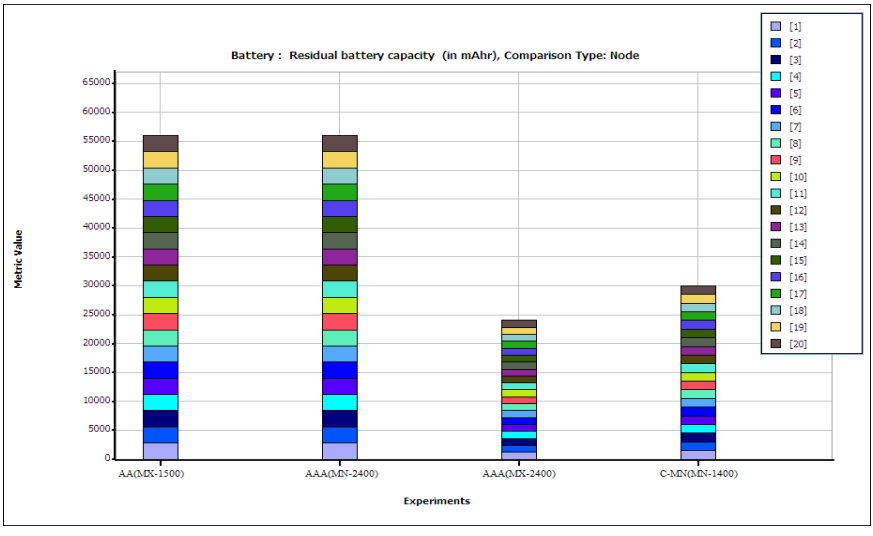


Fig. 4. Residual Battery Capacity for Fisheye Routing

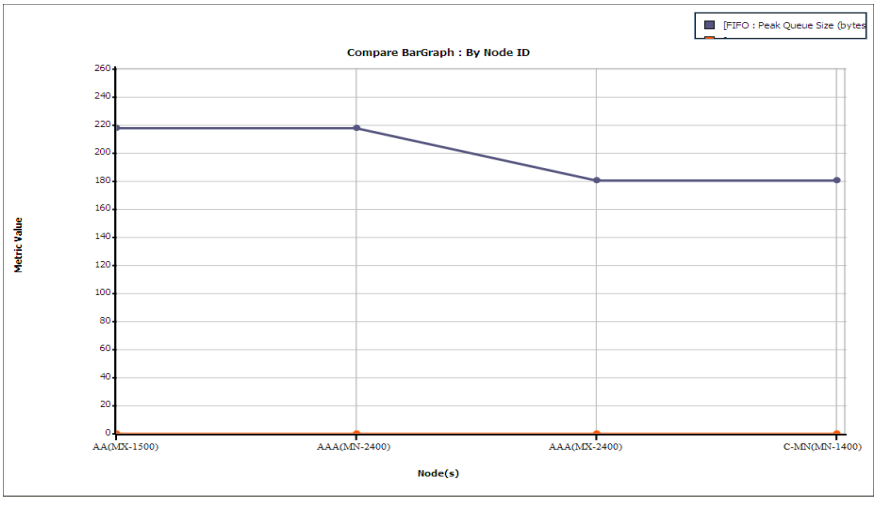


Fig. 5. FIFO Peak Queue Size for Fisheye Routing

7 Conclusions

Evaluation of the feasibility and the expected quality of VANETs operated as per various battery models, shows significant results. It has been found that the battery models depending upon their impedance, volume, weight, various temperature operating range shows significant variations in performance. Since energy conservations is most required now days. Hence its study leads to solutions to many

problems. As we know battery capacity has been increased three times only as compared to data transfer rate from few kbps to several tens of mbps. In future many parameters like longitude, latitude, geographical location, traffic, can also be considered for the exact results to real world.

References

[1] Lasson, T., Hedman, N.: Routing protocols in wireless Ad hoc network. Lulea university of technology, Stockholm (1998)
[2] Sommer, C., Dressler, F.: The DYMO Routing Protocol in VANET Scenarios. University of Erlangen-Nuremberg, Germany (2009)
[3] Mustafa, B., Raja, U.W.: Issues of Routing in VANET. Thesis MCS-2010-20 School of Computing at Blekinge Institute of Technology (2010)
[4] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11-1997
[5] Forouzan, B.A.: Data communications and Networking. Networking series. Tata Mcgraw-Hill (2005)
[6] Qualnet simulator version 5.0.1. Scalable Network Technologies, http://www.scalable-networks.com/
[7] Khan, I.: Performance evaluation of Ad hoc routing protocols for Vehicular ad hoc networks. Thesis Mohammad Ali Jinnah University
[8] Perkins, C., Royer, E., Das, S., Marina, K.: Performance comparison of two on-demand routing protocols for ad hoc network, pp. 16–28. IEEE Personal Communications (February 2001)
[9] Thong, L.K.: Performance analysis of mobile adhoc routing protocols. Thesis Naval post graduate college, Monterey, California (2004)
[10] Raw, R.S., Das, S.: Performance comparison of Position based routing Protocols in vehicle-to-vehicle(V2V) Communication. International Journal of Engineering Science and Technology 3(1), 435–444 (2011)
[11] Gunther, S.H., Binns, F., Carmean, D.M., Hall, J.C.: Managing the Impact of Increasing Microprocessor Power Consumption. Intel Technology Journal (First Quarter 2001)
[12] Buchmann, I.: Batteries in a portable world, http://www.cadex.com
[13] Chandrakasan, A.R., Brodersen, R.W.: Low Power Digital CMOS Design. Kluwer Academic Publishers, Norwell (1995)
[14] Rabaey, J., Pedram, M. (eds.): Low Power Design Methodologies. Kluwer Academic Publishers, Norwell (1996)
[15] http://www.duracell.com/
[16] http://www.panasonic.com/industrial/batteries-oem

A Collaborative Filtering Framework Based on Fuzzy Case-Based Reasoning

Shweta Tyagi and Kamal K. Bharadwaj

School of Computer and Systems Sciences,
Jawaharlal Nehru University,
New Delhi-110067, India
{shwetakaushik2006,kbharadwaj}@gmail.com

Abstract. Personalized recommendation systems have gained an increasing importance with the rapid development of Internet technologies. Collaborative filtering (CF) is the most promising technique in recommender systems, providing personalized recommendations to users based on their previously expressed preferences and those of other similar users. However, data sparsity and prediction accuracy are still major concerns related to CF techniques. Generally, the user-item matrix is quite sparse, which directly leads to the poor quality of predictions. In order to handle these problems, this paper proposes a novel approach to CF employing fuzzy case-based reasoning (FCBR), called CF-FCBR technique. Using fuzzy set theory for the computation of similarity between users and items, the proposed approach is twofold: offline and online. The offline processing is used to predict the missing values of user-item matrix and the online processing is employed for the process of recommendations generation. Our proposed approach helps in alleviating sparsity problem thereby improving recommendation accuracy. The experimental results clearly reveal that the proposed scheme, CF-FCBR is better than other traditional methods.

Keywords: Recommender systems, Collaborative filtering, Fuzzy sets, Case-based reasoning.

1 Introduction

Collaborative filtering (CF) [1] automatically predicts the interest of an active user by collecting rating information from other similar users or items. These techniques have been widely adopted in some large, renowned commercial systems, such as Amazon [2], AdaptiveInfo.com [3]. The research of CF is based on two types of approaches: model-based approaches and memory-based approaches [1]. The main problem of memory-based approaches is the data sparsity of the user-item matrix. There are several algorithms that have been proposed to alleviate the data sparsity problem [4], [5], [6]. In this article a novel approach is developed to deal with the problem of sparsity in the framework of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 279–288.
springerlink.com

CF. The proposed approach works by first filling missing values through the methodology of fuzzy case based reasoning (CBR) and then generating recommendations via CF.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of CBR in the framework of CF and a brief introduction of fuzzy CBR called FCBR. Section 3 explains the architecture of proposed CF-FCBR method. The results of an empirical analysis are presented in Section 4, followed by a conclusion in Section 5.

2 Related Work

2.1 CBR in the Framework of CF

The classic definition of CBR was coined by Reisbeck and Schank [7] in 1989. They described CBR as a problem solving technique which solves new problems by adapting previously successful solutions to similar problems. At a later stage, Watson [8], illustrated that CBR describes a methodology for problem solving but does not prescribe any specific technology. His research described four applications of CBR that variously use: nearest neighbor, induction, fuzzy logic and SQL. Some researchers used the application of CBR in the context of CF to tackle the problems of CF [5], [6], [9]. In the present work, we are employing CBR concepts and techniques in the framework of CF to handle the problem of sparsity.

The CF algorithm predicts a user's preferences for items based on the recorded preferences of other similar users or items. In a simpler manner, we can say that the CF works based on the assumption that the active user will prefer those items which the similar users prefer. On the other hand, CBR makes direct use of past cases to solve a new problem by adapting previously successful solutions of the past similar problems. The main steps of CBR process are: *Retrieve* the most similar case or cases, *Reuse* and *Revise* the solution suggested by these matching cases, *Retain* the solution to the problem [8].

According to, Guo et al [5], the steps of CF are much same as the steps of CBR and therefore the CF algorithm can be combined with CBR naturally.

2.2 Fuzzy Set Theory in CBR

Fuzzy set theory has proven to be very useful in addressing the problems of CBR [10, 11]. The most basic problems in CBR are the retrieval and selection of cases since the remaining operations will succeed only if the selected cases are relevant ones. Fuzzy Logic (FL) can be used in case representation to provide a characterization of imprecise and uncertain information. Moreover, in the phase of case retrieval, FL can be used to evaluate partial matches by means of fuzzy matching techniques [10]. Furthermore, FL can be utilized in case adaptation to

modify the selected case by using the concept of gradual rules [11]. In the proposed work, the fuzzy set theory is employed at the case representation level and also at the level of retrieval of appropriate cases.

3 Proposed Framework

The proposed work incorporates fuzzy set theory with CBR; called FCBR, in the framework of CF. Fig. 1 clearly delineates the architecture of proposed CF-FCBR model.

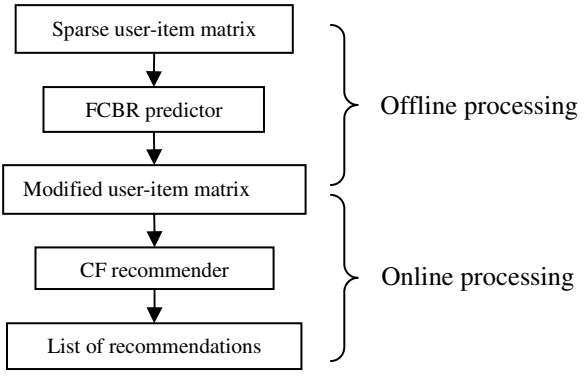


Fig. 1. Architecture of proposed CF-FCBR model

The proposed FCBR model deals with fuzzy similarity based CBR models stating that *"similar problems lead (or may lead) to similar solutions*". Using FCBR, the proposed work is twofold: offline and online. In offline processing, the FCBR model is employed to predict the missing values in user-item matrix. Whereas, online processing uses CF in order to make recommendations for the active user.

3.1 Offline Processing

Case Representation. The success of CBR system heavily depends on the structure or representation and content of its collection of cases. In this work, a case represents the user's rating vector and his profile vector. Table 1 shows the representation of cases in the case base. Here each user is represented as a case in the case base. In the user rating vector (URV), $\langle r_{i,1}, r_{i,2}, \ldots, r_{i,m} \rangle$, the value $r_{i,j}$ represents the rating of jth item by ith user u_i. Moreover, in the user profile vector (UPV) of ith user u_i, age, gender, and occupation are denoted by Age_i, $Gend_i$, and $Occup_i$ respectively and genre interestingness measure (GIM) [12] of kth genre is represented by $GIM_{i,k}$.

Table 1. Case Representation

Cases	User Profile Vector (UPV)	User Rating Vector (URV)
u_1	$\langle Age_1, Gend_1, Occup_1, GIM_{1,1}, GIM_{1,2}, \ldots, GIM_{1,18}\rangle$	$\langle r_{1,1}, r_{1,2}, \ldots, r_{1,m}\rangle$
u_2	$\langle Age_2, Gend_2, Occup_2, GIM_{2,1}, GIM_{2,2}, \ldots, GIM_{2,18}\rangle$	$\langle r_{2,1}, r_{2,2}, \ldots, r_{2,m}\rangle$
…	…	…
…	…	…
…	…	…
u_n	$\langle Age_n, Gend_n, Occup_n, GIM_{n,1}, GIM_{n,2}, \ldots, GIM_{n,18}$	$\langle r_{n,1}, r_{n,2}, \ldots, r_{n,m}\rangle$

Fuzzification. The crisp description of features does not reflect the actual case for human decisions. For example, two users of age 14 and 18 have the age difference of 4 years, while both users belong to the same age group, i.e. teenager. Therefore, a great advantage can be gained by fuzzifying the features of the cases in order to retrieve most similar cases. The fuzzy set concept provides a smooth transition between members and nonmembers of a set. The members of fuzzy sets have degrees of membership between 0 and 1. This value is assigned by the membership function (MF) associated with each fuzzy set.

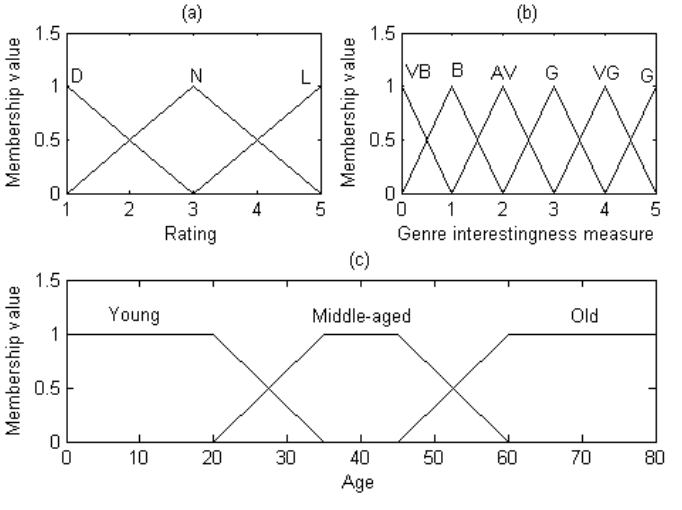


Fig. 2. Membership functions for (a) Rating, (b) Genre interestingness measure, and (c) Age

Distinct fuzzy sets are used to fuzzify the various features of a case. First of all numeric ratings of items are fuzzified into three fuzzy sets, Like (L), Neutral (N) and Dislike (D) as shown in Fig. 2(a). The GIM is fuzzified into six fuzzy sets, very bad (VB), bad (B), average (AV), good (G), very good (VG), and excellent

(E). Fig. 2(b) shows the membership value of GIM in different fuzzy sets. Furthermore, age is fuzzified into three fuzzy sets, young, middle-aged, and old as shown in Fig. 2(c). The values of other features, gender and occupation, are considered as fuzzy points with membership value of one.

Similarity Measure. The most important issue in the CBR system is to define the similarity measure. Basically, the similarity between two cases is computed based on the similarity between attributes of the cases. In our model, the attributes are fuzzified by using various fuzzy sets. Therefore, to compute the similarity between two cases (users) x and y, the fuzzy distance between each attribute is determined. The fuzzy distance between attributes of the cases is called the local fuzzy distance (LFD). In order to calculate the LFD between ith attribute values x_i and y_i of respective cases x and y, the Euclidean distance function is used.

$$LFD(x_i, y_i) = \sqrt{\sum_{j=1}^{l} \left(x_{i,j} - y_{i,j}\right)^2}, \tag{1}$$

where $x_{i,j}$ is the membership value of the feature x_i in its jth fuzzy set and l is the number of fuzzy sets. Consequently, the LFD between attributes is employed to obtain the fuzzy distance between the cases. This fuzzy distance between the cases is called the global fuzzy distance (GFD). To compute the GFD between two cases, we employ an aggregation operator. Therefore the GFD between two cases x and y, $GFD(x, y)$, can be defined as:

$$GFD(x, y) = \frac{1}{p} \sum_{i=1}^{p} LFD(x_i, y_i), \tag{2}$$

where p is the number of attributes of cases x and y. Using the GFD between two cases, the similarity between these cases is computed.

$$Sim(x, y) = 1 - GFD(x, y) \tag{3}$$

In the proposed FCBR model, the similarity between two cases is computed over UPVs as well as URVs. Since, two cases (users) are considered similar if they have similar rating patterns as well as similar demographic information and GIM. The similarity computed over URV is called Sim_{URV} and the similarity computed over UPV is called Sim_{UPV}. Finally, the two similarity measures are combined together by taking their weighted sum to find the similarity between cases x and y.

$$Sim(x, y) = \lambda * Sim_{URV}(x, y) + (1 - \lambda) * Sim_{UPV}(x, y), \tag{4}$$

where λ is a weight factor which can be selected empirically. The parameter λ is used for determining how the similarity relies on UPVs and URVs. λ = 1 states that similarity depends completely upon URVs and λ = 0 states that similarity depends completely upon UPVs. Once, the similarities between the new case and the cases in the case base are computed, a set of best matches is retrieved.

Missing Rating Prediction. For the prediction of missing rating values, each case in the case base with missing rating value is treated as a problem. Thus a set of similar cases that are potentially useful to the current problem are retrieved. The solutions to similar cases are adapted in order to retain the solution of current problem.

In the proposed framework, the current problem denotes a user $\boldsymbol{t}$ who has not rated an item $\boldsymbol{k}$. Hence, the prediction of missing rating value $\boldsymbol{r_{t,k}}$ is generated by employing Resnick's prediction formula [13] as follows:

$$P(r_{t,k}) = \bar{t} + \frac{\sum_{u \in S(t)} similarity(t,u) * (r_{u,k} - \bar{u})}{\sum_{u \in S(t)} similarity(t,u)}, \tag{5}$$

where $\bar{t}$ denotes the average rating of user t and $S(t)$ denotes the set of similar users. The set $S(t)$ is retrieved through the retrieval phase of the proposed model. In this way, by filling missing values, a dense user-item matrix is retained through FCBR model.

3.2 Online Processing

The dense user-item matrix assists the user-based CF algorithm in the process of prediction generation for an active user through online processing. The user-based CF first looks for users similar to active user. There are several similarity algorithms that have been used in user-based CF. In this work, the most widely used Pearson correlation is employed to define the similarity between two users x and y.

After finding a set of similar users, the user-based CF employs the ratings from those similar users to predict the ratings for the active user. The well known Resnick's prediction formula [13] is used to predict the rating for an active user.

4 Experiments and Results

To perform the experiments, we randomly choose 400 users out of 943 users of MovieLens[1] dataset. The experiments are conducted under several configurations by altering the size of training set to be 300 (ML300), 200 (ML200), and 100 (ML100) users. The remaining users, out of 400 users, are taken as active users. The number of training ratings provided by the active users are varied from 5, 10, to 20, and named as Given5, Given10, and Given20 respectively.

In order to estimate the predictive accuracy of a recommendation list, we rely on the metric - mean absolute error (MAE). The metric, MAE, computes the average of the difference between actual ratings and predicted ratings of unseen items [14], [15]. The coverage is also computed to measure that how many actually rated items are covered by the method [12].

[1] `http://www.grouplens.org/`

4.1 Comparison of Our Framework of CF with Other Methods

In order to show the performance increase of proposed CF-FCBR method, we compare our algorithm with some traditional algorithms: user-based CF [13], called UBCF and a new CF metric [16] for UBCF, called NUBCF. For making comparisons of various techniques, we kept the same neighborhood size of 35 users and set the value of parameter λ equal to 0.5. We compared the MAE and coverage of various competitive schemes as shown in Table 2 and Table 3.

Table 2. Comparison of MAE of proposed CF-FCBR method with UBCF and NUBCF methods on MovieLens

Training Users	Methods	Given5	Given10	Given20
ML100	UBCF	0.8922	0.8723	0.8464
	NUBCF	0.8702	0.8362	0.8249
	CF-FCBR	0.8465	0.8026	0.7917
ML200	UBCF	0.8783	0.8641	0.8371
	NUBCF	0.8598	0.8343	0.8182
	CF-FCBR	0.8340	0.7903	0.7710
ML300	UBCF	0.8639	0.8559	0.8404
	NUBCF	0.8407	0.8225	0.8098
	CF-FCBR	0.8174	0.8051	0.7702

Table 3. Comparison of coverage of proposed CF-FCBR method with UBCF and NUBCF methods on MovieLens

Training Users	Methods	Given5	Given10	Given20
ML100	UBCF	91.333	92.221	93.193
	NUBCF	93.779	94.388	94.906
	CF-FCBR	96.361	97.403	98.372
ML200	UBCF	92.851	93.082	93.752
	NUBCF	93.193	95.030	96.164
	CF-FCBR	96.500	97.483	98.278
ML300	UBCF	92.936	94.272	94.711
	NUBCF	94.664	95.883	96.830
	CF-FCBR	97.689	98.687	99.386

The results summarized in Table 2 and Table 3, clearly reveal that the CF-FCBR approach improves the recommendation quality of CF significantly. It is also observed from these experiments that the MAE and coverage of the methods improves with the increment in the size of training users as well as with the increment in the number of training items. However, the MAE and coverage of proposed scheme is always better than other competitive schemes for all configurations of the dataset.

Additionally, in order to analyze the sensitivity of the neighborhood size, we carried out an experiment (see Fig. 3). The MAE is computed for all configurations by varying the number of neighbors and results for the configurations ML100 with Given5 and ML300 with Given20 are shown in Fig. 3(a) and Fig. 3(b) respectively. We can observe that the accuracy of prediction improves with the increment in neighborhood size. Additionally, the proposed method outperforms other methods even with variable size of neighborhoods.

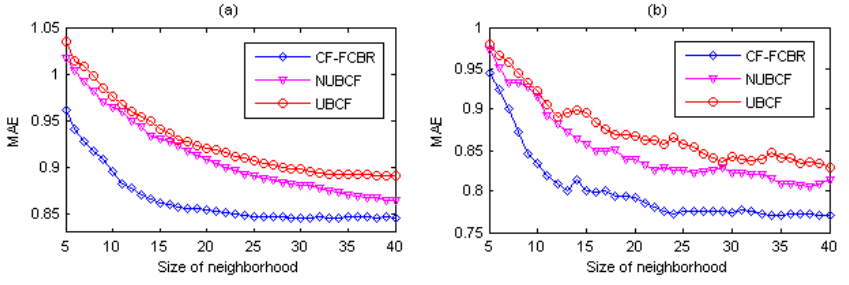


Fig. 3. Sensitivity of neighborhood size on MAE for the configurations: (a) ML100 and Given5 and (b) ML300 and Given20

4.2 Impact of λ

For missing data prediction, our algorithm computes the fuzzy similarity between users by taking a weighted sum of the similarity based on UPVs and the similarity based on URVs. The parameter λ is used as a weight factor to balance the two similarity measures. We varied the value of λ and analyzed the impact of λ on MAE of proposed algorithm. We test the proposed algorithm for all configurations of ML100, ML200, and ML300 of training users. The corresponding experimental results are reported in Fig. 4(a), Fig. 4(b), and Fig. 4(c) respectively. We observed from Fig. 4, that at the extreme values (0 and 1) of λ, the MAE of proposed algorithm is maximum for almost all configurations. This clearly indicates that the combination of similarity measures improves the recommendation accuracy.

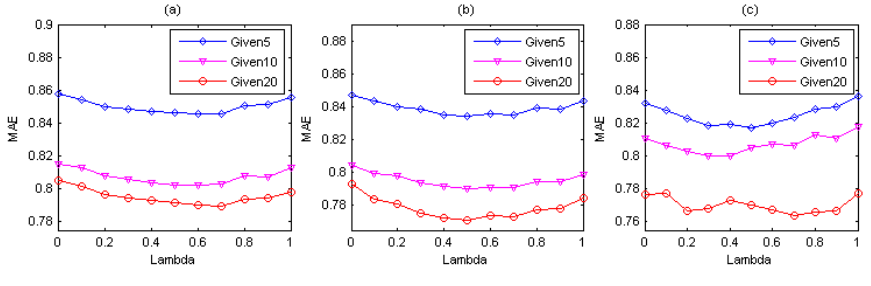


Fig. 4. Impact of Lambda (λ) on MAE for the configurations: (a) ML100, (b) ML200, and (c) ML300

5 Conclusion

In this paper, we propose a CF framework based on fuzzy case based reasoning (FCBR). This method first of all predicts the missing values and then generates a list of recommendations for the active user. Therefore, the proposed method successfully deals with the problem of sparsity and generates high accuracy recommendations based on FCBR. Empirical analysis shows that our proposed CF-FCBR algorithm outperforms other state-of-the-art CF approaches.

In the present work, we have considered all the features equally important for FCBR. By adding the feature importance factor to similarity measure, the accuracy of the recommender system can be further improved. Therefore, one of the possible extensions would be to take into consideration the weights of the features. Further in the present work, our main focus was to handle sparsity problem of CF and therefore trust and reputation [17] concepts have not been incorporated. However, one promising direction for future work would be to consider trust and reputation mechanisms to improve performance of the proposed recommender system.

References

[1] Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)

[2] Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1), 76–80 (2003)

[3] Billsus, D., Brunk, C.A., Evans, C., Gladish, B., Pazzani, M.: Adaptive Interfaces for Ubiquitous Web Access. Comm. ACM 45(5), 34–38 (2002)

[4] Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: SI-GIR 2007: Proc. 30th Annual International Conference on Research and Development in Information Retrieval, pp. 39–46. ACM, New York (2007)

[5] Guo, Y., Deng, G., Zhang, G., Luo, C.: Using Case-based Reasoning and Social Trust to Improve the Performance of Recommender System in E-Commerce. In: Proc. Second International Conference on Innovative Computing, Information and Control. IEEE, DC (2007)

[6] Zhu, X., Ye, H., Gong, S.: A Personalized Recommendation System Combining Case-Based Reasoning and User-Based Collaborative Filtering. In: Proc. 21st Annual International Conference on Chinese Control and Decision Conference, CCDC 2009, pp. 4062–4064. IEEE, NJ (2009)

[7] Riesbeck, C.K., Schank, R.: Inside Case-Based Reasoning. Erlbaum, Northvale (1989)

[8] Watson, I.: Case-based reasoning is a methodology not a technology. Knowledge-Based System 12, 303–308 (1999)

[9] Roh, T.H., Oh, K.J., Han, I.: CF recommendations based on Som cluster-indexing CBR. Expert Systems with Applications 25, 413–423 (2003)

[10] Dubois, D., Esteva, F., Garcia, P., Godo, L., de Mantaras, R.L., Prade, H.: Fuzzy set-based models in case-based reasoning. In: Tech. Rep. IRIT/96-54-R, Touluse, France (1997)

[11] Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
[12] Al-Shamri, M.Y.H., Bharadwaj, K.K.: Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. Expert Systems with Applications 35(3), 1386–1399 (2008)
[13] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proc. of ACM Conference on Computer Supported Cooperative Work, pp. 175–185. ACM, NY (1994)
[14] Candillier, L., Meyer, F., Boullé, M.: Comparing State-of-the-Art Collaborative Filtering Systems. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 548–562. Springer, Heidelberg (2007)
[15] Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1), 5–53 (2004)
[16] Bobadilla, J., Serradilla, F., Bernal, J.: A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems 23, 520–528 (2010)
[17] Bharadwaj, K.K., Al-Shamri, M.Y.H.: Fuzzy computational models for trust and reputation systems. Electronic Commerce Research and Applications 8(1), 37–47 (2009)

Distributed Load Balancing (DisLB) in Grid Wireless Sensor Network

Swimpy Pahuja, Jaya Chugh, and Ram Kumar

Assistant Professors, Department of Computer Science and Engineering
Rayat Bahra Innovative Institute of Technology and Management, Ganaur
Sonepat, Haryana 131001, India
{swimpy.pahuja,engg.jaya,ram.amu786}@gmail.com

Abstract. As Sensor area network is a vast network with thousands of nodes, the workload increases upto a great extent. This increased workload leads to increase in packet loss and decrease in network lifetime. Even the rate of message failure as well as node failure due to network congestion also increases. In order to avoid such problems and enhance the network lifetime, we are introducing a new algorithm for the purpose of load balancing i.e. Distributed Load Balancing (DisLB). DisLB is a type of load balancing approach in which load has to be distributed on neighboring nodes instead of central nodes in which an All to All communication mode has been assumed.

Keywords: Load balancing, routing protocols, shortest path routing, sensor network.

1 Introduction

Wireless sensor network (WSN) consists of large number of scattered sensing devices which monitor physical or environmental conditions such as temperature, sound, pressure, motion or pollutants. These sensing devices are responsible for data communication. The sensing nodes communicate with each other using low power wireless data routing protocols [1]. Sensor network is a subclass of ad hoc network and varies with it in terms of number of nodes, deployment strategy, failure rate, power etc [2].

1.1 Sensor Network Communication Architecture

The sensor nodes are geographically distributed as shown in Fig. 1. These scattered sensor nodes has data collecting and routing capability. Gathered information has to be routed back to the sink node through multihop communication architecture.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 289–299.
springerlink.com

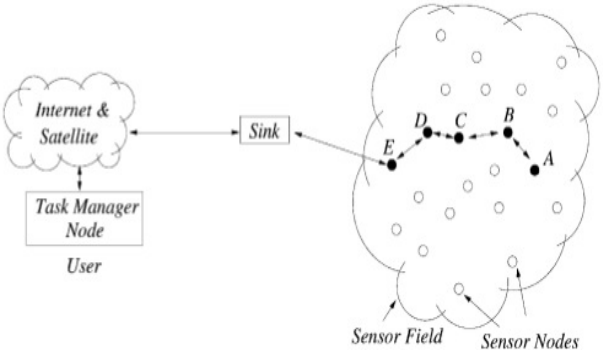


Fig. 1. Sensor nodes scattered in a sensor field [2]

The sink communicates with the task manager node or user through Internet or satellite communication [2].

2 Load Balancing Strategies

The three important parameters need to be taken into account in order to select best among the different load balancing algorithm:

a) who is responsible for making the load balancing decision
b) what information is used to make the load balancing decision
c) where the decision has been made.

Let us consider the load balancing strategies in detail.

2.1 Centralized Load Balancing

In centralized load balancing [3], central node is responsible for calculation of amount of load to be transferred as well as load distribution to other sensor nodes.

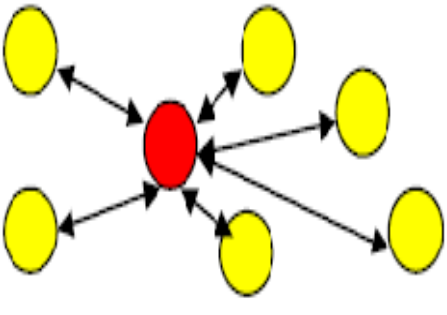


Fig. 2. Centralized load balancing

In fig.2, red coloured circle represents the central node and yellow coloured circles represent the surrounding nodes. As the central node is responsible for task distribution, the whole network goes down if this node fails. However, the policy involves one-to-all as well as all-to-one communication mode.

2.2 Local or Global Load Balancing Strategy

In local load balancing strategy [3], all the sensing devices are divided into heterogeneous groups provided that each group has same performance profile i.e. same computational power. The decision making node would send polling message to each group head which in turn sends the message to its group members demanding the information about the amount of load each node can handle. Based on this information, the decision making node decides to which node the load has to be distributed. In contrast, global load balancing strategy [3] involves the exchange of performance profile information globally among each and every node. The single point of failure and network latency problems of local load balancing technique has been eliminated by this strategy.

2.3 Static or Dynamic Load Balancing

In static load balancing strategy, the load has been assigned to all the nodes in the beginning and there is no variation in the load in between. This type of strategy does not require constant load monitoring and is not well suited for real time applications.

In dynamic load balancing [4][5], load scheduling is done at run time based on the current status of the participating nodes. It is an adaptive policy as the frequent changes in terms of channel traffic; channel delay etc can be entertained. Dynamic methods generally react better to changes in the system state compared to the static methods and as a result, have better performance.

2.4 Sender-Initiated or Receiver –Initiated Load Balancing

In sender-initiated policies [5], congested nodes attempt to move the work to lightly-loaded nodes while in receiver-initiated policies [5], lightly-loaded nodes look for heavily-loaded nodes from which work can be received.

3 Existing System

3.1 Centralized Load Balancing

In this approach as the network traffic pass from some central node network congestion increases which leads to packet loss at that node. If this central node is paralyzed then the whole network gets stuck. Centralized load balancing as shown in fig 3, is used when the nodes requires global information about the state of the

system [6]. Various centralized load balancing strategies such as Random LB, Greedy LB, and Horizontal Vertical Routing etc are available but each has its own limitations as explained in section 3.2. Thus, several authors have proposed different methods of load balancing. As Servetto recently proposed in [7], a routing algorithm which reduces the load on the central node in a single source–single destination communication. It divides the network into expansion and compression phases. Sensor nodes belong to different diagonals of the grid. During expansion phase, the load per node decreases with the increase in number of nodes on diagonal.

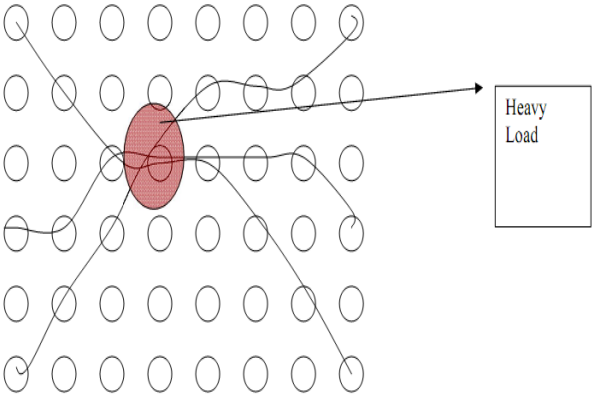


Fig. 3. Grid Wireless Sensor network

During the compression phase, the reverse process is carried out and with the decrease in number of nodes on each diagonal, the load per node increases. In [8], Stefan analyzes the reliability of the system in the case of node failures. The reliability of the system can be increased by providing several paths from source to destination and sending the same packet through this paths. In this it analyze the mechanism through which the trade off between traffic and reliability can be controlled by splitting in K sub packets.

All the approaches defined by various authors are complex as well as deal with central node only. So, there is a great need of a new load balancing strategy which helps to decrease the amount of load on the central node thereby ignoring the central node.

3.2 Limitations of Centralized Load Balancing

Various limitations of centralized load balancing scheme are:

- The scheme leads to increase in congestion in the whole network.
- It leads to creation of hot spots.

- Number of failed messages increases in the network
- It unnecessarily creates heavily loaded nodes.
- Network lifetime decreases
- Packet Loss increases
- Formation of holes

The existing centralized scheme was implemented using Network Simulator version 2 (NS2) [9]. The packet loss as well as packet received rate was taken as a parameter for computing the overall performance of the network.

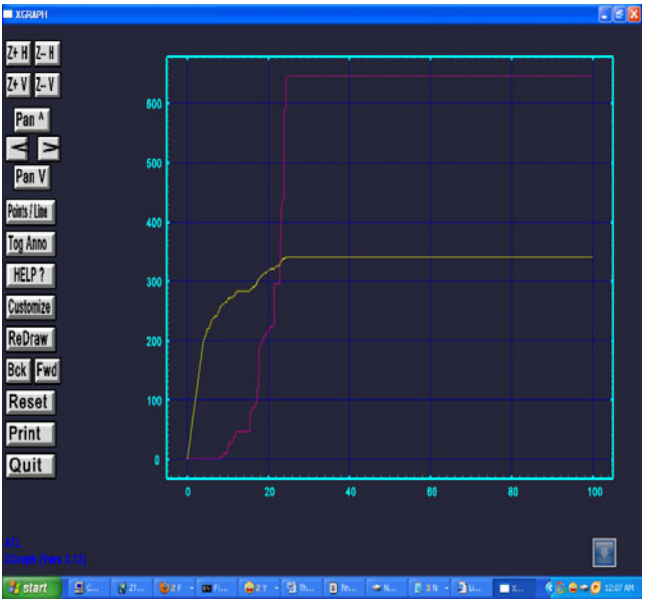


Fig. 4. Packet Loss vs. Packet Received (existing)

These rates of packet received as well as packet loss are then plotted on a graph using Xgraph [9]. In case of existing system i.e. centralized load balancing, packet received is more as compared to packet loss as shown in fig 4. However, the amount of packet received can be increased at a greater rate using distributed load balancing. To overcome the bottleneck due to congestion at central node we are proposing an approach to distribute the load on all the neighboring nodes.

4 Proposed Scheme

Wireless sensor networks are increasingly gaining importance in various time critical applications. Generally, greedy approach is followed by the sensing devices for data transmission. As a sensor node has fix power to work on, when

the load exceeds its threshold value, it starts behaving abnormally and starts packet as well as energy loss while the corner nodes are underutilized. This uneven load distribution results in heavily loaded nodes to discharge faster as compared to others. This causes few over-utilized nodes which fail and result in formation of holes in network, resulting in increase of failed messages in the network. Hence, a new load balancing strategy which is termed as Distributed Load Balancing (DisLB) has been proposed so that the underutilized nodes can be better utilized and the workload on over-utilized nodes can be decreased upto some extent. In this, the load is distributed on all those neighbouring nodes which are in coverage area of central node.

4.1 Basic Idea

The main objective behind this research is to develop a center-compromised routing algorithm for congested grid network as well as to share the load among the under-utilized nodes so that inbound internet protocol (IP) traffic can be distributed across multiple sensing nodes.

4.2 Objectives

The main objectives behind this research are as follows:

- To increase energy efficiency of whole network.
- To increase the network lifetime.
- To Decrease the congestion in whole network.
- To minimize the formation of holes.
- To minimize the packet loss at central node.
- To increase the packet received rate.
- To Maintain Network Connectivity.

4.3 Proposed Algorithm

A grid network has been considered in the algorithm where each sensor node can transfer data to any of the other sensor node. The proposed algorithm has been divided into two parts namely Node Detection and Load Distribution. The algorithm makes use of two data structures namely FIFO queue and list. The queue has been used for the purpose of detection of under-utilized nodes while the list has been used for the purpose of storing the nodes present in shortest path to designated node.

Various assumptions for DisLB are as follows:

- Initially, the workload has been assigned to all the sensing nodes present in the network.
- The amount of data packets to be distributed should be known in advance.
- All the threshold values should be set first.

```
/* Node Detection Algorithm*/
/*S is the Source Node and D represents the Destination Node */
  Step 1  Find the Shortest Path between S to D using
          Dijkstra Algorithm.
  Step 2  Maintain a List1 L1 ( I0,I1,....In) of all the  routing nodes present on the shortest path.
  Step 3  for each (Node In in L1)
          {
              If (Load(In)>Threshold)
              Put In in List2 L2 (F1,F2,....Fm)
          }
  Step 4   X= MAX (F1,F2,....Fm)
  Step 5   Set X as Central Node
  Step 6   Find Compromising Nodes of X and put them in queue called CNQ (C0 to Cq)
```

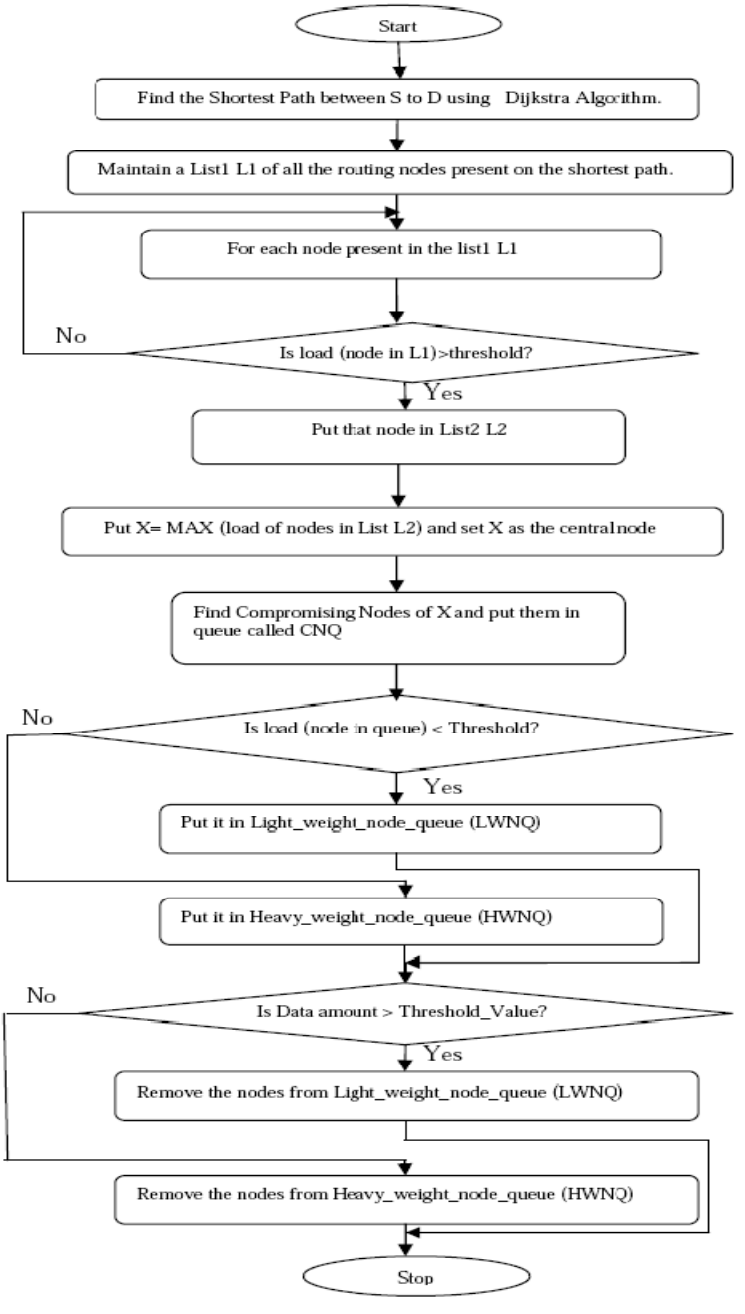
The algorithm requires two threshold values namely Threshold (This is the maximum value of load assigned to the particular node and Threshold_value is the maximum amount of data to be transmitted over the network

```
/*Load Distribution Algorithm*/
If(Load(Cq)< Threshold)     /*Quick Sort Algorithm*/
Put Cq in Light_weight_node_queue LWNQ
Else
Put Cq in Heavy_weight_node_queue HWNQ
If( Amount of data> Threshold_Value)
Remove the nodes from Light_weight_node_queue and distribute the load on
these nodes
Else
Remove the nodes from Heavy_weight_node_queue
and distribute the load on these nodes
```

.

4.4 Proposed Flowchart

The flowchart of proposed algorithm is as follows:



5 Results

The proposed scheme was implemented using Network Simulator version 2(NS2) [10]. The amount of packets received were recorded for both proposed as well as existing scheme and the graph showing the comparison between both schemes was plotted using Xgraph as shown in fig 5.

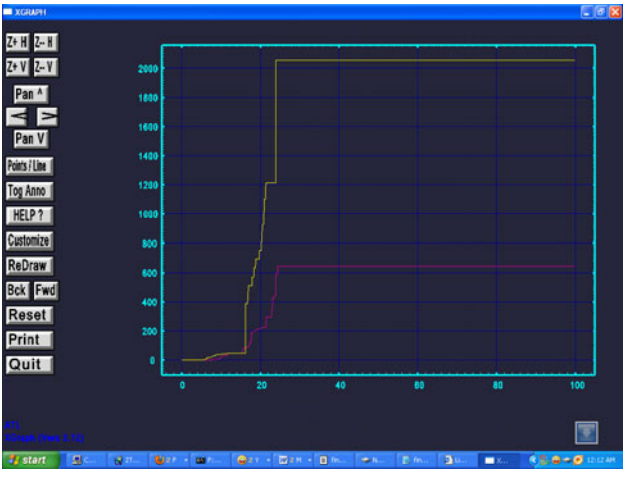


Fig. 5. Packet Received (proposed vs existing)

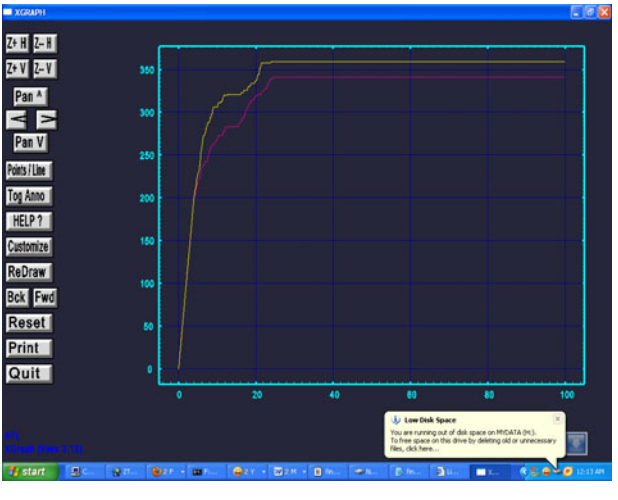


Fig. 6. Packet Loss (existing vs proposed)

In this, yellow coloured line represents proposed system packet received rate which is 2100 while pink coloured line represents existing system packet received rate which is 605. Thus, the rate of packet reception is higher in case of proposed system. Similarly, the rate of loss of packets is also measured for both the schemes and the corresponding comparison graph was plotted as shown in fig 6.

In this, yellow coloured line represents existing system packet loss which is 355 while pink coloured line represents proposed system packet loss which is 330. Therefore, the packet loss in case of proposed algorithm is very less as compared to existing centralized algorithm.

6 Conclusion

The Distributed load Balancing (DisLB) algorithm has been proposed as well as implemented on network simulator. It has overcome the drawbacks of existing centralized load balancing strategy. Finally, the workload on central node has been decreased by routing the data packets from surrounding nodes or we can say neighbouring nodes. Due to reduction in workload of central node, the rate of packet reception has increased and the rate of packet loss has decreased as compared to existing centralized algorithm. This has lead to the decrease in node failure rate as well as in network congestion thereby increasing the network lifetime.

References

[1] Lewis, F.L.: Wireless Sensor Networks Smart Environments: Technologies, Protocols, and Applications. In: Cook, D.J., Das, S.K. (eds.), pp. 1–18. John Wiley, New York (2004)

[2] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks. Proc. of the IEEE Communications Magazine 40(8), 102–114 (2002)

[3] Zaki, M., Li, W., Parthasarathy, S.: Customized Dynamic Load Balancing for a Network of Workstations. In: Proc. of HPDC 1996 (1996)

[4] Lee, B.: Dynamic Load Balancing in a Message Passing Virtual Parallel Machine. Technical Report, Division of Computer Engineering, School of Applied Science, Nanyang Technological University, Singapore (1995)

[5] Dandamudi, S.: Sensitivity Evaluation of Dynamic Load Sharing in Distributed Systems, Technical Report TR 97 -12, Carleton University, Ottawa, Canada

[6] Cortes, A., Ripoll, A., Senar, M., Luque, E.: Performance comparison of dynamic load-balancing strategies for distributed computing. In: Proc. of the 32nd Annual Hawaii International Conference on System Sciences (1999)

[7] Servetto, S.D., Barrenechea, G.: Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks. In: Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, pp. 12–21 (September 2002)

[8] Dulman, S., Nieberg, T., Wu, J., Havinga, P.: Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks. In: WCNC Workshop, New Orleans, vol. 3, pp. 1918–1922 (March 2003)
[9] Law, A., Kelton, W.: Simulation Modeling and Analysis. McGraw-Hill (2000)
[10] Ns Manual, http://www.isi.edu/nsnam/ns/ns-documentation.html

Development of an Automated Examination Seating Arrangement Generation System Using Music Inspired Harmony Search Algorithm

Arnav Acharyya, Arpan Losalka, Pravir Singh Gupta, and Ganapati Panda

Indian Institute of Technology Bhubaneswar, India
{arnav.acharyya,arpanlosalka,pravir.singh.gupta}@gmail.com
gpanda@iitbbs.ac.in

Abstract. This paper addresses the solution of examination seating arrangement problem using the metaheuristic music inspired Harmony Search algorithm. The objective function has been defined keeping the constraints of the problem in mind. The algorithm has been modified to suit the needs of the problem at hand, keeping the basic structure unchanged. The results obtained show an optimized seating arrangement, where none of the students appearing for the same subject have been allotted adjacent seats.

1 Introduction

The problem at hand is the 'Seating arrangement of students from different departments in an examination'. The problem involves combinatorial optimization of the seats of the students so that the seating arrangement is legitimate enough to be used in examinations. In examinations we intend to design such a seating arrangement which efficiently decreases the scope of plagiarism among the students which is only plausible to happen among the students from the same department taking the exam on the same subject. Now, to generate a feasible seating arrangement, firstly some specific hard constraints are to be satisfied, most of them being technical and physical constraints. In addition to that, soft constraints should also be considered which determine the efficiency of the seating arrangement in reducing plagiarism or exchange of answers among the students to fit the criterion of being used in exams effectively. If the soft constraints are satisfied the solution may be reckoned to be an optimal one. So from the point of view of minimizing the soft constraints-violations the 'Seating Arrangement problem' can be considered as an Optimization problem.

Harmony search (HS) is a metaheuristic optimization algorithm which mimics the improvisation process of musicians. The advantages of using HS algorithm over other optimization algorithms are: (a) As the HS algorithm uses stochastic random searches, derivative information is not necessary unlike in Particle Swarm Optimization (PSO) where the velocity of the vectors needs to be initialized by magnitude

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 301–309.
springerlink.com

and direction (direction being towards the most fit vector usually). (b) The HS algorithm generates a new vector considering all the vectors in the population (Harmony Memory) unlike Genetic algorithm (GA) which employs methods (like crossover) to generate a new vector from only two parent vectors. This algorithm has proved to be efficient for other optimization problems such as tour planning problem [3], parameter estimation of the nonlinear Muskingum model [4], vehicle routing problem [5], timetabling problem [6] and others.

2 Problem Statement

The aim is to design an optimized seating arrangement for examination where the students appearing for the same subject are to be separated as far as possible using Harmony Search algorithm. Here certain hard constraints are considered conforming to which the possible seating arrangements are to be generated and a soft constraint which enhances the quality of the seating arrangement to be used in the examination.

The various parameters used in the problem definition are the set of students $\mathbf{S} = \{s_i\}$ with $i = 1, 2, \ldots, |S|$, the set of subjects $\mathbf{N} = \{n_j\}$ with $j = 1, 2, \ldots, |N|$ and the set of seats $\mathbf{P} = \{p_{rc}\}$ with $r = 1, 2, \ldots, |R|$, $c = 1, 2, \ldots, |C|$, where $|S|$,$|N|$,$|R|$ and $|C|$ are the total number of students, total number of subjects, total number of rows of seats and total number of columns of seats respectively.

Let a indicate the correspondence between students and different subjects

$$a(s_i, n_j) \begin{cases} 1 & \text{if student } s_i \text{ is enrolled in } n_j \text{ subject} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and b denote the correspondence between students and different seats.

$$b(s_i, p_{rc}) \begin{cases} 1 & \text{if student } s_i \text{ sits at } p_{rc} \text{ seat} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The row number and column number of the seat of a particular student s_i are given by $Row(s_i) = \sum_{r=1}^{|R|} \sum_{c=1}^{|C|} b(s_i, p_{rc}) \times r$ and $Col(s_i) = \sum_{r=1}^{|R|} \sum_{c=1}^{|C|} b(s_i, p_{rc}) \times c$.

A set of four hard constraints are used in defining the problem as follows:

- **H1:** Each student must be enrolled in only one subject
  i.e. $\forall_{s \varepsilon S} \sum_{j=1}^{|N|} a(s_i, n_j) = 1$.
- **H2:** Each subject must have at least one student enrolled in it
  i.e. $\forall_{n \varepsilon N} \sum_{i=1}^{|S|} a(s_i, n_j) \geq 1$.
  H1 and **H2** imply that $|N| \leq |S|$ where $|N| = |S|$ when each subject has only one student enrolled.
- **H3:** More than one student cannot sit at a particular seat
  i.e. $\forall_{p \varepsilon P} \sum_{i=1}^{|S|} a(s_i, p_{rc}) \leq 1$.

- **H4:** More than one student cannot sit at a particular seat
  i.e. $\forall_{s \varepsilon S} \sum_{r=1}^{|R|} \sum_{c=1}^{|C|} b(s_i, p_{rc}) = 1$.

Here we define a new set D which contains the square of the distance between all possible pairs of students enrolled in the same subject.
Now, it is clear that,

$$\sum_{k=1}^{|N|} a(s_i, n_k) a(s_j, n_k) = \begin{cases} 1 & \text{if } s_i \text{ and } s_j \text{ are from the same department} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The set D is given by, $D = \{(Row(s_i) - Row(s_j))^2 + (Col(s_i) - Col(s_j))^2$

$$\left| 1 \le i, j \le |S|, \{s_i, s_j\} \in S, i \ne j, \sum_{k=1}^{|N|} a(s_i, n_k) a(s_j, n_k) = 1 \right\} \tag{4}$$

Elements of D are represented as d_m where $m = 1, 2, 3, \ldots, |D|$.
We define d_{min} as,

$$d_{min} = min\{D\} \tag{5}$$

The soft constraint to be implemented is given as:

- **S1:** No two students enrolled in the same subject must be seated adjacent (row wise or column wise adjacent) to each other i.e. $d_{min} \ge 2$

We define a variable c which indicates the correspondence between pairwise distance of particular two students from the same department and the overall minimum distance.

$$c(d_m) \begin{cases} 1 & \text{if } d_m = d_{min} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Another variable M gives the frequency of occurrence of d_{min} in set D.

$$M = \sum_{m=1}^{|D|} c(d_m) \tag{7}$$

The problem can be formulated as a optimization problem, with an objective of maximizing

$$f = d_{min} + \frac{0.9}{M} \tag{8}$$

Hence, the function f is also referred to as the **objective** or **fitness function**

N.B.: All the indices i, j, k, m, r, c are natural numbers.

3 Basic Harmony Search Algorithm

The Harmony Search algorithm mimics the improvisation process of musicians. When a musician plays a particular tune, he has the option of improvising the tune by either varying the tune slightly from the original one or by playing a completely random tone. Else, he can play the exact tune from his memory. The process is given shape using mathematics to formulate the Harmony Search Algorithm.

The parameters of Harmony Search algorithm are **`HMS`** - Harmony Memory Size (usually 10 to 100), **`HMCR`** - Harmony Memory Consideration Rate (usually .7 to .99), **`PAR`** - Pitch Adjustment Rate (usually 0.1 to 0.5), δ - Difference between two adjacent values in case of discrete variable, **`FW`** - Fret Width (in case of continuous variable).

The steps of the algorithm are as follows:

1. HMS number of random vectors are generated and stored in the 'Harmony Memory'.
2. A fitness function is defined suitably with respect to the problem at hand. The fitness of the vectors in the Harmony Memory are calculated.
3. A new vector is generated considering the following rules for each component of the vector:
   - With probability HMCR, the component is picked from a random vector of the Harmony Memory.
   - With probability 1-HMCR, the component is picked at random within the allowed range.
4. If the generated component is chosen from the Harmony Memory,
   - With probability PAR, the value of the component is changed by a small amount δ (discrete variable) or a random fraction of FW (continuous variable).
   - With probability (1-PAR), the component is left unchanged.
5. The fitness of the generated vector is calculated.
6. If the generated vector is more fit than the least fit vector of the harmony memory, then the least fit vector is substituted by the generated vector. Else, the harmony memory is left unaltered.
7. Steps 3 and 4 are repeated until a fitness criterion is satisfied.

The correspondence between the musical improvisation process and the HS algorithm is as follows: (a) a tune corresponds to a vector (b) different tones of the tune correspond to the components of the vector (c) the memory of the musician corresponds to Harmony Memory (d) the musician playing a tone from his memory is equivalent harmony memory consideration (e) the musician altering the tone slightly is equivalent to pitch adjustment (f) the musician playing a completely random tone is equivalent to random generation within the allowed range (g) the aesthetics of the listener is equivalent to the fitness function.

4 Proposed Approach

The equivalence between the Harmony Search algorithm and the problem can be drawn by setting vectors analogous to seating arrangements and components of vectors analogous to seats of the students. The position of the component in the vector specifies the student. The harmony memory is given by

$$\mathbf{HM} = \begin{bmatrix} x_1^1 & \cdots & x_{|S|}^1 & | & f(\mathbf{x}^1) \\ \vdots & \ddots & \vdots & | & \vdots \\ x_1^{hms} & \cdots & x_{|S|}^{hms} & | & f(\mathbf{x}^{hms}) \end{bmatrix} \tag{9}$$

x_i^k denoting the seat of the ith student in the kth seating arrangement and $f(\mathbf{x}^k)$ denoting the fitness of the kth seating arrangement in harmony memory.

The fitness function (eq. 8) takes into account the square of the minimum distance between students appearing for the same subject in the examination given by d_{min} (from eq. 5) and also the frequency of occurrence of d_{min} given by M (from eq.7). Therefore, maximizing the fitness function tends to increase the minimum distance (d_{min}) between the students enrolled in the same subject and simultaneously tends to decrease the number of pairs of students enrolled in the same subject with d_{min} separation. The constant '0.9' ensures that a seating arrangement with a greater d_{min} is always favored over a seating arrangement with smaller d_{min}. The frequency of occurrence comes into play whenever the d_{min} of two seating arrangements are equal. Then the one with lesser frequency is favored. Hence maximizing the objective function primarily minimizes the soft constraint violations and then enhances the seating arrangement beyond the soft constraint.

The algorithm had to be modified slightly for implementation in this problem. This is because in harmony search algorithm, the components of the vector are independent of each other. However, in the problem at hand, the seat of one student will depend on the seats assigned to the other students for a particular seating arrangement, because two or more students cannot occupy the same seat (hard constraint). Hence, (1) during random selection as per the algorithm, the components already generated are removed from the universal set, i.e. the set of all possible seats. Also (2) during harmony memory consideration, the seats which have already been allotted are not taken into consideration. In the case where all the seats in the harmony memory corresponding to the position of the component being generated have already been allotted, random selection has been used. Finally (3) during pitch adjustment, in case the seat after adjustment turns out to be already allotted, then the unadjusted seat is swapped with the seat of the student who initially occupied the seat obtained after adjustment.

Algorithm 1. Pseudo-code for the Harmony Search Algorithm adopted

```
Generate HMS number of feasible seating arrangements (Initialize HM)
Compute the fitness value of each seating arrangement in HM
Arrange the seating arrangements in descending order of their fitness
A new seating arrangement (x_1^{hms+1}, x_2^{hms+1}, ..., x_{|S|}^{hms+1}) is created as follows
repeat
  for Each student s_i do
    Generate a uniform random number in (0, 1)
    if Generated random number < HMCR then
      if The seats in i^th column of HM are pre-occupied in the arrangement then
        Randomly assign a non-occupied seat
      else
        Randomly choose a non-occupied seat from i^th column of HM
      end if
    else
      Randomly assign any non-occupied seat
    end if
    if Seat is chosen from Harmony Memory then
      Generate a uniform random number in (0,1)
      if Generated random number < PAR then
        The seat is stored in a variable
        The seat is randomly adjusted to any of the eight possible adjacent seats
        if The adjusted seat coincides with a seat pre-occupied by student s_j then
          Replace the seat of s_j with the seat of s_i before adjustment
          The adjusted seat is assigned to s_i
        else
          The adjusted seat is assigned to s_i keeping other seats unchanged
        end if
      end if
    end if
  end for
  Calculate the fitness value of the newly formed seating arrangement
  if Fitness of new arrangement > fitness of least-fit arrangement in HM then
    Include the new seating arrangement in HM and discard the least-fit one
  end if
until Soft constraint is satisfied
```

5 Simulation Study

The problem has been simulated according to the proposed approach and verified for generalised cases with seat matrix, number of subjects and number of students per subject as variables. To cite an example we consider a typical case of 12×8 seat

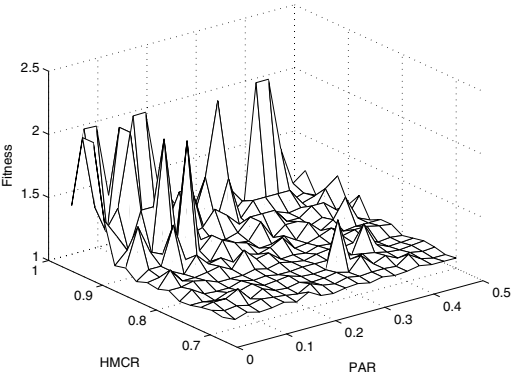


Fig. 1. Plot of average fitness val versus HMCR and PAR for a fix number of iterations

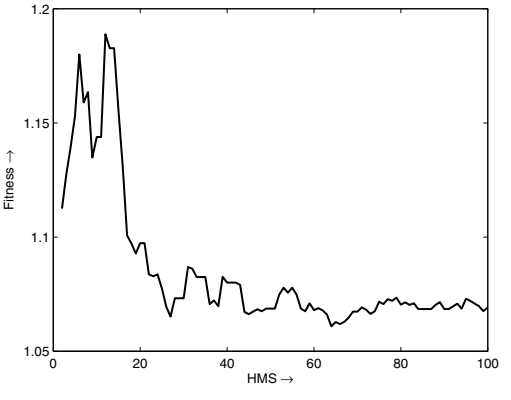


Fig. 2. Plot of average fitness val versus HMS for a fixed number of erations

matrix and three subjects with equal number of students enrolled in each subject i.e. 32 students per subject. The optimum values of the parameters HMS, HMCR and PAR are found out for this particular problem statement by varying the parameter values and comparing their corresponding fitness values after a fixed number of iterations.

Fig. 1 shows the fitness values of the parameters HMCR and PAR averaged over different values of HMS for a fixed number of iterations.

Fig. 2 shows the fitness values of the parameter HMS averaged over different values of HMCR and PAR for a fixed number of iterations. The optimum values of HMS, HMCR and PAR for this problem statement are obtained from the peaks of the plots to be 12, 0.94 and 0.2 respectively.

The optimum values of the parameters found out are employed in simulating the example case mentioned earlier. Fig. 3 shows the simulation output of the mentioned example case i.e. an optimized seating arrangement for a 12×8 seat matrix with three subjects and 32 number of students for each subject. As expected, no two students taking the examination on the same subject are seated row wise or column

wise adjacent to each other. In Fig. 4 the solid line shows the fitness of the most-fit seating arrangement in harmony memory and the dotted line shows the fitness of the least-fit seating arrangement in harmony memory. From the definition of the fitness function it is clear that it takes discrete values as d_{min} and M take only integral values. Hence the fitness of both the least-fit and most-fit member of the harmony memory increase in a discrete fashion to converge into a near optimal solution.

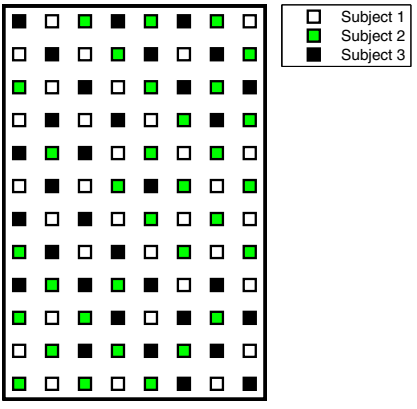


Fig. 3. Optimized seat matrix in an examination with 3 different subjects and a (12×8) matrix of seats using the proposed approach

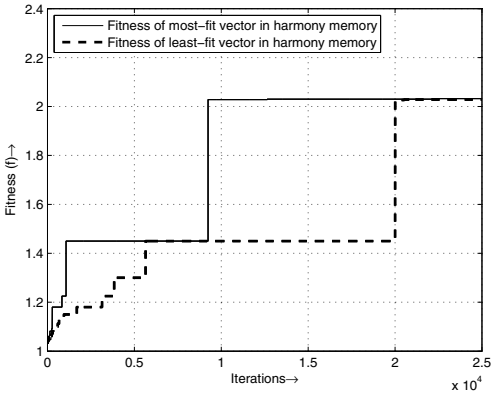


Fig. 4. Plot of fitness value versus iteration showing the gradual increase of the fitness

6 Conclusion and Future Scope of Research

This paper introduces the optimization of an examination seating arrangement, which will be unarguably very useful in practice. The scope of its use may be extended beyond examinations to room allotment in hostels, seat allotment in classrooms, etc. where placing two incompatible persons (in aspects relevant to the problem) adjacent to each other may render counterproductive results. Here, iteration of the optimization process was terminated if the soft constraint was satisfied. However, continuing to iterate the whole process even after the soft constraint is satisfied will lead to further increase in efficiency of the seating arrangement. Also the problem can be further generalized through the introduction of more soft constraints

such as seating restrictions for certain students, customization of the available seating arrangement options etc. Further scope of research in this area lies in the comparison of the proposed approach using Harmony Search with other metaheuristic algorithms and analyze their performances in terms of computational time, memory usage, etc.

References

1. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A New Heuristic Optimization Algorithm: Harmony Search. SIMULATON (2001), doi: 10.1177/003754970107600201
2. Geem, Z.W. (ed.): Music-Inspired Harmony Search Algorithm. Springer, Heidelberg (2009)
3. Geem, Z.W., Tseng, C.-L., Park, Y.-J.: Harmony search for generalized orienteering problem: Best touring in china. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 741–750. Springer, Heidelberg (2005)
4. Kim, J.H., Geem, Z.W., Kim, E.S.: Parameter estimation of the nonlinear Muskingum model using harmony search. J. Am. Water Resour. Assoc. 37(5), 1131–1138 (2001)
5. Geem, Z.W., Lee, K.S., Park, Y.: Application of Harmony Search to Vehicle Routing. American Journal of Applied Sciences 2(12), 1552–1557 (2005)
6. Mohammed, A.A.-B., Khader, A.T., Gani, T.A.: A harmony search algorithm for university course timetabling. In: Mohammed, A.A.-B., Khader, A.T. (eds.) Annals of Operations Research, pp. 1–29. Springer, Netherlands (2010)
7. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 188(2), 1567–1579 (2007)
8. Kang, S.L., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. Comput. Struct. 82(9-10), 781–798 (2004)
9. Soza, L., Becerra, C.R., Riff, M.C., Coello, C.A.C.: Solving timetabling problems using a cultural algorithm. Applied Soft Computing 11(1), 337–344 (2011)

Differential Evolution for Data Envelopment Analysis

Pravesh Kumar, Sandeep Kumar Mogha, and Millie Pant

Indian Institute of Technology, Roorkee, India
{praveshtomariitr,moghadma}@gmail.com, millifpt@iitr.ernet.in

Abstract. Differential Evolution (DE) is a versatile technique used for solving a wide range of problems occurring in various fields. In the present study, we propose a novel application of simple differential evolution (SDE), for data envelopment analysis (DEA). The mathematical model of the problem is constrained in nature for which Pareto ranking method is embedded in SDE.

Keywords: Differential Evolution, Data Envelopment Analysis, Constrained Optimization Problem.

1 Introduction

Data envelopment analysis (DEA) is a decision support tool developed by Charnes et al. [1] to assist decision makers in comparing the performance of a group of candidate solutions and selecting the best one [2]. It is a linear programming based approach used to measure and analyzes efficiency of different entities such as productive and non-productive, private and public, profit and non-profit seeking firms. It is non-parametric approach to calculate the efficiency of each unit in the sample by making a linear program [3].

Differential evolution algorithm (DE) was proposed by Stron and Price in 1995 [9]. It is a population set based evolutionary algorithm to solve global optimization problem. DE does not depend on the nature of the objective function i.e. no matter that function is continuous or discontinuous, differentiable or non-differentiable, convex or non-convex etc. During some past years DE has been applied in the many real life application of engineering and science field [10], chemical engineering [11], engineering design problem [12], neural network [13], multi-objective [14], and so on.

The purpose of the paper is to use Simple Differential Evolution algorithm (SDE) in data envelopment analysis (DEA) to measure and analyze the relative sales efficiency of 8 hospital and medical research centers of Delhi (India) on the

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 311–319.
springerlink.com

basis of experimental data that is collected from these centers for the period 2009-2010.

The rest of the paper is organized as follows: In section 2 brief discussion of data envelopment analysis (DEA) is given. Simple Differential evolution (SDE) is explained in section 3. In section 4 experimental data, parameter settings for SDE and results are given; finally the paper concludes with section 5.

2 Data Envelopment Analysis (DEA)

DEA is a linear programming based technique for measuring the efficiency of Decision-Making Unit (DMU) on the basis of multiple inputs and outputs [4]. DMU can include manufacturing units, departments of big organizations such as universities, schools, hospitals, power plants, police stations, tax offices, prisons, a set of firms etc [5].

Most of the DMU are non-profit organizations, where the measurement of performance efficiency is difficult. The performance of DMU is evaluated in DEA using the concept of efficiency or productivity, which the ratio of weights sum of outputs to the weights sum of inputs [6] i.e

$$effciency = \frac{weights\ sum\ of\ \ outputs}{weight\ sum\ of\ \ inputs} \tag{1}$$

2.1 Mathematical Formulation of DEA

The two basic DEA models are the CCR (Charnes, Cooper and Rhodes) model [1] and the BCC (Banker, Charnes and Cooper) model [7]. These two models differentiate on the returns to scale assumed. The former assumes constant returns-to-scale whereas the latter assumes variable returns-to-scale [4]. A comprehensive text with the DEA model variations and applications can be found in [8]. In the present study we use CCR model which is defined below:

Assume that there are N DMUs and each unit have I input and O outputs then the efficiency of m^{th} unit is obtained by solving the following model which is proposed by Charnes et al [1].

$$Max\ E_m = \frac{\sum_{k=1}^{O} w_k output_{k,m}}{\sum_{l=1}^{I} z_l input_{l,m}} \tag{2}$$

s.t.

$$0 \le \frac{\sum_{k=1}^{O} w_k output_{k,n}}{\sum_{l=1}^{I} z_l input_{l,n}} \le 1; n = 1,2,...,m,...N$$

$$w_k, z_l \ge 0; \forall k, l$$

where

E_m is the efficiency of the m^{th} DMU

k=1 to O, l=1 to I and n=1 to N.

$output_{k,m}$ is the k^{th} output of the m^{th} DMU and w_k is weight of output $output_{k,m}$

$input_{l,m}$ is the l^{th} input of m^{th} DMU and z_l is the weight of $input_{l,m}$

$output_{k,n}$ and $input_{l,n}$ are the k^{th} output and l^{th} input respectively of the n^{th} DMU, where n=1, 2, …m,..., N

The fractional program shown in Equ-2 can be converted in a linear program which is shown in Equ-3

$$\begin{aligned} &Max\ E_m = \sum_{k=1}^{O} w_k output_{k,m} \\ &s.t. \\ &\sum_{l=1}^{I} z_l input_{l,m} = 1 \\ &\sum_{k=1}^{O} w_k output_{k,n} - \sum_{l=1}^{I} z_l input_{l,n} \le 0, \forall n \\ &w_k, z_l \ge 0; \forall k, l \end{aligned} \tag{3}$$

To calculate the efficiency score of each DMU we run the above program run N times. A DMU is considered efficient if the efficiency score is 1 otherwise it is considered as inefficient.

3 Simple Differential Evolution (SDE)

Simple DE algorithm is a kind of evolutionary algorithm, used to optimize the functions. In this paper SDE refers to the DE/rand/1/bin scheme [15]. SDE starts with a randomly generated set of solutions when we have not any preliminary knowledge about the solution space. This set of solutions is term as population.

Let $P=\{X_i^G$, i=1,2,...$NP\}$be the population at any generation G where X_i^G $=(x_{1,i}^G, x_{2,i}^G..., x_{D,i}^G)$ is D-dimensional solution vector and NP is population size. For simple DE (*DE/rand/1/bin*), the mutation, crossover, and selection operators are defined as follows:

i. **Mutation:** For each target vector X_i^G select three different vector say X_{r1}^G, X_{r2}^G and X_{r3}^G from the population P such that $r1 \neq r2 \neq r3 \neq i$ then the mutant vector $V_i^G = (v_{1,i}^G, v_{2,i}^G \ldots, v_{D,i}^G)$ is defined as:

$$V_i^G = X_{r1}^G + F(X_{r2}^G - X_{r3}^G) \tag{4}$$

Here, F is a real and constant factor having value between [0, 2] and controls the amplification of differential variation $(X_{r2}^G - X_{r3}^G)$.

ii. **Crossover:** Perform crossover operation to create a trial vector $U_i^G = (u_{1,i}^G, u_{2,i}^G \ldots, u_{D,i}^G)$ as

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, if\ Cr < rand(01) \forall j = jrand \\ x_{j,i}^G \quad otherwise \end{cases} \tag{5}$$

rand(0, 1) is uniform random number between 0 and 1; *Cr* is the crossover constant takes values in the range [0, 1] and *jrand* ∈ 1, 2,..., *D;* is the randomly chosen index.

iii. **Selection:** During the selection operation we generate a new population $Q = \{X_i^{G+1}, i=1,2,...NP\}$ for next generation $G+1$ by choosing the best vector between trial vector and target vector.

$$X_i^{G+1} = \begin{cases} U_i^G, if\ f(U_i^G) < f(X_i^G) \\ X_i^G \quad otherwise \end{cases} \tag{6}$$

3.1 Pseudo Code of SDE Algorithm

1	Begin
2	Generate uniformly distribution random population $P = \{X_i^G, i=1,2,...NP\}$. $X_i^G = X_{lower} + (X_{upper} - X_{lower}) * rand(0,1)$, where $i = 1, 2,..,NP$
3	Evaluate $f(X_i^G)$
4	While (Termination criteria is met)
5	{
6	For i=1:NP
7	{
8	Select three vectors X_{r1}^G, X_{r2}^G and X_{r3}^G such that $r1 \neq r2 \neq r3 \neq i$
9	Perform mutation operation as defined by Equation-4
10	Perform crossover operation as defined by Equation-5
11	Evaluate $f(U_i^G)$
12	Select fittest vector from X_i^G and U_i^G to the population of next generation by using Equation-6

```
13              }
14          Generate new population Q={X_i^{G+1} , i=1,2,...NP}
15        } /* end while loop*/
16      END
```

3.2 Constraint Handling

For the constraint problems various methods have been suggested in literature. A survey of different methods for constraint handling can be found in [16] and [17]. In this paper Pareto-Ranking method is used for handling the constraints [18].

4 Experimental Setting and Results

In this section the experimental data for hospitals and medical centers, parameter setting for SDE, and simulated results are given.

4.1 Experimental Data and DEA Model of Hospitals and Medical Research Centers

Table 1. Name of 8 different hospital and medical research institutes of New Delhi, India

DMU	Name
1	A D S Diagnostic Ltd
2	Alps Hospital Ltd
3	Escorts Heart Institute & Research Centre Ltd
4	Escorts Hospital & Research Centre Ltd
5	Fortis Healthcare (India) Ltd
6	HSCC (India) Ltd
7	Indraprastha Medical Corporation Ltd
8	Max Healthcare Institute Ltd

Table 2. Financial Data of 8 different Hospital of New Delhi, India that is taken from [19]

DMUs (*n*)	Annual Earning/ Total Sales (*TS*)	Net Fixed Capital /Net Block (*NB*)	Power and Fuel (*PW*)l	Wages and salaries (*WS*)	Capital Employed (*CE*)	Total Current Assets (*TA*)
1	3.32	1.62	0.04	0.46	3.86	3.22
2	38.86	39.95	1.39	8.48	24.22	23.18
3	275.87	59.76	8.47	73.28	255.06	4.27
4	74.53	92.64	2.15	15.8	48.64	26.35
5	213.14	72.21	4.35	39.07	2,824.46	1,355.19
6	21.38	6.32	0.29	12.97	72.39	464.34
7	424.43	205.87	13.06	81.9	164.99	90.66
8	339.06	228.24	7.58	82.71	864.97	598.24

On the basis of the above data the DEA model of m^{th} DMU will be as follows:

$$\begin{aligned} & Max TS_m, \\ & s.t. \\ & z_1 NB_m + z_2 PW_m + z_3 WS_m + z_4 CE_m + z_5 TA_m = 1 \\ & wTS_n - (z_1 NB_n + z_2 PW_n + z_3 WS_n + z_4 CE_n + z_5 TA_n) \le 0 \\ & \forall\ n = 1, 2.., m, ...8 \end{aligned} \tag{7}$$

4.2 Parameter Setting for Differential Evolution Algorithm

In this paper we have applied SDE to solve the DEA model. The parameter settings for SDE are given in Table-3.

Table 3. Parameter setting for SDE

Parameter	Setting
Population Size (*NP*)	100
Scaling Factor (*F*)	0.5
Crossover Probability (*Cr*)	0.5
Max Iteration	3000

The program is implemented is DEV C++ and all the uniform random number is generated using the inbuilt function *rand ( )* in DEV C++. The fitness value is taken as the average fitness value in 30 runs and the program is terminate when reach to Max-Iteration.

4.3 Results and Discussions

In the Table-4 results of all DMUs are given. From this Table we can see that for hospital no 1, 3, 5, 6 and 7, the efficiency score is 1 so these hospitals are assumed to be 100% efficient while efficiency score for hospital no 2, 4 and 8 are 0.8377, 0.9375 and 0.6656 respectively. So these hospitals are not as efficient and among these, hospital no-8 is probably the most inefficient in comparison to all other hospitals.

Figure-1 gives the histogram of efficiency of all hospitals.

Table 4. Average efficiency and weighted of all hospitals and medical centres in 30 runs

		value of input and output weight					
DMU	**E**	**w**	**z1**	**z2**	**z3**	**z4**	**z5**
1	**1.00**	0.3012	0.2291	0.4608	0.8889	0.0198	0.0388
2	0.8377	0.0215	8.08e-006	0.0096	0.0960	0.0068	0.0002
3	**1.00**	0.0036	8.28e-005	0.0074	0.0003	0.0033	0.0129
4	0.9375	0.0125	2.65e-005	0.3230	0.0026	0.0019	0.0062
5	**1.00**	0.0046	0.0059	0.0012	0.0127	1.21e-005	2.36e-005
6	**1.00**	0.0469	0.0804	0.9225	0.0007	0.0017	0.0001
7	**1.00**	0.0023	0.0001	0.0002	0.0034	0.0023	0.0033
8	0.6656	0.0019	0.0028	0.0386	0.0005	5.27e-006	9.85e-006

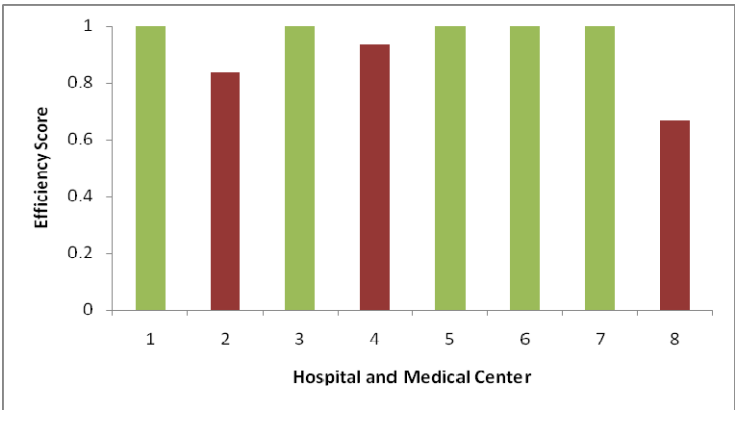


Fig. 1. Histogram of all hospitals and medical centers with their efficiency score

5 Conclusions

Differential Evolution has proven its versatility in solving a wide range of real life application problems. In the present study, Simple Differential Evolution (SDE), the basic version of DE is employed for dealing with the well known problem of Data Envelopment Analysis (DEA). Although, efficient softwares are available for DEA in the present study we consider the mathematical model as a constrained benchmark to which SDE is applied. Numerical results validate the efficiency of SDE for dealing with such problems.

References

1. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2(6), 429–444 (1978)
2. David, T.W., Luis, F.O., Gareth, P.H.: Modified GA and data envelopment analysis for multistage distribution network expansion planning under uncertainty. IEEE Transaction on Power System 26(2), 897–904 (2011)
3. Khalid, S.K.A.D., Ahmed, H.B.A.: Using data envelopment analysis to measure cost efficiency with an application on Islamic banks. Scientific Journal of Administrative Development 4, 134–156 (2006)
4. Dimitris, K.S., Lamprini, V.S., Yiannis, G.S.: Data envelopment analysis with nonlinear virtual inputs and outputs. European Journal of Operational Research 202, 604–613 (2009)
5. Ramanathan, R.: An Introduction to Data Envelopment Analysis: A Tool for Performance Measurement. Sage Publication Ltd., New Delhi (2003)
6. Srinivas, T.: Data envelopment analysis: models and extensions. Production/Operation Management Decision Line, 8–11 (2000)
7. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science 30, 1078–1092 (1984)
8. Cooper, W.W., Seiford, L.M., Tone, K.: Data envelopment analysis. Springer, US (2007)
9. Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, Berkeley, CA, Tech. Rep. TR-95-012 (1995)
10. Plagianakos, V., Tasoulis, D., Vrahatis, M.: A Review of Major Application Areas of Differential Evolution. In: Advances in Differential Evolution, vol. 143, pp. 197–238. Springer, Berlin (2008)
11. Wang, F., Jang, H.: Parameter estimation of a bioreaction model by hybrid differential evolution. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC-2000), pp. 410–417 (2000)
12. Joshi, R., Sanderson, A.: Minimal representation multisensor fusion using differential evolution. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 29(1), 63–76 (1999)
13. Ilonen, J., Kamarainen, J., Lampine, J.: Differential evolution training algorithm for feed-forward neural networks. Neural Process. Lett. 17(1), 93–105 (2003)
14. Ali, M., Siarry, P., Pant, M.: An efficient differential evolution based algorithm for solving multi-objective optimization. European Journal of Operational Research (2011)
15. Das, S., Abraham, A., Chakraborty, U., Konar, A.: Differential evolution using a neighborhood-based mutation operator. IEEE Transaction on Evolutionry Computing 13(3), 526–553 (2009)
16. Jouni, L.: A constraint handling approach for differential evolution algorithm. In: Proceeding of IEEE Congress on Evolutionary Computation (CEC 2002), pp. 1468–1473 (2002)
17. Coello, C.A.C.: Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering 191(11-12), 1245–1287 (2002)

18. Ray, T., Kang, T., Chye, S.K.: An evolutionary algorithm for constraint optimization. In: Whitley, D., Goldberg, D., Cantu-Paz, E., Spector, L., Parmee, I., Beyer, H.G. (eds.) Proceeding of the Genetic and Evolutionary Computation Conference (GECCO 2000), pp. 771–777 (2000)
19. Capitaline Database, http://www.capitaline.com

Dynamic Tasks Scheduling Model for Performance Evaluation of a Distributed Computing System through Artificial Neural Network

M.P. Singh[1], P.K. Yadav[2], Harendra Kumar[1], and Babita Agarwal[1]

[1] Department of Mathematics & Statistics, Gurkula Kangari University, Hardwar-249404, (U.K.)
[2] Research Planing & Business Development C.B.R.I., Roorkee-247667, U.K. (INDIA)
{mpsinghgkv,prd_yadav}@rediffmail.com, balyan.kumar@gmail.com, bab_agg12@yahoo.co.in

Abstract. As technology has quickly and relentlessly advanced in the field of computer hardware, Distributed Computing System [DCS] has become increasingly popular. Performance enhancement is one of the most important issues in distributed systems. In this paper we have proposed a dynamic task allocation model based on artificial neural network [ANN] scheduling approach to arrange the tasks to the processors. Relocating the tasks from one processor to another at certain points during the course of execution of the program that contributes to the total cost of the running program has been taken into account. Phase-wise Execution Cost [EC], Inter Task Communication Cost [ITCC], Residence Cost [RC] of each task on different processors and Relocation Cost [REC] for each task has been considered while preparing the model.

Keywords: Distributed Computing System, Artificial Neural Network, Phase-wise Execution, Inter-Tasks Communication Cost, Execution Cost, Relocation Cost

1 Introduction

Distributed computing is a form of parallel computing, which is most commonly used to describe program parts running simultaneously on multiple processors in the same system. In a heterogeneous computing system, the tasks of an application program must be assigned to the processors to utilize the computational capabilities and resources of the system efficiently. Static & dynamic algorithms can be extremely efficient in heterogeneous system. However, in homogeneous system, how the tasks are assigned to the processor with the fewest outstanding jobs and the optimal assignment of the system for any number of processors

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 321–331.
springerlink.com

constrained into a tree; the shortest algorithm was developed by Bokhari [1,2]. While Stone [3] has suggested an efficient optimal algorithm for the problem of assigning tasks to two processors by making use of the well known network flow algorithm in the related two terminal network graph. Towslay [4] generalized the result of Bokhari to the case of series parallel structure. A further generalized programme structure is a partial k-tree [5] assuming the system as fully interconnected i.e. there exists a communication link between any two machines. However, the general n- processors problem (n>3) in a fully interconnected system is known to be NP-complete [6-7]. Cho and Park [8] and Lee et al [9] have suggested an optimal algorithm for the general structure problems in a linear array network with any number of processors.

All the above assignment models are static because they assume that once a task assigned to a processor it remains there during the execution. While in a dynamic model each task may have multiple computational characteristics. Dynamic algorithms are used more frequently because they offer a better efficiency having the flexibility of the tasks migration. Bokhari [10] analyzed the problem of dynamic assignment in a two processors system, which permits relocation of tasks from one processor to the other at certain points during the execution of the program. This relocation incurs a predefined relocation cost that contributes to the total cost of running the program and code conversion overheads. Yadav et al [11-12] generalized the algorithm of [10] to a number of processors. This paper focuses on heterogeneous DCS and addresses dynamic scheduling of the tasks.

2 Definitions

2.1 Phase

A phase of a distributed program is a continuous period of time during which n number of tasks are executed. A task can complete its execution only in one phase, and the number of phases of the executing program is k=ceil (m/n), here m is number of tasks.

2.2 Function Ceil (m/n)

$$ceil(m/n = \begin{Bmatrix} m/n, \textit{if } m \textit{ is the multiple of } n \\ \lfloor m/n \rfloor, \textit{ otherwise} \end{Bmatrix})$$

2.3 Residing Cost

Residence cost r_{ij} (where $1 \le i \le m$ & $1 \le j \le n$) is the amount of time spent by the task t_i residing on the processor P_j.

2.4 Relocation Cost

REC rel_{jk} is the cost of reassigning a from the j^{th} processor to another processor at the end of the k^{th} phase.

2.5 Maxlinkmod Function

This function finds first maximally linked tasks and remaining in decreasing order according to the sum of inter tasks communication costs with other tasks. All tasks are stored according to this order in a linear array Torder { }. Neural network of maxlinkmod () function is defined as:

It is a three layers feed forward network using neuron having the uni-polar activation function given in figure 1. Output of each layer is described as:

For first layer

Input vector Inp1 { } $\leftarrow \{c_{12}, c_{13} \ldots c_{m*(m-1)}\}$ (elements of the upper triangular matrix of ITCC (,)), Weight matrix W_{max1} (,)$\leftarrow [w_{ij}]_{m \times m*(m-1)/2}$ (value of each weight line is 1)

Output O_1 ( ) $\leftarrow W_{max1.} Inp_1^T$ (of order m x 1).

For second layer

Input vector Inp_2 ( ) $\leftarrow O_1$ ( ) (i.e output of first layer), Weight matrix W_{max2} (,)$\leftarrow [w_{ij}]_{(m*(m-1)/2) \times m,}$ (Weight of blue lines is 1, for red lines it is sgn(c_{ij} – 0.5), and for others it is 0),

Output O_2 ( ) $\leftarrow W_{max2} . O_1$, (O_2 ( ) is of order m* (m -1) x 1 elements). , In this layer, each element is compared with other (m -1) elements to search the maximum linked module.

For third layer

Input vector Inp_3 ( ) $\leftarrow O_2$ ( ), Weight matrix W_{max3} (,) = $[w_{ij}]_{n \times (m*(m-1)/2) \times m,}$ (Each line has weight value 1), O_3 ( ) $\leftarrow W_{max3}$ (,) O_2. Therefore using all these layers, Mlink_task (ITCC) gives output O_3 i.e.

$$Mlink\_task(ITCC) = \begin{Bmatrix} 1, if\ maxlinked\ module\ occurs \\ 0, otherwise \end{Bmatrix}$$

If the number of tasks obtained by maximum linked module is not equal to the number of processors then we select remaining modules according to maximum inter processing communication cost and stored in Mlink_task( ). Thus the module

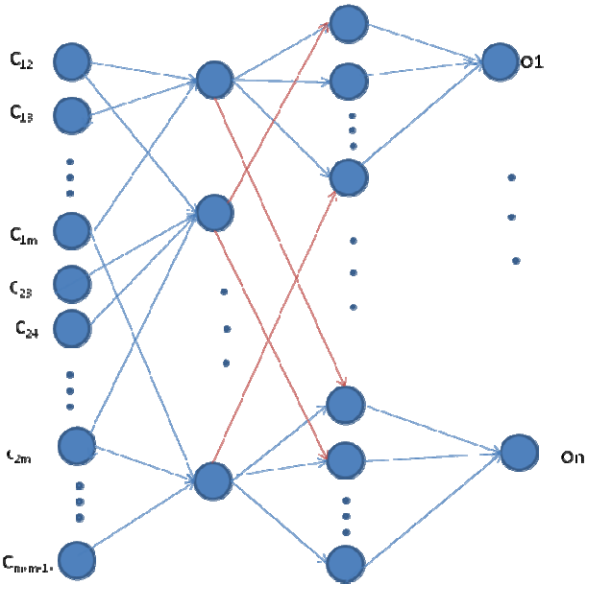


Fig. 1. Network for maximally link module

clusters are to be formed around the maximally linked module and one module is assigned to one processor out of these n modules using heap sorting method.

3 Notations

ETask{ }$_k$: A linear array to store n executing tasks during k^{th} phase.

NETask{ }$_k$: A linear array to store (m-n) non executing tasks during k^{th} phase.

ECM_ETask (,): Represents executing cost matrix for executing tasks of order n x n.

RCM_NETask (,) :Represents residing cost matrix for non executing tasks of order (m –n) x n.

NRECM(,) : New relocation matrix of order m x n.

NRECM_ETask (,): Represents relocation cost matrix for executing tasks of order n x n.

NRECM_NETask (,): Represents relocation cost matrix for non executing tasks of order (m –n) x n.

SETask(,) : A matrix of order n x n.

SNETask(,) : A matrix of order (m –n) x n

NITCCM(,): New inter task communication cost matrix of executing tasks with non executing tasks of order (m – n) x n.

AEtask { }$_k$: An array which store the assignment of executing tasks with their positions during k^{th} phase.

ANEtask{ }k : An array which store the assignment of non executing tasks with their positions during kth phase.

4 Assumptions

i. In each phase, n tasks are executed and remaining tasks are free to reside on either of the processors.
ii. Execution cost matrix of executing tasks ECM_ETask (,), Residing cost matrix of non executing tasks RCM_NETask (,) of non executing tasks, inter-task communication cost matrix ITCCM (,) and relocation cost matrix RECM (,) for reassigning each task from one processor to others at the end of phases are known.
iii. Some of the tasks have restrictions for being allocated on some particular processors because of unavailability of desired facilities which may include access to particular peripheral device, access to a particular task, and high speed arithmetic capability of the processor. The execution costs of such tasks are taken to be infinite on those processors.
iv. Relocation Cost in the first phase at each processor is zero.

5 Scheduling Problem

Consider a distributed program consisting of a set T= {t_1, t_2, t_3, - - - t_m} of m tasks to be allocated to a set P= {p_1, p_2, p_3, - - - p_n} of n processors divided into k phases. With each phase the following information are associated:

i. Program will find the executing tasks during this phase and their execution costs on each processor in a heterogeneous system.
ii. Residing costs of the remaining tasks, except for the executing tasks, on each processor.
iii. Inter task communication cost between the executing tasks and all others tasks if they are on different processor.
iv. Relocation cost for reassigning each task from one processor to the others at the end of the phases.

In general, the objective of tasks assignments is to minimize processing cost of a distributed program by proper allocation of the tasks to the processors. The cost of an assignment A, i.e. TCOST (A), is the sum of execution, inter task communication, residing, and relocation costs.

$$TCOST\ (A) = EC\ (A) + RC\ (A) + REC\ (A) + IT\ CC\ (A)$$

6 Allocation Technique

To allocate tasks dynamically, a mathematical model has been developed by using artificial neural network. As shown in the flowchart figure 4, first the order of the tasks has been decided in which these will execute. In each phase, n tasks are assigned and others are resided. After this, the relocation cost of the tasks in each phase is added in ECM (,) and RCM (,).

6.1 Neural Network to Calculate the Effect of Relocation Cost at r^{th} Processor during k^{th} phase

It is a two layers network as shown in figure2. In first layer, weight matrix $w_{rel1}(,)$ =RECM(,k)

Input vector $x=\{x_1, x_2 - - - x_n\}$ where

$$x_i = \begin{Bmatrix} 1 \text{ for } j = r, r \text{ repersents the processor}, 1 \le r \le n, 1 \le j \le n \\ 0 \text{ otherwise} \end{Bmatrix}$$

Output vector $O_1 = w_{rel1}\, x^T$ is of order mx1. It represents maximum relocation cost on r^{th} processor. In second layer we get the actual relocation cost on the r^{th} processor. In second layer the weight matrix W_{rel2} is a diagonal matrix of order mxm, and defined as:

$$wrel2ii = \begin{Bmatrix} 0, i^{th} \text{ for } j = r, r \text{ repersents the processor}, 1 \le i \le m \\ 1, \text{otherwise} \end{Bmatrix}$$

RECM (1: m, r) = w_{rel2}. O_1. It gives actual relocation cost on the r^{th} processor i.e. r^{th} column of NRECM (, k). It is an iterative procedure and will execute for $\mathbf{1} \le r \le n$.

6.2 Neural Network to Find ITCC of Executing Tasks with Non Executing Tasks during k^{th} Phase

It is a single layer network as shown in figure 3. Here input vector $x=\{x_1,x_2,..x_n\}$ where $xi = c_{e_i r_j} \in ITCC(,)$ here e_i (task's number of $\mathbf{i}^{th}$ executing task) of ETask $\{\}_K$, r_j(task's number of j^{th} non executing task) of NETask$\{\}_k$, $1 \le i \le n$, and $1 \le j \le (m-n)$.

Weight matrix w is of order nxn and its elements are defined as:

$$w_{kl} = \begin{Bmatrix} 0, \text{if task at } i^{th} \text{ place ids assign to } k^{th} \text{ processor} \\ 1, \text{otherwise} \end{Bmatrix}$$

ITCC (j,1:n) = $(w\, x^T)^T$ gives j^{th} row of the $NITCCM(,)_K$

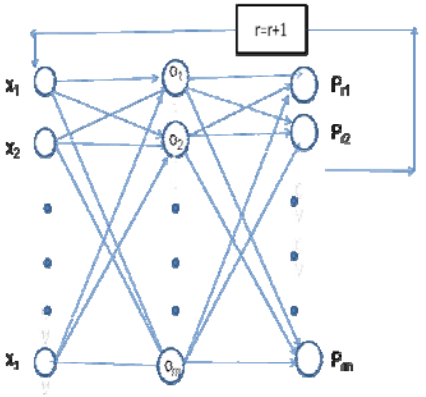


Fig. 2. Neural Network to Calculate Relocation Cost

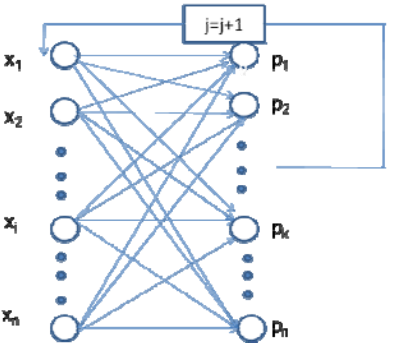


Fig. 3. Neural Network to select ITCC

6.3 Proposed Method

Initially the problem for allocating a program dynamically finds number of phases i.e. k. The starting of the algorithm decides the order of the tasks in which these will execute and divide into the group of n tasks. In each phase a group of executing tasks will be assigned and non executing tasks will be resided. In the first phase, relocation costs on all processors are zero. Relocation cost for assigning the i^{th} task from one processor to another at the end of k^{th} phase is not constant. If the number m of tasks is not multiple of the number n of processors then to make it multiple dummy tasks are added in last group with all costs zero. The executing task $t_s \in$ ETask{ }$_k$ must be assigned to processor p_r where $1 \leq r \leq n$ during k^{th} phase for which the sum of its execution and relocation costs is minimum. TEC_k represents the total executing cost. When residing task $t_g \in$ NETask{ }$_k$resides on a processor, it has three- type costs, that is residing, relocation, and ITCC with the executing task t_s. TRC_k reprsents the total costs due to the residing tasks in the k^{th} phase.

The Total Cost $(TCOST)_k$ for the k^{th} phase is the sum of TEC_k and TRC_k. This is represented as:

$$(TCOST)_k = TEC_k + TRC_k$$

7 Algorithm

Step 1: *Input m, n, ECM (,), RCM (,), ITCCM (,)*

Step 2: *k← ceil (m/n),*

(Add dummy tasks if required with all costs zero and m is modified)

*m ←k * n*

Input RECM (, ,), Algorithm modified the matrices ECM(,), RCM(,), ITCCM(,)corresponding to dummy tasks

Step.3: *Torder{ }←maxlinkmod (ITCCM (,), m, n) (It gives the order of tasks in which these will execute)*

Step 4: *for q =1 to k*

Step 4.1: *ETask{ }$_q$ ←Torder{(q-1)*n+1: q*n}, NETask{ }$_q$ ← Torder { }\ETask{ }$_q$*

Step 4.2: *If q=1 then NRECM(,)←0*

else

NRECM (,)$_q$

(Neural network to calculate NRECM (,))

Step 4.3: *SETask(,)$_q$← (ECM_Task (,)$_q$ + NRECM_ETask(,)$_q$)*

Step 4.4: *A_{Etask} { }$_q$ ← assign the tasks by using Heap Sorting Algorithm for matrix SETask{ }$_q$ and store the tasks and their positions on the processors.*

TEC$_q$← sum of execution costs for executing tasks.

Step 4.5: *NITCCM (,)$_q$ (Neural network to calculate NITCCM (,))*

Step 4.6: *SNETask (,)$_q$ ←(RCM_NETask (,)$_q$ + NRECM_NETask(,)$_q$ + NITCCM(,)$_q$)*

Step 4.7: *Store assignment of non executing tasks for matrix SNETask { }$_q$ to that processors for which their residing costs are minimum and their positions on the processors in $A_{NEtask.}${ }$_q$. TRC$_q$ ← (total residing cost due to residing tasks).*

Step 4.8: *(TCOST)$_q$ ← TEC$_q$ + TRC$_q$*

Step 4.9: *q ←q + 1, If q=k+1, go to step 5 else go to step4.*

Step.5: $TCOST = \sum_{q=1}^{k} (TOCST)_q$

Step 6: *End.*

Input ECM (,), RCM (,) ITCCM (,), for m=7 and n= 3 are given as:

ECM(,)=

	p_1	p_2	P_3
t_1	6	4	8
t_2	2	6	4
t_3	∞	6	8
t_4	6	3	∞
t_5	5	4	2
t_6	8	6	4
t_7	4	7	8

RCM(,)=

	p_1	p_2	P_3
t_1	3	1	2
t_2	2	1	3
t_3	3	4	2
t_4	4	3	1
t_5	4	3	2
t_6	2	1	1
t_7	3	1	2

ITCCM(,)=

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
t_1	0	2	3	2	4	2	4
t_2	2	0	2	2	2	0	3
t_3	3	2	0	4	2	2	3
t_4	2	2	4	0	3	2	1
t_5	4	2	2	3	0	4	3
t_6	2	0	2	2	4	0	2
t_7	4	3	3	1	3	2	0

Here k=ceil (7/3) =3, Here we add two dummy tasks.t_8 and t_9. Using the steps 2 to 6 of the algorithm, we get the total cost of the system 108. The results of the problem are shown by figure 5. (The tasks represented by * are executing tasks

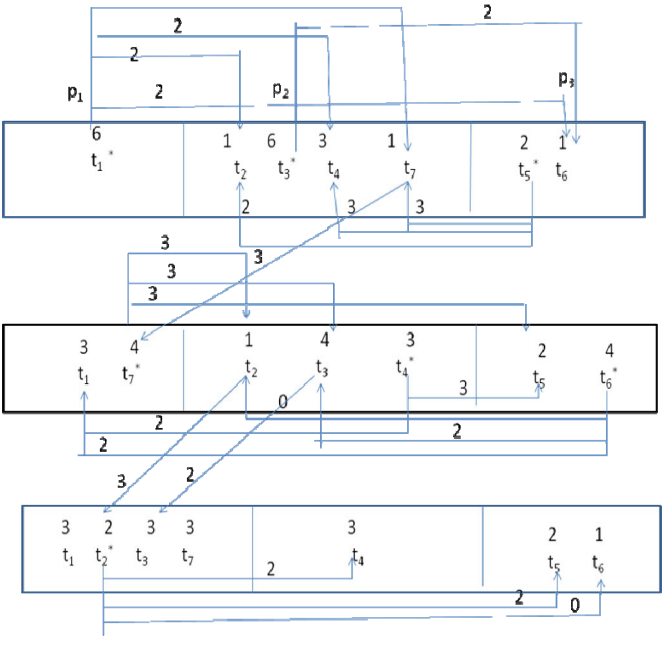


Fig. 5. Optimal assignment graph (TCOST = 108)

while remaining are residing tasks) costs associated with tasks are their execution costs (for executing tasks represented by *) or residing costs. The edges connecting the executing task with other residing tasks during the same phase represent the communication and weights of these edges represent communication costs between the executing task and other residing tasks. The edges connecting the task of a phase to the task of next phase represent the relocation of the tasks and the weights of these edges represent the costs of relocating the tasks. For example the task t_7 is relocating from processor p_2 to processor p_1 between the phase 1 and phase 2. The above result shows that the assignment of the tasks is changing phase to phase. Optimal cost for this task program is 108. The above neural network is valid for any number of processor and tasks.

8 Conclusions

We have presented an efficient solution technique for tasks execution dynamically in a DCN. Starting with the definitions of the phase of a modular program, we have developed an ANN based mathematical model for allocating the tasks of a program which is divided into k phases to $n \geq 2$ processors. The present model is compared with [Yada08]. The algorithm suggested by [Yada08] can execute only one task in a phase and others may reside and wait for their execution. It causes the increase in total execution cost but in the present algorithm all processors are busy for executing the tasks and none of the processor is idle. The present algorithm is very fast and efficient while executing the tasks. The total program executing cost is very much reduced as compared to others algorithms. The suggested algorithm is simple in use and many examples have been tested and it is found that the results are better in all the cases.

References

[1] Bokhari, S.H.: A Shortest Tree Algorithm for Optimal Assignments across Space and Time in a Distributed Processor Systems. IEEE Trans. Software Eng. SE-7(6), 583–589 (1981)

[2] Bokhari, S.H.: Assignment Problem in Parallel and Distributed Computing. Kulwer Academic Publishers (1987)

[3] Stone, H.S.: Multiprocessor Scheduling with the Aid of Network Flow Algorithms. IEEE Trans. Software SE-3(l), 85–93 (1977)

[4] Towsley, D.F.: Allocating Programs Containing Branches and Loops within a Multiple Processor System. IEEE Trans. Software Eng. SE-12(10), 1018–1024 (1986)

[5] Baca, D.F.: Allocating Modules to Processors in a Distributed System. IEEE Trans. Software Eng. SE-15(11), 1427–1436 (1989)

[6] Ghafoor, Yang, J.: A Distributed Heterogeneous Supercomputing Management System. IEEE Comput. 6, 78–86 (1993)

[7] Singh, M.P., Yadav, P.K., Kumar, H.: A Tasks Allocation Model for Efficient Utilization Of Processor's Capacity in Heterogeneous Distributed Systems. Presented 9th Conference of International Academy of Physical Sciences, Held at Agra University Agra February 03-05 (2007)

[8] Cho, S.Y., Park, K.H.: Dynamic Task Assignment in Heterogeneous Linear Array Networks for Metacomputing. In: Proceeding of IPPS, pp. 66–71 (1994)

[9] Lee, C.H., Lee, D., Kim, M.: Optimal Task Assignment in Linear Array Networks. IEEE Trans. Comput. C-41(7), 877–880 (1992)

[10] Bokhari, S.H.: Dual Processor Scheduling with Dynamic Re-Assignment. IEEE Trans. On Software Engineering SE-5, 341–349 (1979)

[11] Kumar, V., Singh, M.P., Yadav, P.K.: An Efficient Algorithm for Multiprocessor Scheduling with Dynamic Re-assignment. In: Proceeding of 6th National Seminar on Theoretical Computer Science, pp. 105–118 (1996)

[12] Yadav, P.K., Singh, M.P., Kumar, H.: Scheduling Algorithm: Tasks Scheduling Algorithm for Multiple Processors with Dynamic Re-assignment. Journal of Computer Systems, Networks, and Communications (1), Article ID 578180, 1–9 (2008), doi:10.1155/2008

An Analysis of Generalised Approximate Equalities Based on Rough Fuzzy Sets

B.K. Tripathy, Abhishek Jhawar, and Ekta Vats

SCSE, VIT University, Vellore – 632014, Tamilnadu, India
tripathybk@rediffmail.com,
{avio.abhishek,ektavats}@gmail.com

Abstract. Three types of rough equalities were introduced by Novotny and Pawlak ([7, 8,9]), which take care of approximate equalities of sets. These sets may not be equal in the usual sense. These notions were generalized by Tripathy, Mitra and Ojha ([12]), who introduced the concepts of rough equivalences of sets. These approximate equalities of sets capture equality of the concerned sets at a higher level than their corresponding rough equalities. Some more properties were proved in [13]. Two more approximate equalities were introduced by Tripathy [11] and comparative analysis of their efficiency was provided. In this paper, we generalise these approximate equalities by considering rough fuzzy sets instead of only rough sets. A concept of leveled approximate equality is introduced and properties are studied. We also provide suitable examples to illustrate the applications of the new notions and provide an analysis of their relative efficiency.

Keywords: Bottom approximate equality, top-approximate equality, leveled approximate equality, fuzzy set, rough set, rough fuzzy set.

1 Introduction

The observation that most of our traditional tools for formal modeling, reasoning and computing are crisp, deterministic and precise in character, which restricts their applicability in real life situations, led to the extension of the concept of crisp sets so as to model imprecise data and enhance their modeling power. Two of the most fruitful efforts in this direction are the notion of fuzzy sets introduced by Zadeh [16] and the notion of rough sets introduced by Pawlak [2].

The basic notion of equality of two sets is independent of the user or more precisely the user knowledge about the universe dealt with. In an attempt to incorporate the user knowledge about the structure of the universe in concluding about the equality of two sets the notions of approximate equalities were introduced by Novotny and Pawlak ([7, 8,9]). This is an important feature as the sets considered may not be equal in the normal sense but they have close features to assume that they are approximately equal. That is, basing upon our knowledge

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 333–346.
springerlink.com

and requirement we can assume that the two sets are indistinguishable. Even these definitions of approximate equalities were found to have limited scope in defining approximate equality of sets and the concepts of rough equivalences were introduced and studied by Tripathy et al. ([12]). Their notions capture the approximate equalities of two sets at a higher level and obviously are more general. In [11] two more types of approximate equalities of sets were introduced and an analysis of the four kinds of approximate equalities was made regarding their applicability and efficiency. It was shown that the notion of approximate equivalence seems to be the most natural among these four.

It is well known that the notions of fuzzy set and rough set complement each other. So, in this paper we extend the study of approximate equalities thus far by considering the hybrid notion of rough fuzzy set in order to achieve this. In fact, we introduce the notion of leveled approximate equality also.

Properties of approximate equalities established by Novotny and Pawlak [7, 8,9] were analysed in [12] and it was shown that some of the statements were incorrect. In [12] attempts were made to extend these properties to the generalized situation of rough equivalences. It was found that the properties failed to hold in their full generality and mostly parts were found to hold true. The other parts were established under suitable conditions. The validity of some basic algebraic properties involving union, intersection and complementation of sets were tested for their validity with equality of sets being replaced by rough equivalence in [13]. We shall verify the validity of the approximate equalities under the generalised circumstances.

1.1 Basic Rough Sets

Let U be a universe of discourse and R be an equivalence relation over U. By U/R we denote the family of all equivalence class of R, referred to as categories or concepts of R and the equivalence class of an element $x \in U$, is denoted by $[x]_R$. By a knowledge base, we understand a relational system $K = (U, R)$, where U is as above and R is a family of equivalence relations over U. For any subset P $(\neq \phi) \subseteq R$, the intersection of all equivalence relations in P is denoted by IND (P) and is called the indiscernibility relation over P. Given any $X \subseteq U$ and $R \in$ IND (K), we associate two subsets, $\underline{R}X = \cup\{Y \in U/R : Y \subseteq X\}$ and $\overline{R}X = \cup\{Y \in U/R : Y \cap X \neq \phi\}$, called the R-lower and R-upper approximations of X respectively.

The R-boundary of X is denoted by BN_R (X) and is given by BN_R (X) = $\overline{R}X - \underline{R}X$. The elements of $\underline{R}X$ are those elements of U, which can certainly be classified as elements of X, and the elements of $\overline{R}X$ are those elements of U, which can possibly be classified as elements of X, employing knowledge of R. We say that X is rough with respect to R if and only if $\underline{R}X \neq \overline{R}X$, equivalently BN_R (X) $\neq \phi$. X is said to be R-definable if and only if $\underline{R}X = \overline{R}X$, or BN_R (X) $= \phi$.

1.2 Fuzzy Sets

As mentioned in the introduction, fuzzy set introduced by Zadeh [16] is one of the approaches to capture vagueness in concepts. In the fuzzy set approach every member x of a set $X \subseteq U$ is associated with a grade of membership, which we denote by X(x) and is a real number lying in [0, 1]. The set of all functions from U to the unit interval [0, 1], is called the fuzzy power set of U and is denoted by F(U). It follows that P(U) $\subseteq$ F(U).

1.3 Rough Fuzzy Sets

It was established by Dubois and Prade [1] that instead of being rival theories, the two theories of fuzzy sets and rough sets complement each other. In fact they combined these two concepts to develop the hybrid models of fuzzy rough sets and rough fuzzy sets. Rough fuzzy sets are defined [1] as follows.

Let (U, R) be an approximation space. Then for any X$\in$ F(U), the lower and upper approximations of X with respect to R are given by

(1.3.1) $(\underline{R}X)(x) = \inf_{y\in[x]_R} X(y)$, for all x $\in$ U and

(1.3.2) $(\overline{R}X)(x) = \sup_{y\in[x]_R} X(y)$, for all x $\in$ U.

2 Approximate Equalities of Sets

As described in the introduction, sometimes exact equality (equality in the mathematical sense) is too stringent to apply in day to day life. We often talk about equality of sets or domains, which can be considered to be equal for the purpose or under the circumstances in real life situations. So, approximate equalities play a significant role in our reasoning. Also, it is dependent upon the knowledge the assessors have about the domain under consideration as a whole but mostly not the knowledge about individuals. The following four kinds of approximate equalities using rough sets have been introduced and studied so far (see [11] for a comparative analysis)

2.1 The Kinds of Approximate Equalities

Taking different combinations of the two types of conditions on lower approximations and the two types of conditions on upper approximations we get four kinds of approximate equalities of sets. This is summarized in the following table:

Table 1.

Upper approximation / Lower approximation	$\overline{R}X = \overline{R}Y$	$\overline{R}X$ and $\overline{R}Y$ are U or $\neq U$ together
$\underline{R}X = \underline{R}Y$	Rough Equalities	Approximate rough equalities
$\underline{R}X$ and $\underline{R}Y$ are ϕ or $\neq \phi$ together	Approximate rough equivalences	Rough Equivalences

As noted by Pawlak ([2], p.26), all these approximate equalities of sets are of relative character, that is things are equal or not equal from our point of view depending on what we know about them. So, in a sense the definition of rough equality refers to our knowledge about the universe.

It is worth noting that each of the above four types of approximate equalities are defined through three notions called bottom equality, top equality and total equality defined in [3]. Several properties of these three approximate equalities were established by Novotny and Pawlak [7, 8, 9 and also see 3]. The properties obtained from these properties by interchanging bottom and top approximate equalities are called replacement properties [3, 10, 11, 12, 13]. It was mentioned in [3] that these replacement properties do not hold in general. But, it is established in [10] that some of these properties actually hold and others hold under suitable conditions. We avoid presenting both these types of properties here.

The following two properties of lower and upper approximations of rough sets are instructive and it has been observed that the inclusions cannot be replaced with equalities in general:

(2.1.1) $\underline{R}X \cup \underline{R}Y \subseteq \underline{R}\left(X \cup Y\right)$ and

(2.1.2) $\overline{R}\left(X \cap Y\right) \subseteq \overline{R}X \cap \overline{R}Y$

2.2 Other Rough Equalities

In order to show the superiority of rough equivalence the following example and analysis was provided in [13].

2.2.1 Universe of Cattle

Let us consider the cattle in a locality as our universal set C. We define a relation R over C by x R y if and only if x and y are cattle of the same kind. This is an equivalence relation and decomposes the universe into disjoint equivalence classes. Suppose for example, C = {Cow, Buffalo, Goat, Sheep, Bullock}. Let X and Y be the set of cattle owned by two persons P_1 and P_2 in the locality. We cannot talk about the equality of X and Y in the usual sense as the cattle cannot be owned by two different people. Similarly, we cannot talk about the rough equality of X and Y except the trivial case when both the persons do not own any cattle.

The different possibilities for approximate equality or otherwise of X and Y to hold have been discussed in [12] and are categorized under six cases as follows:

Case (i). $\bar{R}X$, $\bar{R}Y$ are not U and $\underline{R}X$, $\underline{R}Y$ are ϕ. That is P_1 and P_2 both have some kind of cattle but do not have all cattle of any kind in the locality. So, they are equivalent.

Case (ii) $\bar{R}X$, $\bar{R}Y$ are not U and $\underline{R}X$, $\underline{R}Y$ are not ϕ. That is P_1 and P_2 both have some kind of cattle and have all cattle of some kind in the locality. So, they are equivalent.

Case (iii) $\bar{R}X$, $\bar{R}Y$ are U and $\underline{R}X$, $\underline{R}Y$ are ϕ. That is P_1 and P_2 both cattle of all kinds but do not have all cattle of any kind in the locality. So, they are equivalent.

Case (iv) $\bar{R}X$, $\bar{R}Y$ are U and $\underline{R}X$, $\underline{R}Y$ are not ϕ. That is P_1 and P_2 both have all kinds of cattle and also have all cattle of some kind in the locality. So, they are equivalent.

There are two different cases under which we can talk about the non- equivalence of P1 and P2.

Case (v) one of $\bar{R}X$ and $\bar{R}Y$ is U and the other one is not. Then, out of P_1 and P_2 one has cattle of all kinds and the other one does not have so. So, they are not equivalent.

Case (vi) Out of $\underline{R}X$ and $\underline{R}Y$, one is ϕ and the other one is not. Then, out of P_1 and P_2 one does not have all cattle of any particular kind, where as the other has all cattle of at least one kind. So, they are not equivalent.

2.2.2 Basic Properties of Rough Equivalence

Several properties of rough equivalences were obtained in [12]. In order to obtain these properties the concepts of different rough inclusions ([5]) and rough comparisons ([12]) are needed.

Definition 2.2.2.1: Let K = $(U, \Re)$ be a knowledge base, X, Y $\subseteq$ U and R$\in$ IND(K). Then

(i) We say that X is *bottom R-included* in Y ($X \subsetneq_R Y$) if and only if $\underline{R}X \subseteq \underline{R}Y$

(ii) We say that X is *top R-included* in Y $\left(X \tilde{\subset}_R Y\right)$ if and only if $\bar{R}X \subseteq \bar{R}Y$.

(iii) We say that X is *R-included* in Y $\left(X \tilde{\subsetneq}_R Y\right)$ if and only if $X \subsetneq_R Y$ and $X \tilde{\subset}_R Y$

Definition 2.2.2.2: (i) we say X, Y $\subseteq$ U are *bottom rough comparable* if and only if $X \subsetneq Y$ or $Y \subsetneq X$ holds.

(ii) We say X, Y $\subseteq$ U are *top rough comparable* if and only if $X \tilde{\subset} Y$ or $Y \tilde{\subset} X$ holds.

(iii) We say X, Y $\subseteq$ U are *rough comparable* if and only if X and Y are both top and bottom rough comparable.

Property 2.2.2.1: (i) If $X \bigcap Y$ is b_eqv. to both X and Y then X is b_eqv. to Y.
(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that X and Y are bottom rough comparable.
Property 2.2.2.2: (i) If $X \bigcup Y$ is t_eqv. to both X and Y then X is t_eqv. to Y.
(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that X and Y are top rough comparable.
Property 2.2.2.3: (i) If X is t_eqv. to X' and Y is t_eqv. to Y' then it may or may not be true that $X \bigcup Y$ is t_eqv. to $X' \bigcup Y'$.
(ii) A sufficient but not necessary condition for the result in (i) to be true is that X and Y are top rough comparable and X' and Y' are top rough comparable.
Property 2.2.2.4: (i) X is b_eqv. to X' and Y is b_eqv. to Y' may or may not imply that $X \bigcap Y$ is b_eqv. to $X' \bigcap Y'$
(ii) A sufficient but not necessary condition for the result in (i) to be true is that X and Y are bottom rough comparable and X' and Y' are bottom rough comparable.
Property 2.2.2.5: (i) X is t_eqv. to Y may or may not imply that $X \bigcup (-Y)$ is t_eqv. to U.
(ii) A sufficient but not necessary condition for result in (i) to hold is that $X \backsimeq Y$.
Property 2.2.2.6: (i) X is b_eqv. to Y may not imply that $X \bigcap (-Y)$ is b_eqv. to ϕ.
(ii) A sufficient but not necessary condition for the result in (i) to hold true is that X $\simeq$ Y.
Property 2.2.2.7: If $X \subseteq Y$ and Y is b_eqv. to ϕ then X is b_eqv. to ϕ.
Property 2.2.2.8: If $X \subseteq Y$ and X is t_eqv. to U then Y is t_eqv. to U.
Property 2.2.2.9: X is t_eqv. to Y if and only if –X is b_eqv. to –Y.
Property 2.2.2.10: X is b_eqv. to ϕ, Y is b_eqv. to $\phi \Rightarrow X \bigcap Y$ is b_eqv. to ϕ.
Property 2.2.2.11: If X is t_eqv. to U or Y is t_eqv. to U then $X \bigcup Y$ is t_eqv. to U.

2.2.3 Replacement Properties for Rough Equivalence

In parallel to the properties of interchange for rough equalities, it was proved in [12] that properties (2.2.2.7) to (2.2.2.11) hold true under replacements. Also, the same is true for properties 2.2.2.1(i), 2.2.2.2(i), 2.2.2.3(i), 2.2.2.4(i), 2.2.2.5(i), 2.2.2.6(i). For the other parts of these properties to be true, additional conditions are necessary.

2.2.4 Other Approximate Equalities

In [13] the other two types of approximate equalities called the approximately rough equivalence and the approximately rough equal were introduced as follows.

Definition 2.2.4.1: We say that two sets X and Y are approximately rough equivalent if and only if both $\underline{R}X$ and $\underline{R}Y$ are ϕ or not ϕ together and $\overline{R}X = \overline{R}Y$.

Definition 2.2.4.2: We say that two sets X and Y are approximately rough equal if and only if $\underline{R}X = \underline{R}Y$ and both $\overline{R}X$ and $\overline{R}Y$ are U or not U together.

2.3 Comparisons of Approximate Equalities

In [11] the following comparative analysis of the four types of approximate equalities was made.

2.3.1 The condition that two sets are lower approximately equal if and only if the two sets have the same lower approximation is satisfied in only rare and restricted cases. Since we are using this property in case of both rough equal and approximately rough equal definitions, these two cases of rough equalities seem to have lesser utility than the corresponding rough equivalences

2.3.2 The condition that the two upper approximations be equal provides freedom to define equality in a very approximate sense and is quite general than these two being equal to U or not simultaneously. But, sometimes it seems to be unconvincing.

2.3.3 The concept of approximate rough equivalence is neither unconvincing nor unnatural. This is the most natural and best among the four types.

2.3.4 The fourth type of approximate rough equality happens to be the worst among the four types of approximate equalities considered.

2.4 Properties of Approximate Rough Equivalence and Approximate Rough Equality

The satisfiability of the basic properties by approximate rough equivalence and approximate rough equality were deduced by Tripathy [11]. It is shown that some of these properties hold true; whereas some other do not hold true in general.

Also, similar results have been obtained for replacement properties for these two types of approximate equalities in [11].

3 Approximate Equalities Using Rough Fuzzy Sets

Following the four types of approximate equalities, we can define four types of generalised approximate equalities depending upon rough approximation of fuzzy sets as follows.

Here, we take X and Y to be fuzzy sets. $\underline{R}X$ and $\underline{R}Y$ are the lower approximations of X and Y. $\overline{R}X$ and $\overline{R}Y$ are the upper approximations of X and Y respectively. For any fuzzy set A, we denote the $\alpha - cut$ of A by A_α and it consists of elements of U having membership value greater than α. When $\alpha = 0$, we obtain the support set of A, denoted by $A_{>0}$. Since the $\alpha - cut$ of a fuzzy set is a crisp set, the equalities in the table below are crisp set equalities.

Table 2.

Upper Approximation ⟶ / Lower Approximation ↓	$(\overline{RX})_\alpha = (\overline{RY})_\alpha$	$(\overline{\mathrm{R}}\mathrm{X})_\alpha$ and $(\overline{\mathrm{R}}\mathrm{Y})_\alpha$ are U or not U together
$(\underline{RX})_\alpha = (\underline{RY})_\alpha$	α – Rough Fuzzy Equality	α – Approximate Rough Fuzzy Equality
$(\underline{\mathrm{R}}\mathrm{X})_\alpha$ and $(\underline{\mathrm{R}}\mathrm{Y})_\alpha$ are ϕ or not ϕ together	α – Approximate Rough Fuzzy Equivalence	α – Rough Fuzzy Equivalence

Let T(i, j) ; i = 1, 2 and j = 1,2 denote the (i, j)th position in the above table. When X and Y are crisp sets, the concepts in table 3.1 reduce to the corresponding concepts in table 1. Since α is a real number lying in [0, 1], we get infinitely many levels of equalities under each of the four categories in table 2.

3.1 Examples

In this section we provide examples to illustrate the relative efficiencies of the four types of approximate equalities of fuzzy sets introduced in the above section.

Example 3.1
We illustrate here with one example, the applicability and relative efficiency of the four approximate equalities of fuzzy sets. Let U = $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$. R be an equivalence relation on U such that

$$\mathrm{U/R} = \{\{x_1, x_2\}, \{x_3, x_4, x_5, x_6\}, \{x_7, x_8\}\},$$
$$X = \{(x_1, 0.2), (x_2, 0.8), (x_3, 0.7), (x_4, 0.4), (x_5, 0.6), (x_6, 0), (x_7, 1), (x_8, 0)\} \textit{and}$$
$$Y = \{(x_1, 0), (x_2, 0.7), (x_3, 0.9), (x_4, 0.6), (x_5, 0.4), (x_6, 0.2), (x_7, 0.8), (x_8, 0.8)\}$$

be two fuzzy sets defined over U. Then we have,

$$\underline{RX} = \{(x_1, 0.2), (x_2, 0.2), (x_3, 0), (x_4, 0), (x_5, 0), (x_6, 0), (x_7, 0), (x_8, 0)\}. \text{So,}$$
$$(\underline{RX})_{>0} = \{x_1, x_2\} \neq \phi.$$
$$\underline{RY} = \{(x_1, 0), (x_2, 0), (x_3, 0.2), (x_4, 0.2), (x_5, 0.2), (x_6, 0.2), (x_7, 0.3), (x_8, 0.3)\}.$$
$$So, (\underline{RY})_{>0} = \{x_3, x_4, x_5, x_6, x_7, x_8\} \neq \phi.$$
$$X^C = \{(x_1, 0.8), (x_2, 0.2), (x_3, 0.3), (x_4, 0.6), (x_5, 0.4), (x_6, 1), (x_7, 0), (x_8, 1)\}$$
$$\underline{RX}^C = \{(x_1, 0.2), (x_2, 0.2), (x_3, 0.3), (x_4, 0.3), (x_5, 0.3), (x_6, 0.3), (x_7, 0), (x_8, 0)\}$$
$$\overline{RX} = \{(x_1, 0.8), (x_2, 0.8), (x_3, 0.7), (x_4, 0.7), (x_5, 0.7), (x_6, 0.7), (x_7, 1), (x_8, 1)\}$$
$$(\overline{RX})_{>0} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} = U.$$

$Y^C = \{(x_1,1),(x_2,0.3),(x_3,0.1),(x_4,0.4),(x_5,0.6),(x_6,0.8),(x_7,0.2),(x_8,0.7)\}$

$\underline{R}Y^C = \{(x_1,0.3),(x_2,0.3),(x_3,0.1),(x_4,0.1),(x_5,0.1),(x_6,0.1),(x_7,0.2),(x_8,0.2)\}$

$\overline{R}Y = \{(x_1,0.7),(x_2,0.7),(x_3,0.9),(x_4,0.9),(x_5,0.9),(x_6,0.9),(x_7,0.8),(x_8,0.8)\}$

$(\overline{R}Y)_{>0} = \{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8\} = U.$

We find that the two fuzzy sets are similar to each other in the sense that the membership values of the elements in the two sets differ by very small numbers. Still these are neither rough fuzzy equal nor approximately rough fuzzy equal. But these two are approximately rough fuzzy equivalent and rough fuzzy equivalent. So, the last two notions are more realistic than the earlier two.

Example 3.2

Let us modify example 3.1 slightly by taking

$X = \{(x_1,0),(x_2,0),(x_3,0.7),(x_4,0.4),(x_5,0.6),(x_6,0.2),(x_7,1),(x_8,0)\}$ *and*

$Y = \{(x_1,0.2),(x_2,0.2),(x_3,0.9),(x_4,0.6),(x_5,0.2),(x_6,1),(x_7,0),(x_8,0)\}$

Then we have,

$\underline{R}X = \{(x_1,0),(x_2,0),(x_3,0.2),(x_4,0.2),(x_5,0.2),(x_6,0.2),(x_7,0),(x_8,0)\}$ .So,

$(\underline{R}X)_{>0} = \{x_3,x_4,x_5,x_6\} \neq \phi.$

$\underline{R}Y = \{(x_1,0.2),(x_2,0.2),(x_3,0.2),(x_4,0.2),(x_5,0.2),(x_6,0.2),(x_7,0),(x_8,0)\}.$

$So, (\underline{R}Y)_{>0} = \{x_1,x_2,x_3,x_4,x_5,x_6\} \neq \phi.$

$X^C = \{(x_1,1),(x_2,1),(x_3,0.3),(x_4,0.6),(x_5,0.4),(x_6,0.8),(x_7,0),(x_8,1)\}$

$\underline{R}X^C = \{(x_1,1),(x_2,1),(x_3,0.3),(x_4,0.3),(x_5,0.3),(x_6,0.3),(x_7,0),(x_8,0)\}$

$\overline{R}X = \{(x_1,0),(x_2,0),(x_3,0.7),(x_4,0.7),(x_5,0.7),(x_6,0.7),(x_7,1),(x_8,1)\}$

$(\overline{R}X)_{>0} = \{x_3,x_4,x_5,x_6,x_7,x_8\} \neq U.$

$Y^C = \{(x_1,0.8),(x_2,0.8),(x_3,0.1),(x_4,0.4),(x_5,0.2),(x_6,0),(x_7,1),(x_8,1)\}$

$\underline{R}Y^C = \{(x_1,0.8),(x_2,0.8),(x_3,0),(x_4,0),(x_5,0),(x_6,0),(x_7,1),(x_8,1)\}$

$\overline{R}Y = \{(x_1,0.2),(x_2,0.2),(x_3,1),(x_4,1),(x_5,1),(x_6,1),(x_7,0),(x_8,0)\}$

$(\overline{R}Y)_{>0} = \{x_1,x_2,x_3,x_4,x_5,x_6\} \neq U.$

We find that the two fuzzy sets differ much in the membership values of their elements, particularly for the elements x_6 and x_7. Still the two sets are rough fuzzy equivalent as per definition. This is the flexibility provided by this characterization in the upper approximation. But, as $\overline{R}X \neq \overline{R}Y$, the two fuzzy sets are not approximately rough fuzzy equivalent. So, once again we conclude that approximate rough fuzzy equivalence is more realistic than the other three types of approximate equivalences.

3.2 Comparison of Approximate Rough Fuzzy Equalities

3.2.1 The condition that two fuzzy sets are lower approximately equal if and only if the two sets have the same $\alpha-$ support set of lower approximation is satisfied in only those rare and restricted cases where the members have same membership values from and after α. Since we are using this property in case of both rough fuzzy equal and approximately rough fuzzy equal definitions, these two cases of rough fuzzy equalities seem to have lesser utility than the corresponding rough fuzzy equivalences

3.2.2 The condition that the two upper approximations be equal provides freedom to define equality in a very approximate sense and is quite general than these two being equal to U or not simultaneously. As illustrated in the above examples, the later restriction sometimes seems to be less unconvincing.

3.2.3 The concept of approximate rough fuzzy equivalence is neither unconvincing nor unnatural. This is the most natural and best among the four types as provided through examples 3.1 and 3.2 above.

3.2.4 The fourth type of approximate rough fuzzy equality happens to be the worst among the four types of approximate equalities considered.

3.3 Properties of Approximate Equalities of Fuzzy Sets

Properties similar to the basic properties rough sets hold true for fuzzy rough sets. The following four of these properties are to be used by us in establishing the properties of approximate equalities of fuzzy sets.

Let R be an equivalence relation defined over U and $X, Y \in F(U)$. Then

(3.2.1) $\underline{R}(X \cap Y) = \underline{R}(X) \cap \underline{R}(Y)$

(3.2.2) $\underline{R}(X \cup Y) \supseteq \underline{R}(X) \cup \underline{R}(Y)$

(3.2.3) $\overline{R}(X \cap Y) \subseteq \overline{R}(X) \cap \overline{R}(Y)$

(3.2.4) $\overline{R}(X \cup Y) = \overline{R}(X) \cup \overline{R}(Y)$

3.3.1 General Properties

The definitions of bottom-inclusion, top R-inclusion and R-inclusion for crisp sets can be extended in a natural way to fuzzy sets as follows.

Definition 3.3.1.1: Let R be an equivalence relation defined on U and X, Y $\in$ F(U). Then

(i) We say that X is bottom (R, α)-included in Y ($X \tilde{\subseteq}_{R,\alpha} Y$) if and only if $(\underline{R}X)_\alpha \subseteq (\underline{R}Y)_\alpha$

(ii) We say that X is top (R, α)-included in Y $\left(X \tilde{\subset}_{R,\alpha} Y\right)$ if and only if $(\overline{R}X)_\alpha \subseteq (\overline{R}Y)_\alpha$.

(iii) We say that X is (R, α)-included in Y $\left(X \tilde{\subseteq}_{R,\alpha} Y\right)$ if and only if $X \subsetneq_{R,\alpha} Y$ and $X \tilde{\subset}_{R,\alpha} Y$

Definition 3.3.1.2: (i) we say X, Y $\in$ F(U) are $\alpha-$ bottom rough comparable if and only if $X \subsetneq_{R,\alpha} Y$ or $Y \subsetneq_{R,\alpha} X$ holds.

(ii) We say X, Y $\in$ F(U) are $\alpha-$ top rough comparable if and only if $X \tilde{\subset}_{R,\alpha} Y$ or $Y \tilde{\subset}_{R,\alpha} X$ holds.

(iii) We say X, Y $\in$ F(U) are $\alpha-$ rough comparable if and only if X and Y are both top and bottom rough $\alpha-$ comparable.

Definition 3.3.1.3: We say that two fuzzy sets X and Y are $\alpha-$ bottom (rough equal/approximately rough equal, rough equivalent/approximately rough equivalent) if and only if ($(\underline{R}X)_\alpha = (\underline{R}Y)_\alpha$, $(\underline{R}X)_\alpha$ and $(\underline{R}Y)_\alpha$ are equal to φ or not equal to φ together).

Definition 3.3.1.4: We say that two fuzzy sets X and Y are $\alpha-$ top (rough equal/approximately rough equal, rough equivalent/approximately rough equivalent) if and only if ($(\overline{R}X)_\alpha = (\overline{R}Y)_\alpha$, $(\underline{R}X)_\alpha$ and $(\underline{R}Y)_\alpha$ are equal to U or not equal to U together).

We state below the properties of some of the four types of approximate $\alpha-$ rough fuzzy equalities below with out proof. The proofs are similar to the approximate rough equality cases [12].

Property 3.3.1.1: (i) If $X \cap Y$ is $\alpha-$ b_eqv. to both X and Y then X is $\alpha-$ b_eqv. to Y

(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that X and Y are $\alpha-$ bottom rough comparable.

Property 3.3.1.2: (i) If $X \cup Y$ is $\alpha-$ t_eqv. to both X and Y then X is $\alpha-$ t_eqv. to Y.

(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that X and Y are top $\alpha-$ rough comparable.

Property 3.3.1.3: (i) If X is $\alpha-$ t_eqv. to X' and Y is $\alpha-$ t_eqv. to Y' then it may or may not be true that $X \cup Y$ is $\alpha-$ t_eqv. to $X' \cup Y'$.

(ii) A sufficient but not necessary condition for the result in (i) to be true is that X and Y are $\alpha-$ top rough comparable and X' and Y' are $\alpha-$ top rough comparable.

Property 3.3.1.4: (i) X is $\alpha-$ b_eqv. to X' and Y is $\alpha-$ b_eqv. to Y' may or may not imply that $X \cap Y$ is $\alpha-$ b_eqv. to $X' \cap Y'$

(ii) A sufficient but not necessary condition for the result in (i) to be true is that X and Y are $\alpha-$ bottom rough comparable and X' and Y' are $\alpha-$ bottom rough comparable.

Property 3.3.1.5: (i) X is $\alpha-$ t_eqv. to Y may or may not imply that $X \bigcup (-Y)$ is $\alpha-$ t_eqv. to U.
(ii) A sufficient but not necessary condition for result in (i) to hold is that $X \simeq Y$.

Property 3.3.1.6: (i) X is $\alpha-$ b_eqv. to Y may not imply that $X \bigcap (-Y)$ is $\alpha-$ b_eqv. to ϕ.
(ii) A sufficient but not necessary condition for the result in (i) to hold true is that $X \simeq Y$.

Property 3.3.1.7: If $X \subseteq Y$ and Y is $\alpha-$ b_eqv. to ϕ then X is $\alpha-$ b_eqv. to ϕ.

Property 3.3.1.8: If $X \subseteq Y$ and X is t_eqv. to U then Y is t_eqv. to U.

Property 3.3.1.9: X is $\alpha-$ t_eqv. to Y if and only if –X is $\alpha-$ b_eqv. to –Y.

Property 3.3.1.10: X is $\alpha-$ b_eqv. to ϕ, Y is b_eqv. to $\phi \Rightarrow X \bigcap Y$ is $\alpha-$ b_eqv. to ϕ.

Property 3.3.1.11: If X is $\alpha-$ t_eqv. to U or Y is t_eqv. to U then $X \bigcup Y$ is $\alpha-$ t_eqv. to U.

3.3.2 Replacement Properties for Fuzzy Rough Equivalence

In parallel to the properties of interchange for rough equalities, the following properties were proved in [12].

Property 3.3.2.1: (i) If $X \bigcap Y$ is $\alpha-$ t_eqv. to both X and Y then X is $\alpha-$ t_eqv. Y.
(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that equality holds in (3.2.3).

Property 3.3.2.2: (i) $X \bigcup Y$ is $\alpha-$ b_eqv. to X and $X \bigcup Y$ is $\alpha-$ b_eqv. to Y then X is $\alpha-$ b_eqv. to Y
(ii) The converse of (i) is not true in general and an additional condition that is sufficient but not necessary for the converse to be true is that equality holds in (3.2.2).

Property 3.3.2.3: (i) X is $\alpha-$ b_eqv. to X' and Y is $\alpha-$ b_eqv. to Y' may not imply $X \bigcup Y$ is $\alpha-$ b_eqv. to $X' \bigcup Y'$.
(ii) A sufficient but not necessary condition for the conclusion of (i) to hold is that equality holds in (3.2.3).

Property 3.3.2.4: (i) X is $\alpha-$t_eqv. to X' and Y is $\alpha-$t_eqv. to Y' may not necessarily imply that $X \cap Y$ is $\alpha-$t_eqv. to $X' \cap Y'$.
(ii) A sufficient but not necessary condition for the conclusion in (i) to hold is that equality holds in (3.2.2).

Property 3.3.2.5: X is $\alpha-$b_eqv. to Y may or may not imply $X \cup -Y$ is $\alpha-$b_eqv. to U.

Property 3.3.2.6: X is $\alpha-$t_eqv. to Y may or may not imply $X \cap -Y$ is $\alpha-$t_eqv. to ϕ.

Properties (3.2.2.7) to (3.2.2.11) hold true under replacements.

4 Conclusions

In this paper we introduced the concept of leveled approximate equality by taking rough fuzzy sets instead of rough sets alone in defining approximate equality. This notion has two advantages. First, it considers the approximate equality of fuzzy sets instead of crisp sets. Second, we introduce the notion of leveled approximate equality, which provides a threshold value for the approximate equality. This provides a control with the user to specify the desired level equality needed. So that, according to the requirement this value can be adjusted to get the desired results. These notions are more realistic than those considered earlier by Novotny and Pawlak [7, 8, 9] and Tripathy [10, 11, 12].

References

[1] Dubois, D., Prade, H.: Rough fuzzy sets model. International Journal of General Systems 46(1), 191–208 (1990)
[2] Pawlak, Z.: Rough sets. Int. Jour. of Computer and Information Sciences 11, 341–356 (1982)
[3] Pawlak, Z.: Rough sets: Theoretical aspects of reasoning about data. Kluwer academic publishers, London (1991)
[4] Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences-An International Journal 177(1), 3–27 (2007)
[5] Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences-An International Journal 177(1), 28–40 (2007)
[6] Pawlak, Z., Skowron, A.: Rough sets and Boolean reasoning. Information Sciences-An International Journal 177(1), 41–73 (2007)
[7] Novotny, M., Pawlak, Z.: Characterization of Rough Top equalities and Rough Bottom Equalities. Bull. Polish Acad. Sci. Math. 33, 91–97 (1985)
[8] Novotny, M., Pawlak, Z.: On Rough Equalities. Bull. Polish Acad. Sci. Math. 33, 99–104 (1985)
[9] Novotny, M., Pawlak, Z.: Black Box Analysis and Rough Top Equality. Bull. Polish Acad. Sci. Math. 33, 105–113 (1985)
[10] Tripathy, B.K.: On Approximation of Classifications, Rough Equalities and Rough Equivalences. In: Abraham, A., Falcon, R., Bello, R. (eds.) Rough Set Theory. SCI, vol. 174, pp. 85–133. Springer, Heidelberg (2009)

[11] Tripathy, B.K.: An Analysis of Approximate Equalities based on Rough Set Theory. Intl. Journal of Advanced Science and Technology 31, 23–36 (2011)
[12] Tripathy, B.K., Mitra, A., Ojha, J.: On Rough Equalities and Rough Equivalences of Sets. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 92–102. Springer, Heidelberg (2008)
[13] Tripathy, B.K., Mitra, A., Ojha, J.: Rough Equivalence and Algebraic properties of Rough Sets. International Journal of Artificial Intelligence and Soft Computing 1(2/3/4), 271–289 (2009)
[14] Tripathy, B.K., Panda, G.K., Mitra, A.: Covering Based Rough Equality of sets and Comparison of Knowledge. In: Proceedings of the Inter. Conf. in Mathematics and Computer Science (ICMCS 2009), Chennai, INDIA, January 5-6, vol. 2, pp. 438–443 (2009)
[15] Tripathy, B.K., Tripathy, H.K.: Covering Based Rough equivalence of Sets and Comparison of Knowledge. In: Proceedings of the IACSIT Spring Conference 2009, Singapore, April 17-20, pp. 303–307 (2009)
[16] Zadeh, L.A.: Fuzzy Sets. Information and Control 8(11), 338–353 (1965)

Nondifferentiable Multiobjective Wolfe Type Symmetric Duality under Invexity

T.R. Gulati and Khushboo Verma

Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee-247 667, India
{trgulati,1986khushi}@gmail.com

Abstract. In the present paper, a pair of Wolfe type nondifferentiable mutiobjective second-order programm involving two kernel function is formulated. We prove Weak, Strong and Converse duality theorem for the second-order symmetric dual programs under Invexity conditions.

Keywords: Nondifferentiable Multiobjective programming, Second-order symmetric duality, Proper efficiency, Invexity.

1 Introduction

Symmetric duality in nonlinear programming, in which dual of the dual is the primal problems, was first introduced by Dorn [4]. He introduced the concept of symmetric duality for quadratic problems. Later on Dantzig et al. [3] formulated a pair of symmetric dual nonlinear programs involving convex/concave functions. These results were extended by Bazaraa and Goode [2] over arbitrary cones.

Mangasarian [7] introduced the concept of second-order duality for nonlinear problems. Suneja et al. [9] studied a pair of Mond-Weir type second-order multiobjective symmetric dual programs and proved the duality results under η-bonvexity/η-pseudobonvexity assumptions. Yang et al. [10] extended the results of Suneja et al. [9] to the nondifferentiable case. Second order symmetric duality for Mond-Weir type duals involving nondifferentiable function has been discussed by Hou and Yang [6] and by Ahmad and Husain [1].

The symmetric dual problems in the above papers involve only one kernel function. Recently, Wolfe and Mond-Weir type second-order differentiable symmetric dual models involving two kernel functions have been studied in [5]. In this paper, we present nondifferentiable symmetric dual multiobjective problems involving two kernel functions.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 347–354.
springerlink.com

2 Prerequisites

We consider the following multiobjective programming problem :

(P) Minimize

$$K(x) = \{K_1(x), K_2(x), ..., K_k(x)\}$$

$$\text{subject to :} \quad x \in X = \{x \in R^n \mid G_j(x) \leqq 0, j = 1,2,...,m\},$$

where $G : R^n \to R^m$ and $K : R^n \to R^k$.

All vectors shall be considered as column vectors. For a function $f : R^n \times R^m \to R^k$, let $\nabla_x f(\nabla_y f)$ denote the $n \times k$ $(m \times k)$ matrix of first order derivative and $\nabla_{xy} f_i$ denote the $n \times m$ matrix of second order derivative.

For a, b $\in R^n$,
a $\geqq b \Leftrightarrow a_i \geqq b_i, i = 1,2,...,n$,
a $\geqslant b \Leftrightarrow a \geqq b$ and $a \neq b$,
$a > b \Leftrightarrow a_i > b_i, i = 1,2,...,n$.

Definition 2.1. A point $\bar{x} \in X$ is said to be an efficient solution of (P) if there exists no $x \in X$ such that $K(x) \leq K(\bar{x})$.

Definition 2.2 [5]**.** The function K is $\eta-$invex at $u \in R^n$ if there exists a vector valued function $\eta : R^n \times R^n \to R^n$ such that

$$K(x) - K(u) - \eta^T(x,u)\nabla K(u) \geqq 0, \forall x \in R^n.$$

Definition 2.3. (Generalized Schwartz Inequality) Let A $\in R^n \times R^n$ be a positive semi-definite matrix. Then for all $x, y \in R^n$,

$$x^T Ay \leqq (x^T Ax)^{1/2}(y^T Ay)^{1/2}.$$

Equality holds if $Ax = \lambda Ay$ for some $\lambda \geqq 0$.

3 Wolfe Type Symmetric Duality

We now consider the following pair of Wolfe type nondifferentiable multiobjective programming problems.

Primal (WP)

Minimize $H(x,y,\lambda,h,p) = f(x,y) + (x^T Dx)^{1/2}e - (y^T \nabla_y(\lambda^T f(x,y)))e$

$$-(y^T(\nabla_{yy}(h^T g(x,y))p))e$$

$$\text{Subject to} \quad \nabla_y(\lambda^T f(x,y)) - Ew + \nabla_{yy}(h^T g(x,y))p \leqq 0, \tag{1}$$

$$w^T E w \leqq 1, \tag{2}$$

$$\lambda^T e = 1, \tag{3}$$

$$\lambda > 0, x \geqq 0, w \in R^m, \tag{4}$$

Dual (WD)

Maximise $G(u,v,\lambda,h,q) = f(u,v) - (v^T E v)^{1/2} e - (u^T \nabla_x(\lambda^T f(u,v)))e$

$$-(u^T(\nabla_{xx}(h^T g(u,v))q))e$$

$$\text{Subject to} \quad \nabla_x(\lambda^T f(u,v)) + Dz + \nabla_{xx}(h^T g(u,v))q \geqq 0, \tag{5}$$

$$z^T D z \leqq 1, \tag{6}$$

$$\lambda^T e = 1, \tag{7}$$

$$\lambda > 0, v \geqq 0, z \in R^n, \tag{8}$$

where
(i) $f : R^n \times R^m \to R^k$ is a twice differentiable function of x and y,
(ii) $g : R^n \times R^m \to R^r$ is a thrice differentiable function of x and y,
(iii) $p \in R^m$, $q \in R^n$, $e = (1,...,1)^T \in R^k$, and
(iv) D and E are positive semi-definite matrices of order n and m, respectively.

Any problem, say (WD), in which λ is fixed to be $\bar{\lambda}$, will be denoted by $(\text{WD})_{\bar{\lambda}}$.

Theorem 3.1 (Weak duality)**.** Let (x,y,λ,h,w,p) be feasible for (WP) and (u,v,λ,h,z,q) be feasible for (WD). Let

(i) $f(.,v) + ((.)^T Dz)e$ be η_1-invex at u for fixed v and z,
(ii) $f(x,.) - ((.)^T Ew)e$ be η_2-invex at y for fixed x and w,
(iii) $\eta_1(x,u) + u \geqq 0$ and $\eta_2(v,y) + y \geqq 0$, and
(iv) $\eta_1^T(x,u)(\nabla_{xx}(h^T g)(u,v)q) \leqq 0$ and $\eta_2^T(v,y)(\nabla_{yy}(h^T g)(x,y)p) \geqq 0$.

Then

$$\{f(u,v) + (v^T E v)^{1/2} e - (u^T \nabla_x(\lambda^T f(u,v)))e - (u^T \nabla_{xx}(h^T g(u,v))q)e$$

$$\not\geq f(x,y) + (x^T D x)^{1/2} e - (y^T \nabla_y(\lambda^T f(x,y)))e - (y^T \nabla_{yy}(h^T g(x,y))p)e\}.$$

Proof. Suppose, to the contrary, that

$$\{f(u,v) + (v^T E v)^{1/2} e - (u^T \nabla_x(\lambda^T f(u,v)))e - (u^T \nabla_{xx}(h^T g(u,v))q)e$$

$$\geq f(x,y) + (x^T D x)^{1/2} e - (y^T \nabla_y(\lambda^T f(x,y)))e - (y^T \nabla_{yy}(h^T g(x,y))p)e\}$$

Since $\lambda > 0$ and $\lambda^T e = 1$, the above vector inequality implies

$$\lambda^T[f(u,v) + (v^T Ev)^{1/2}e - (u^T\nabla_x(\lambda^T f(u,v)))e - (u^T\nabla_{xx}(h^T g(u,v))q)e]$$

$$> \lambda^T[f(x,y) + (x^T Dx)^{1/2}e - (y^T\nabla_y(\lambda^T f(x,y)))e - (y^T\nabla_{yy}(h^T g(x,y))p)e]$$

or

$$\lambda^T f(u,v) + (v^T Ev)^{1/2} - u^T\nabla_x(\lambda^T f(u,v)) - u^T\nabla_{xx}(h^T g(u,v))q$$

$$> \lambda^T f(x,y) + (x^T Dx)^{1/2} - y^T\nabla_y(\lambda^T f(x,y)) - y^T\nabla_{yy}(h^T g(x,y))p. \tag{9}$$

From η_1-invexity of $f(.,v) + ((.)^T Dz)e$, we have

$$f(x,v) + (x^T Dz)e - f(u,v) - (u^T Dz)e \geqq (\nabla_x^T f(u,v) + e(z^T D))\eta_1(x,u).$$

Using $\lambda > 0$, we obtain

$$(\lambda^T f)(x,v) + (x^T Dz) - (\lambda^T f)(u,v) - (u^T Dz)$$

$$\geqq \eta_1(x,u)^T(\nabla_x(\lambda^T f)(u,v) + Dz). \tag{10}$$

From the dual constraint (5) and hypothesis (iii), it follows that

$$\eta_1^T(x,u)(\nabla_x(\lambda^T f(u,v)) + Dz + \nabla_{xx}(h^T g(u,v))q)$$

$$\geqq -u^T(\nabla_x(\lambda^T f(u,v)) + Dz + \nabla_{xx}(h^T g(u,v))q). \tag{11}$$

Now inequalities (10), (11) along with hypothesis(iv), yield

$$(\lambda^T f)(x,v) + (x^T Dz) - (\lambda^T f)(u,v) - (u^T Dz)$$

$$\geqq -u^T(\nabla_x(\lambda^T f)(u,v) + Dz + \nabla_{xx}(h^T g(u,v))q).$$

or

$$(\lambda^T f)(x,v) + (x^T Dz) - (\lambda^T f)(u,v)$$

$$\geqq -u^T(\nabla_x(\lambda^T f)(u,v) + \nabla_{xx}(h^T g(u,v))q). \tag{12}$$

Similarly by η_2-invexity of $f(x,.) - ((.)^T Ew)e$, the primal constraints (1) and hypotheses (iii) and (iv), we obtain

$$(\lambda^T f)(x,y) - (\lambda^T f)(x,v) + (v^T Ew) \geqq y^T(\nabla_y(\lambda^T f(x,y)) + \nabla_{yy}(h^T g(x,y))p). \tag{13}$$

Adding inequalities (12) and (13), we get

$$(\lambda^T f)(x,y) + (x^T Dz) - (y^T\nabla_y(\lambda^T f(x,y))) - (y^T\nabla_{yy}(h^T g(x,y))p)$$

$$\geqq (\lambda^T f)(u,v) + (v^T Ew) - (u^T\nabla_x(\lambda^T f(u,v))) - (u^T\nabla_{xx}(h^T g(u,v))q).$$

Now using the Schwartz inequality and the constraints (2) and (6),

$$(\lambda^T f)(x,y) + (x^T Dx)^{1/2} - (y^T \nabla_y(\lambda^T f(x,y))) - (y^T \nabla_{yy}(h^T g(x,y))p) \geqq$$

$$(\lambda^T f)(u,v) + (v^T Ev)^{1/2} - (u^T \nabla_x(\lambda^T f(u,v))) - (u^T \nabla_{xx}(h^T g(u,v))q),$$

which contradicts inequality (9). Thus the result holds. □

Theorem 3.2 (Strong duality)**.** Let $(\bar{x},\bar{y},\bar{\lambda},\bar{h},\bar{w},\bar{p})$ be an efficient solution for (WP). Suppose that

(i) $\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})$ is nonsingular,
(ii) the set $\{\nabla_y f_i(\bar{x},\bar{y}), i = 1,\ldots,k\}$ is linearly independent, and
(iii) $\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p} \notin \text{span}\{\nabla_y f_1(\bar{x},\bar{y}),\ldots,\nabla_y f_k(\bar{x},\bar{y})\}\backslash\{0\}$.

Then $(\bar{x},\bar{y},\bar{h},\bar{z},\bar{q}=0)$ is feasible for $(\text{WD})_{\bar{\lambda}}$, and the objective function values of (WP) and $(\text{WD})_{\bar{\lambda}}$ are equal. Also, if the hypotheses of Theorem 3.1 are satisfied for all feasible solutions of $(\text{WP})_{\bar{\lambda}}$ and $(\text{WD})_{\bar{\lambda}}$, then $(\bar{x},\bar{y},\bar{h},\bar{z},\bar{q}=0)$ is an efficient solution for $(\text{WD})_{\bar{\lambda}}$.

Proof. Since $(\bar{x},\bar{y},\bar{\lambda},\bar{h},\bar{p})$ is an efficient solution for (WP), by the Fritz-John necessary optimality conditions [8], there exist $\bar{\alpha} \in R^k$, $\bar{\beta} \in R^m$, $\bar{\gamma},\bar{\delta} \in R$, $\bar{\xi} \in R^k$ and $\bar{\eta},\bar{z} \in R^n$ such that the following conditions are satisfied at $(\bar{x},\bar{y},\bar{\lambda},\bar{h},\bar{w},\bar{p})$:

$$(\nabla_x f(\bar{x},\bar{y}) + (D\bar{z})e)\bar{\alpha} + (\nabla_{yx}(\bar{\lambda}^T f)(\bar{x},\bar{y})$$

$$+\nabla_x(\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p}))^T(\bar{\beta} - (\bar{\alpha}^T e)\bar{y}) - \bar{\eta} = 0 \tag{14}$$

$$\nabla_y f(\bar{x},\bar{y})(\bar{\alpha} - (\bar{\alpha}^T e)\bar{\lambda}) + [(\nabla_{yy}(\bar{\lambda}^T f)(\bar{x},\bar{y})$$

$$+\nabla_y(\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p})](\bar{\beta} - (\bar{\alpha}^T e)\bar{y}) - (\bar{\alpha}^T e)\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p} = 0, \tag{15}$$

$$(\bar{\beta} - (\bar{\alpha}^T e)\bar{y})^T \nabla_y f(\bar{x},\bar{y}) + \bar{\delta}e^T - \bar{\xi} = 0, \tag{16}$$

$$(\bar{\beta} - (\bar{\alpha}^T e)\bar{y})^T \nabla_h(\nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p}) = 0, \tag{17}$$

$$(\bar{\beta} - (\bar{\alpha}^T e)\bar{y})^T \nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y}) = 0, \tag{18}$$

$$-E\bar{\beta} + \bar{\gamma}E\bar{w} = 0, \tag{19}$$

$$\bar{\gamma}(\bar{w}^T E\bar{w} - 1) = 0, \tag{20}$$

$$\bar{\delta}(\bar{\lambda}^T e - 1) = 0, \tag{21}$$

$$\bar{\lambda}^T \bar{\xi} = 0, \tag{22}$$

$$\bar{x}^T \bar{\eta} = 0, \tag{23}$$

$$\bar{x}^T D\bar{z} = (\bar{x}^T D\bar{x})^{1/2}, \tag{24}$$

$$\bar{\beta}^T[\nabla_y(\bar{\lambda}^T f)(\bar{x},\bar{y}) - E\bar{w} + \nabla_{yy}(\bar{h}^T g)(\bar{x},\bar{y})\bar{p}] = 0, \tag{25}$$

$$\bar{z}^T D\bar{z} \leqq 1, \tag{26}$$

$$(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\xi}, \bar{\eta}) \geqq 0, \tag{27}$$

$$(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\delta}, \bar{\xi}, \bar{\eta}) \neq 0. \tag{28}$$

As $\bar{\lambda} > 0$, from (22) we conclude that $\bar{\xi} = 0$. By hypothesis (i), equation (18) implies

$$\bar{\beta} = (\bar{\alpha}^T e)\bar{y}. \tag{29}$$

Now suppose, $\alpha = 0$. Then equation (29) implies, $\bar{\beta} = 0$, which along with Eq. (16) yields $\bar{\delta} e = 0$ or $\bar{\delta} = 0$. Also equation(14) implies that $\bar{\eta} = 0$.

From equations (19) and (20), we have

$$\bar{\gamma} = \bar{\gamma}(\bar{w}^T E w) = \bar{w}^T(\bar{\gamma} E \bar{w}) = \bar{w}^T(E\bar{\beta}) = 0.$$

Thus $(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\delta}, \bar{\xi}, \bar{\eta}) = 0$, which contradicts (28). Hence $\bar{\alpha} \neq 0$, i.e., $\bar{\alpha} \geq 0$ or

$$\bar{\alpha}^T e > 0. \tag{30}$$

Therefore equations (29) and (30) yields

$$\bar{y} = \frac{\beta}{\alpha^T e} \geqq 0.$$

Now, from (17) and (29), we have

$$\nabla_y f(\bar{x}, \bar{y})(\bar{\alpha} - (\bar{\alpha}^T e)\bar{\lambda}) = (\bar{\alpha}^T e)\nabla_{yy}(\bar{h}^T g)(\bar{x}, \bar{y})\bar{p}. \tag{31}$$

Using the hypothesis (iii), the above relation implies $(\alpha^T e)\nabla_{yy}(\bar{h}^T g)(\bar{x}, \bar{y})\bar{p} = 0$, which by hypothesis (i) and (30) yield

$$\bar{p} = 0. \tag{32}$$

Therefore equation (31) gives

$$(\nabla_y f)(\bar{x}, \bar{y})(\alpha - (\alpha^T e)\bar{\lambda}) = 0.$$

Since the set $\{\nabla_y f_i, i = 1, \ldots, k\}$ is linearly independent, the above equation implies

$$\bar{\alpha} = (\bar{\alpha}^T e)\bar{\lambda}. \tag{33}$$

Using (29), (30) and (33) in (14), we get

$$\nabla_x((\bar{\lambda}^T f)(\bar{x}, \bar{y}) + D\bar{z}) = \eta \geqq 0. \tag{34}$$

Thus $(\bar{x}, \bar{y}, \bar{h}, \bar{z}, \bar{q} = 0)$ is a feasible solution for the dual problem $(\mathrm{WD})_{\bar{\lambda}}$.
Now from equation (34),

$$\bar{x}^T(\nabla_x((\bar{\lambda}^T f)(\bar{x}, \bar{y}) + D\bar{z}) = \bar{\eta}^T \bar{x} = 0$$

or using (24)

$$x^T(\nabla_x((\bar{\lambda}^T f)(\bar{x},\bar{y}) = -\bar{x}^T D\bar{z} = -(x^T Dx)^{1/2}. \tag{35}$$

Further, from (25), (29), (30) and (32), we obtain

$$\bar{y}^T \nabla_y(\bar{\lambda}^T f)(\bar{x},\bar{y}) = \bar{y}^T E\bar{w}. \tag{36}$$

From equations (19) and (29), we have

$$E\bar{y} = \frac{\bar{\gamma}}{\bar{\alpha}^T e} E\bar{w} \tag{37}$$

or

$$E\bar{y} = aE\bar{w},$$

where

$$a = \frac{\bar{\gamma}}{\bar{\alpha}^T e} \geqq 0.$$

Under this condition the Schwarz inequality holds as an equality.

Therefore $\bar{y}^T E\bar{w} = (\bar{y}^T E\bar{y})^{1/2}(\bar{w}^T E\bar{w})^{1/2}$. In case $\bar{\gamma} > 0$, Equation (22) implies $\bar{w}^T E\bar{w} = 1$ and so $\bar{y}^T E\bar{w} = (\bar{y}^T E\bar{y})^{1/2}$. When $\bar{\gamma} = 0$, equation (37) gives $E\bar{y} = 0$ and so $\bar{y}^T E\bar{w} = 0 = (\bar{y}^T E\bar{y})^{1/2}$.
Thus in either case

$$\bar{y}^T E\bar{w} = (\bar{y}^T E\bar{y})^{1/2},$$

and so Equation (36) becomes

$$\bar{y}^T \nabla_y(\bar{\lambda}^T f)(\bar{x},\bar{y}) = \bar{y}^T E\bar{w} = (\bar{y}^T E\bar{y})^{1/2}. \tag{38}$$

Finally using (35) and (38), we get

$$\begin{aligned} H(\bar{x},\bar{y},\bar{\lambda},\bar{h},\bar{p}=0) &= f(\bar{x},\bar{y}) + (\bar{x}^T D\bar{x})^{1/2} e - (\bar{y}^T \nabla_y(\bar{\lambda}^T f)(x,y))e \\ &= f(\bar{x},\bar{y}) - (\bar{x}^T \nabla_x(\bar{\lambda}^T f)(x,y))e - (\bar{y}^T E\bar{y})^{1/2} e \\ &= G(\bar{x},\bar{y},\bar{\lambda},\bar{h},\bar{q}=0). \end{aligned}$$

That is two objective function values are equal. Using weak duality it can be easily shown that $(\bar{x},\bar{y},\bar{h},\bar{z},\bar{q}=0)$ is an efficient solution of $(\mathrm{WD})_{\bar{\lambda}}$ □

Theorem 3.3 (Converse duality)**.** Let $(\bar{u},\bar{v},\bar{\lambda},\bar{h},\bar{z},\bar{q})$ be an efficient solution for (WD). Suppose that

(i) $\nabla_{xx}(\bar{h}^T g)(\bar{u},\bar{v})$ is nonsingular,
(ii) the set $\{\nabla_x f_i(\bar{u},\bar{v}), i = 1,\ldots,k\}$ is linearly independent, and
(iii) $\nabla_{xx}(\bar{h}^T g)(\bar{u},\bar{v})\bar{q} \notin \mathrm{span}\{\nabla_x f_1(\bar{u},\bar{v}),...,\nabla_x f_k(\bar{u},\bar{v})\}\backslash\{0\}$.

Then $(\bar{u},\bar{v},\bar{h},\bar{z},\bar{p}=0)$ is feasible for $(\mathrm{WP})_{\bar{\lambda}}$, and the objective function values of $(\mathrm{WP})_{\bar{\lambda}}$ and (WD) are equal. Also, if the hypotheses of Theorem 3.1 are satisfied for all feasible solutions of $(\mathrm{WP})_{\bar{\lambda}}$ and $(\mathrm{WD})_{\bar{\lambda}}$, then $(\bar{u},\bar{v},\bar{h},\bar{w},\bar{q})$ is an efficient solution for $(\mathrm{WP})_{\bar{\lambda}}$.

Proof. Follows on the lines of Theorem 3.2. □

Acknowledgement. The second author is thankful to the MHRD, Government of India for providing financial support.

References

1. Ahmad, I., Husain, Z.: Nondifferentiable second order symmetric duality in multiobjective programming. Appl. Math. Lett. 18, 721–728 (2005)
2. Bazaraa, M.S., Goode, J.J.: On symmetric duality in nonlinear programming. Oper. Res. 21, 1–9 (1973)
3. Dantzig, G.B., Eisenberg, E., Cottle, R.W.: Symmetric dual nonlinear programming. Pac. J. Math. 15, 809–812 (1965)
4. Dorn, W.S.: A symmetric dual theorem for quadratic programming. J. Oper. Res. Soc. Jpn. 2, 93–97 (1960)
5. Gulati, T.R., Geeta: Mond-Weir type second-order symmetric dualty in multiobjective programming over cones. Appl. Math. Lett. 23, 466–471 (2010)
6. Hou, S.H., Yang, X.M.: On second-order symmetric duality in nondifferentiable programming. J. Math. Anal. Appl. 255, 491–498 (2001)
7. Mangasarian, O.L.: Second and higher-order duality in nonlinear programming. J. Math. Anal. Appl. 51, 607–620 (1975)
8. Schechter, M.: More on subgradient duality. J. Math. Anal. Appl. 71, 251–261 (1979)
9. Suneja, S.K., Lalitha, C.S., Khurana, S.: Second order symmetric duality in multiobjective programming. Eur. J. Oper. Res. 144, 492–500 (2003)
10. Yang, X.M., Yang, X.Q., Teo, K.L., Hou, S.H.: Second order symmetric duality in nondifferentiable multiobjective programming with F-convexity. Eur. J. Oper. Res. 164, 406–416 (2005)

Multi-robot Box-Pushing Using Differential Evolution Algorithm for Multiobjective Optimization

Pratyusha Rakshit[1], Arup Kumar Sadhu[1], Anisha Halder[1], Amit Konar[1], and R. Janarthanan[2]

[1] ETCE Dept, Jadavpur University, Kolkata-700032, India
[2] Jaya Engineering College, Chennai
{pratyushar1,arup.kaajal,halder.anisha}@gmail.com,
konaramit@yahoo.co.in, srmjana_73@yahoo.com

Abstract. The paper provides a new approach to multi-robot box pushing using a proposed Differential evolution for multiobjective optimization (DEMO) algorithm. The proposed scheme determines time-, energy- and friction sensitive-optimal solution to the box-pushing problem. The performance of the developed DEMO algorithm is compared to NSGA-II in connection with the given problem and the experimental results reveal that the DEMO outperforms NSGA-II in all the experimental set-ups.

Keywords: multi-robot box pushing, friction compensation, differential evolution for multiobjective optimization.

1 Introduction

Optimization of multi-objective functions is essential to many engineering problems. The box pushing problem, dealt with in this paper, is related to the well known "Piano Mover's Problem": given an arbitrary rigid polyhedral environment, determine a continuous collision-free trajectory of motion of the object from a source configuration to a desired destination configuration.

The box-pushing problem represents a challenging domain for the study of object manipulation in a multi-robot environment. Since 1990's researchers took active interest in formulating and solving the box-pushing problem by different techniques. Some of the well known works in this regard include adaptive action selection by the robots without communication [7], mutual cooperation by intension inference [8], cooperative conveyance by velocity adaptation of robots [9], and role of perceptual cues in multi-robot box-pushing [6].

One of the most studied strategies is the well-known "Pusher-Watcher" approach [13, 14]: A robot observes (watcher) the movement of the object and

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 355–365.
springerlink.com

control the operations of the team (pushers) that manipulate the box. Recent multi-robot strategies make use of the model known as "swarm intelligence". Li [15] experiments using communities of homogenous robots. These self-organized systems are based on decentralized and collective behaviors. Another recent strategy is the reinforcement learning. Wang [16] implements a variant including a mechanism of decision based on a Markov process known as "Q-Learning". The main concern about this technique is related to the storage capacities and high demand of process capabilities. Gene [17] and Lengegyel [18] model the environment by the construction of a configuration space (C-space) and both use the conventional wave front algorithm to compute the trajectories of the box.

In the proposed work, two similar robots have to locally plan the trajectory of motion of the box in a complex terrain, where the robot's workspace has different frictional coefficients in different regions. In [2], it has been attempted to satisfy multiple objectives concerning minimization of both time and energy required in local trajectory planning of the robots by NSGA-II.

The work proposed in this paper is different by two counts. First, we consider the frictional forces offered by the contact surface to the box. The forces applied by robots must be sufficient to counteract this frictional force. Second, the box-pushing problem has been solved by Differential evolution for multiobjective optimization (DEMO), pioneered by Robic and Filipi [1] as a state-of-the-art literature survey indicates that DE has already proved itself as a promising candidate in the field of evolutionary multi-objective optimization (EMO) [10, 11, 12].

The justification of the use of DEMO in the proposed problem are point wise indicated below: 1) better solution quality 2) efficiently achieving the two goals of multiobjective optimization, i.e., the convergence to the true Pareto front and uniform spread of individuals along the front. It has also been verified in the paper that DEMO has outperformed NSGA-II in trajectory planning of the robots in all the different types of workspace. It is also compared with Multiobjective optimization using differential evolution (MODE) - based simulation and it has been proved that DEMO is better than MODE.

The remaining paper has been organized into sections as follows: In Section-2, a formulation of the problem is presented. In Section-3, a pseudo-code for solving the optimization problem using DEMO is provided and in Section-4, computer simulation for the problem has been laid down for comparison of results using DEMO and NSGA-II.

2 Formulation of the Problem

Suppose two robots R_1 and R_2 are applying forces perpendicularly at points E(x_e, y_e) and F(x_f, y_f) on the front edge BC of the box (Fig. 1). Let O (x_c, y_c) be the centre of gravity of the box . After being rotated by an angle α around the point $I(x_I, y_I)$ due to the forces applied by robots $R_1(F_{1r})$ and $R_2(F_{2r})$ the corresponding new co-ordinates of O, E and F become

$$\left.\begin{aligned} x_{inew} &= x_I(1-\cos\alpha) + x_i\cos\alpha - \sin\alpha\,(y_i - y_I) \\ y_{inew} &= y_I(1-\cos\alpha) + y_i\cos\alpha - \sin\alpha\,(x_i - x_I) \end{aligned}\right\} \text{ where, } i \in P = \{o, e, f\}. \qquad (1)$$

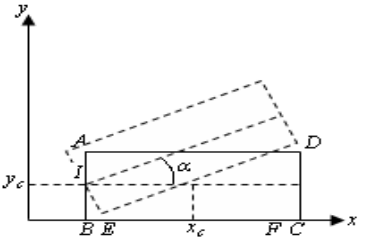


Fig. 1. Position of the box before (solid line) and after rotation (dashed line)

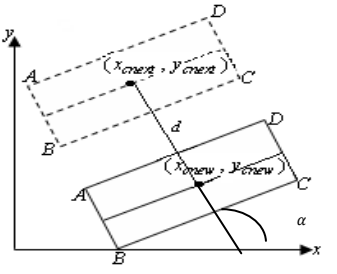


Fig. 2. Current (solid line) and next position (dashed line) of the box after rotation

Now, the box is displaced by d unit (Fig.2) due to transportation forces applied by $R_1(\ F_{1t})$ and $R_2(\ F_{2t})$. The new position of the centre of the gravity is given as

$$x_c^{'} = x_{cnew} + d\cos\alpha\,;\ y_c^{'} = y_{cnew} + d\sin\alpha \tag{2}$$

We, now form two objective functions concerning minimization of time and energy [2], which have three components each, time and energy required for rotation (t_1, E_1), for translation of the box to the next position (t_2, E_2), and predicted cost required for the transportation of the box from the next to the goal position through a distance s (t_3, E_3). The total time required (f_1) and total energy consumption $(f_2)$are obtained as

$$f_1 = t_1 + t_2 + t_3 \tag{3}$$

$$f_2 = E_1 + E_2 + E_3 \tag{4}$$

$$t_1 = \sqrt{\frac{2\alpha J}{T}},\ t_2 = \sqrt{\frac{2md}{F_{1t}+F_{2t}}},\ t_3 = k_t\sqrt{s};\ E_1 = T\alpha,\ E_2 = (F_{1t}+F_{2t})d\ ,\ E_3 = k_e s \tag{5}$$

where, J=mass moment of inertia, T= Torque, m=mass of the box, k_t and k_e are constants.

In order to keep the distance of the nearest obstacle dis_obs in the direction of movement as high as possible we introduce one penalty term in f_1.

$$f_1 = t_1 + t_2 + t_3 + f_{st}/dis\_obs \quad \text{with} \quad f_{st} = 500. \tag{6}$$

The third objective function is about minimization of frictional energy consumption during translation of the box in the workspace having different frictional coefficients in different regions (Fig. 3) and is denoted by, f_3 where

$$f_3 = F_c \times d \ \text{ where, } F_c = \text{Coulomb force of friction} = \sum_{i=1}^{n} \mu_i \times m_i \times g. \tag{7}$$

where, n = total number of point contacts between the two surfaces in contact which is dependent on dimension of the box and μ_i and m_i represent the frictional coefficient of the surface and mass of the box at that point of contact.

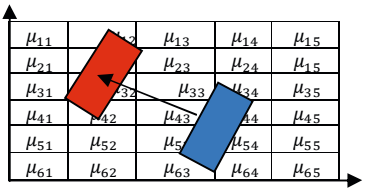


Fig. 3. Initial (blue) and final (red) position of the box in workspace with different frictional co-efficient

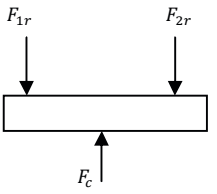


Fig. 4. Forces applied on the box by robots (F_{r1} and F_{r2}) and contact surface (F_c)

Considering the constraint that when the robots are rotating and translating the box, the combined force of the two robots must be greater than the frictional force offered by the surface in contact with the box, which is assumed to be acting on the centre of gravity of the box as (Fig. 4), we have the required condition as

$$F_{1r} + F_{2r} > F_c \tag{8}$$

So, with $d$unit displacement of the equation (7) has been modified to

$$f_3 = \sum_{i=1}^{n} \mu_i \times m_i \times g \times d + f_{fp}/(F_{1r} + F_{2r} - F_c) \tag{9}$$

where, g= acceleration due to gravity, and f_{fp} is a constant which is taken as 100. f_{st} and f_{st} are determined in a manner so that the terms on the right hand side of (9) 1nd (6) should be of same order of magnitude.

3 Solving Constraint Optimization Problem Using Differential Evolution for Multiobjective Optimization (DEMO) Algorithm

DE [3] is a population-based global optimization algorithm that uses a real-coded representation. It generates a population of uniformly distributed *Np* real-coded *D* dimensional search variable vectors$X_{i,G} = [x_{i,1,G}, x_{i,2,G}, \ldots, x_{i,D,G}]$. Now, in each generation, a donor $V_{i,G}$ is created in the following way:

$$V_{i.G} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G}) \text{ where, } r_1, r_2, r_3 \in [1, N_p], r_1 \neq r_2 \neq r_3. \tag{10}$$

Next, we use 'binomial' crossover to generate a trial vector $U_{i,G}$ for each target vector $X_{i,G}$ in which case the number of parameters inherited from the mutant has a (nearly) binomial distribution.

$$U_{i,j,G} = \begin{cases} V_{i,j,G}, if\ rand_{i,j}(0,1) < Cr\ or\ j = j_{rand} \\ X_{i,j,G}, \qquad otherwise \end{cases} \tag{11}$$

for $rand_{i,j}(0,1) \in [0, 1]$, $j_{rand} \in [1,2,\ldots., D]$ is a randomly chosen index which ensures that $U_{i,G}$ gets at least one component from $V_{i,G}$. To keep the population size constant over subsequent generations, the next step of the algorithm calls for

'selection' in order to determine which one between the target vector and trial vector will survive in the next generation i.e. at the next generation G = G+1. If the trial vector yields a better value of the fitness function, it replaces its target vector in the next generation; otherwise the parent is retained in the population:

$$X_{i,G+1} = \begin{cases} U_{i,G}, \text{if } f(U_{i,G}) \le f(X_{i,G}) \\ X_{i,G}, \text{if } f(U_{i,G}) > f(X_{i,G}) \end{cases} \quad (12)$$

where f(.) is the function to be minimized.

DEMO applies the following principle: The candidate $U_{i,G}$ replaces the parent $X_{i,G}$ in the population P_G if it dominates it. If the parent dominates the candidate, the candidate is discarded. Otherwise (when the candidate and parent are non-dominated with regard to each other), the candidate is added to the population P_G. This step is repeated until *Np* number of candidates is created. After that, we get a population of the size between *Np* and *2Np*. If the population has enlarged, we have to truncate it to prepare it for the next step of the algorithm. The truncation consists of sorting the individuals with non-dominated sorting and then evaluating the individuals of the same front with the crowding distance metric. The truncation procedure keeps in the population only the best *Np* individuals (with regard to these two metrics). The described truncation is derived from NSGA-II.

Pseudo code:

Input: Initial (x_c, y_c) and final CG (x_{cg}, y_{cg}) of the box, points of application of the two forces on the box by the two robots (x_e, y_e) and (x_f, y_f) and a threshold value $\in$.
Output: Forces applied by the two robots to move the box and the rotational angle α.
Begin
Set: $x_{curr} \leftarrow x_c; y_{curr} \leftarrow y_c;$
Repeat Until $||curr - G|| < \in$ // curr = (x_{curr}, y_{curr}), G = (x_{cg}, y_{cg}) //
Call DEMO$(x_{curr}, y_{curr}, x_e, y_e, x_f, y_f; \alpha, x_I, y_I, F_{1r}, F_{2r})$; Move-to$(x_{curr}, y_{curr})$;
End.

Procedure DEMO $(x_{curr}, y_{curr}, x_e, y_e, x_f, y_f; \alpha, x_I, y_I, F_{1r}, F_{2r})$
Begin
Initialize all solutions and place them in P_t; Set $R_t = P_t$;
Evaluate the fitness $(fit(X_i))$ of the population.
For Iter=1 to Maxiter do
For each solution X_i do
$V_i = X_j + rand(0,1) * (X_k - X_l), \{j,k,l\} \in [1, NP], i \neq j \neq k \neq l, rand(0,1) \in (0,1)$;
For each parameter d do
If $rand(0,1) < CR$(crossover ratio) **Then** $U_{i,d} = V_{i,d}$;
Else $U_{i,d} = X_{i,d}$;
End If;
End For;
Evaluate $fit(U_i)$;
If U_i dominates X_i **Then** $X_i \leftarrow U_i$;

```
    Else If Xi and Ui are non-dominated Then Rt = Rt ∪ Ui;
    End If;
  End For;
  Call NS (Pt, Rt);
  Memorize the best solution obtained so far;
End For;
Update:
  1. (xcnew, ycnew), (xenew, yenew), (xfnew, yfnew) using equation 1;
  2. xcurr ← xcurr + dcosθ; ycurr ← ycurr + dsinθ;
  3. xe ← xe + dcosθ; ye ← ye + dsinθ;
  4. xf ← xf + dcosθ; yf ← yf + dsinθ;
Return.

Procedure NS (Pt, Rt)
Begin
Construct non-dominated front sets (F1, F2--) from Rt;
Set: Pt+1 = ϕ, i=0;
Repeat until |Pt+1| + |Fi| ≤ NP
   Pt+1 = Pt+1 ∪ Fi; i=i+1;
Sort Fi in descending order of crowding distance;
Pt+1 = Pt+1 ∪ First (Np − |Pt+1|) elements of Fi;
End.
```

4 Experiment and Computer Simulation

The experiment was carried out on a simulated environment on Intel Core 2 Duo processor architecture with clock speed of 2GHz. The structure of solution in DEMO used here is given below.

$F1r$	X_I	α	$F1t$	d	$\mu.m$

Since all solutions in the Pareto front are equally good, to select the one among many possible solutions, we normalize time, energy and frictional loss for the individual solution in the Pareto front. Let *$f_1(i)$, $f_2(i)$ and $f_3(i)$* be the measure of time, energy and frictional energy loss for the i-th solution in the Pareto front.

$$f_1^f(i) = \frac{f_1(i)}{\sum_{i=1}^{n} f_1(i)}; f_2^f(i) = \frac{f_2(i)}{\sum_{i=1}^{n} f_2(i)}; f_3^f(i) = \frac{f_3(i)}{\sum_{i=1}^{n} f_3(i)} \quad (13)$$

The above process is repeated for all solutions in the Pareto front and in each step of the movement the following product is taken.

$$P_i = (f_1^f(i) \times f_2^f(i) \times f_3^f(i)) \quad (14)$$

Now, the effective solution having smallest P_i is identified in each step to move the box by the robots. The significance of this work is that here we have considered frictional loss occurring during the push operation by the robots with non uniform distribution of frictional coefficient in the robots work space. In this work, the position of the obstacles is made to be user defined.

The entire trajectory of motion for the box-pushing problem was obtained for different settings of K_e and K_t in [10, 120]. The experimental results reveal that the optimal performance for the given environment is obtained, when $K_e = 100$ and $K_t = 90$.

The frictional coefficients can also be changed during the course of the program. The trajectory of the box translated by robots on a frictionless surface using DEMO is represented in Fig. 5.

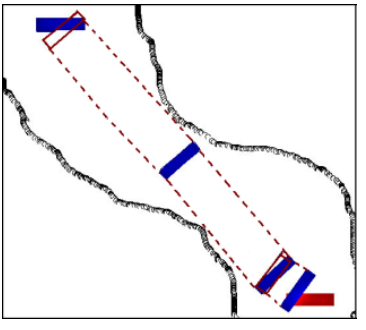


Fig. 5. Final configuration of the world map without friction compensation

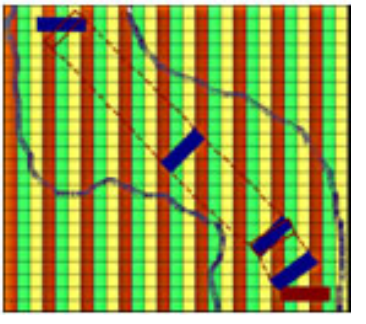


Fig. 6. Final configuration of the world map with static friction coefficients

Now, we have considered that different regions of the surface have got different static frictional coefficients. Under such circumstances, the trajectory of the motion the box has been depicted in Fig. 6. In Fig. 7 we have shown the trajectory of the box when frictional coefficients of different region in the workspace have been changed by user during the runtime of the program. All the required information to govern the motion of the box at each step is given in following Table 1-4.

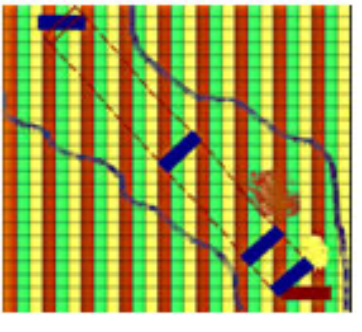


Fig. 7. Final configuration of the world map with dynamic friction coefficients

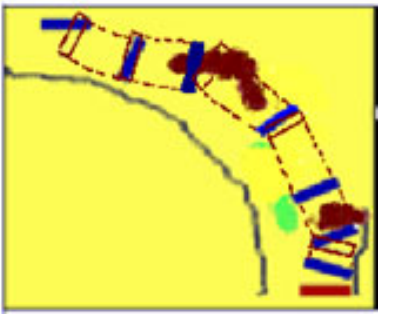


Fig. 8. Final configuration of the world map with friction compensation using DEMO requiring 5 steps

Table 1. Position of Solutions

Steps	F_{1r}	X_I	α	F_{1t}	d	μ. m
1	0.10	98.59	-0.753	0.10	400.00	2.316
2	11.93	253.35	0.024	14.76	290.29	0.713
3	5.55	412.23	-0.00043	0.10	99.53	2.736

Table 2. Position of Centre of Gravity and Point of Rotation

Steps	x_c	y_c	x_i	y_i
1	274.50	230.60	98.59	30.00
2	403.78	375.75	253.35	250.42
3	447.93	425.29	412.22	368.22

Table 3. Time and Energy required and Frictional Force overcome by Robots in each Step

Steps	Time	Energy	Frictional Force
1	1167.11	57.0077	23.167296
2	81.488	5721.355	0.699257
3	576.039	13.373	26.840160

Table 4. Total Time and Energy required and Frictional Force overcome by Robots

Total Time	Total Energy	Total Frictional Energy
1824.646	5791.7372	12141.501

In Fig. 8, it has been shown how friction compensation has been achieved by DEMO-based simulation. Initially the frictional coefficient of the workspace is considered to be high ($\mu = 40$) as shown by yellow. But during the course of the program whenever less frictional path has been provided ($\mu = 10$ in case of green and $\mu = 0.6$ in case of brown), robots always follow the less frictional path.

The relative performance of DEMO- and NSGA-II-based simulation can be studied through the observation of the local trajectory planning by the robots and total time and energy required as well as total frictional energy overcome by robots to move the box from initial position to goal position and also by observing the number of steps required to reach their goals with DEMO and NSGA-II based algorithm as shown in following Figs. DEMO seems to have marginally outperformed NSGA-II considering all the cases.

It is apparent from Table 5 that both energy consumption and frictional loss have been reduced by a margin of 26.46%, 60.81%, 79.45% and 0%, 21.28%, 34.68% respectively due to DEMO. But time required has been increased by 33.03%, 41.93% and 31.94% due to complexity involved in DEMO.

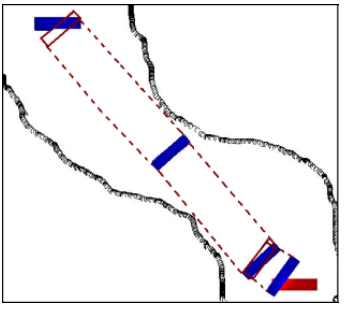


Fig. 9. Final configuration of the world map using DEMO without friction requiring 3 steps

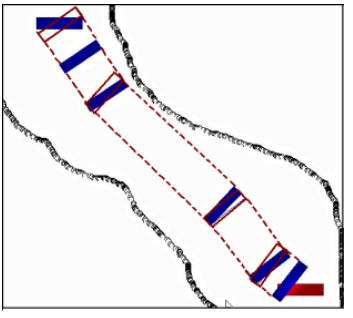


Fig. 10. Final configuration of the world map using NSGA-II without friction requiring 5 steps

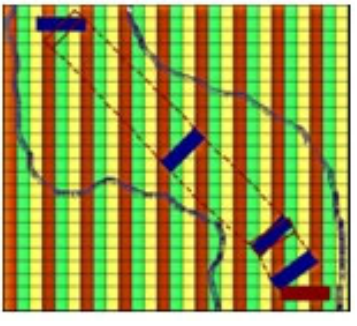


Fig. 11. Final configuration of the world map using DEMO with static friction requiring 3 steps

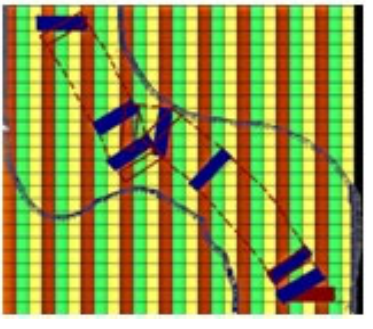


Fig. 12. Final configuration of the world map using NSGA-II with static friction requiring 6 steps

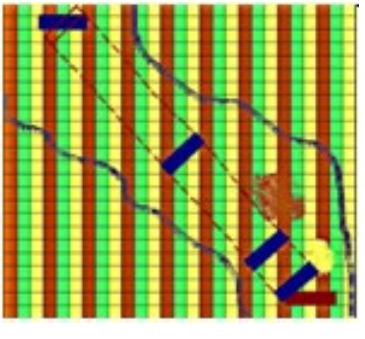


Fig. 13. Final configuration of the world map using DEMO with dynamic friction requiring 3 steps

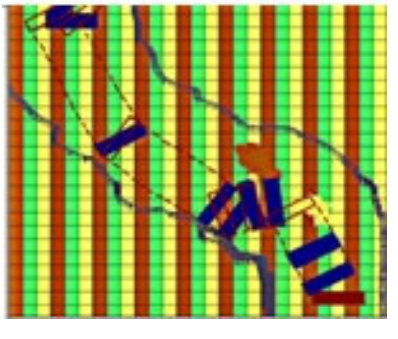


Fig. 14. Final configuration of the world map using NSGA-II with dynamic friction requiring 8 steps

Table 5. Total Time and Energy required and Frictional Force overcome by Robots

Case		Total Time	Total Energy	Total Frictional Energy
No friction	DEMO	248.318189	14617.80759	0
	NSGA-II	186.668398	19878.11030	0
Static friction	DEMO	522.226597	7091.913497	11876.664166
	NSGA-II	367.938531	18094.81926	15086.676629
Dynamic friction	DEMO	1824.646972	5791.737231	12141.501548
	NSGA-II	1382.992704	28193.83169	18587.646525

5 Conclusion

This paper provides a novel approach to handling box-shifting as a multi-objective optimization algorithm, and offers Pareto-optimal solutions in real time by utilizing the power of optimization of the DEMO program. The approach to the problem is unique and is different from the classical behavior based [5] and perceptual cues based [6] multi-robot box-shifting problems.

References

[1] Robič, T., Filipič, B.: DEMO: Differential Evolution for Multiobjective Optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 520–533. Springer, Heidelberg (2005)

[2] Chakraborty, J., Konar, A., Nagar, A., Das, S.: Rotation and translation selective Pareto optimal solution to the box-pushing problem by the mobile robots using NSGA-II. In: IEEE Congress on Evolutionary Computation (2009)

[3] Storn, R., Price, K.: Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11, 341–359 (1997)

[4] Miyata, N., Ota, J., Arai, T., Asama, H.: Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. IEEE Trans. Robot. Automat. 18(5), 769–780 (2002)

[5] Sugie, H., Inagaki, Y., Ono, S., Aisu, H., Unemi, T.: Placing objects with multiple mobile robots- mutual help with intension inference. In: IEEE International Conference on Robotics and Automation, pp. 2181–2186 (1995)

[6] Kube, C.R., Zhang, H.: The use of perceptual cues in multi-robot box pushing. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 2085–2090 (1996)

[7] Yamada, S., Saito, J.: Adaptive action selection without explicit communication for multi-robot box-pushing. In: IEEE International Conference on Intelligent Robots and Systems, pp. 1444–1449 (1999)

[8] Sugie, H., Inagaki, Y., Ono, S., Aisu, H., Unemi, T.: Placing objects with multiple mobile robots- mutual help with intension inference. In: IEEE International Conference on Robotics and Automation, pp. 2181–2186 (1995)

[9] Yamauchi, Y., Ishikawa, S., Uemura, N., Kato, K.: On cooperative conveyance by two mobile robots. In: IEEE International Conference on Robotics and Systems, pp. 1478–1481 (1993)

[10] Abbass, H.A., Sarker, R.: The Pareto differential evolution algorithm. Int. J. Artif. Intell. Tools 11, 531–552 (2002)
[11] Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), Canberra, Australia, vol. 2, pp. 862–869 (2003)
[12] Iorio, A.W., Li, X.: Solving Rotated Multi-Objective Optimization Problems Using Differential Evolution. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS (LNAI), vol. 3339, pp. 861–872. Springer, Heidelberg (2004)
[13] Gerkey, B.P., Matari, M.J.: Pusher-watcher: An approach to fault-tolerant tightly coupled robot coordination. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2002, vol. 1, pp. 464–469 (2002)
[14] Kovac, K., Zivkovic, I., Dalbelo, B.B.: Simulation of multi-robot reinforcement learning for box-pushing problem. In: Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference, MELECON 2004, vol. 2, pp. 603–606 (2004)
[15] Li, Y., Chen, X.: Modeling and simulation of a swarm of robots for box-pushing task. In: 12th Mediterranean Conference on Control and Automation, Kusadasi, Aydin, Turkey (2004)
[16] Wang, Y., de Silva, W.C.: Multi-robot box-pushing single-agent q-learning vs team q-learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3694–3699 (2006)
[17] E. J. Gene, J. Tong-Ying, H. Jun-Da, S. Chien-Min, C. Chih-Yung, C.: A fast path planning algorithm for piano mover's problem on raster. Proceedings of the IEEE, 522–527 (2005)
[18] Jed, L., Mark, R., Bruce, R.D., Donald, P.G.: Real-time robot motion planning using rasterizing computer graphics hardware. In: SIGGRAPH 1990: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, pp. 327–335 (1990)

MPG_AbTR: Ant Based Trusted Routing in MANets Using Mobile Process Groups

Aakanksha, Ravish Sharma, and Punam Bedi

Department of Computer Science, University of Delhi, Delhi, India
aakanksha_v@yahoo.co.in, sharma_ravish123@rediffmail.com, punambedi@ieee.org

Abstract. Routing in MANets is a challenging task due to its dynamic topology. The best and reliable path for routing is one which is more trustworthy and less loaded. This paper proposes an Ant-based Trusted Routing (MPG_AbTR) in MANets using the framework of Mobile Process Groups. The trust on a node is calculated using mobility rate, group membership time and number of overlapping groups the node is part of. As the number of overlapping groups of a node increases, confidence and trust on that node also increases. Consequently, the load on that node also increases as it becomes most trustworthy and reliable for data forwarding. This leads to its selfish and unreliable behavior in future, other nodes and routes being underutilized and unexplored. Therefore, to exploit the most trustworthy node for forwarding and side-by-side balancing the load on the node, ant-based pheromone updating strategy is used in our proposed approach.

Keywords: Ant Colony Optimization, Mobile Process Groups, Trust, MANets.

1 Introduction

A collection of mobile nodes communicating via wireless medium without any fixed infrastructure forms a Mobile Ad-hoc Network (MANet). The frequent topology change and lack of centralized control make these networks vulnerable to security attacks and pose various challenges such as routing, security, bandwidth and power consumption. These challenges can be tackled if nodes co-operate and collaborate with each other in a self-organized manner.

Self-organization can be defined as a process in which the internal level of organization of a system increases automatically without being guided or managed by an outside source. Self-organization techniques have been proposed in literature by taking analogies from nature where social insects self-organize their activities and achieve goals that overcome by far their capabilities as single individuals [13]. One such example of self-organization is foraging behavior of ants. Ants are able to discover the shortest path to a food source and share that information with another ant through stigmergy [8]. Stigmergy is a form of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 367–376.
springerlink.com

indirect communication used by ants which is achieved by laying a chemical substance called pheromone. The pheromone laid by one ant modifies the environment that is sensed by other ants. Stigmergic communication is used by ants to coordinate their problem-solving activities.

Ant Colony Optimization (ACO), a Swarm Intelligence technique, has been applied for routing and load balancing in literature in MANETs [18]. Lack of centralized control and fixed infrastructure in these networks makes it difficult to ensure that a packet has been delivered to the desired destination. This unreliability is due to the presence of malicious and selfish nodes acting as intermediate nodes in the network. Mobile Process Group (MPG) [1, 10, 11, 12] is a collection of processes or mobile nodes used for reliable coordination and communication between mobile processes. MPG_TAR, a trusted routing solution based on the formation of trusted MPG has been proposed in [21] to improve the reliability of communication even in the presence of malicious nodes. As observed in MPG_TAR, the load on a node x increases and it becomes trustworthy and reliable for data forwarding. As the time progresses, increase in load leads to selfish and unreliable behavior of the node x. In addition, other nodes remain underutilized leading to unused and unexplored paths.

This paper proposes MPG_AbTR, a routing and load balancing scheme based on ACO that uses the framework of trusted MPG. The amount of artificial pheromone (for stigmergic communication) is directly proportional to the trust on the next forwarding node. Ant-based pheromone updating strategy exploits the most trustworthy node for forwarding and balances the load on that node.

The rest of the paper is organized as follows: Section II gives related work. Section III presents the proposed MPG_AbTR algorithm. A simulation of the proposed scheme and the results are presented in section IV. Finally, section V concludes the paper.

2 Related Work

ACO technique was proposed by G.D.Caro and Dorigo [8] which defines a framework for solving graph-based combinatorial optimization problems such as Traveling Salesman Problem (TSP), vehicle routing and scheduling [7,9]. Besides these applications, ant algorithms have also been proposed in literature for dynamic and decentralized problems including solving traffic jam problems [2], congestion evasion [15], routing and load balancing. [18]. Di Caro and Dorigo also proposed AntNet [4] and AntHocNet [5,6]. AntNet was designed to be used for packet-switched networks while AntHocNet was designed specifically for purely dynamic and distributed mobile ad-hoc networks. Elke Michlmayr in [14] proposed SemAnt which routes queries efficiently in a network given frequent enough queries and a pre-defined taxonomy. A pheromone-based agent coordination mechanism has been proposed for peer-to-peer networks by Schelfthout et. al in [17]. Schelfthout et. al used object spaces as coordination mechanism to co-ordinate the agents in an open and self-organized system. A framework for reliable cooperation between mobile processes known as Mobile Groups has been proposed in [1,10,11,12]. Using this framework a group of agents

can ensure message delivery guarantees and co-ordinate with each other using some sort of virtual synchrony while moving. A self-organizing, self-healing on-demand loop-free path routing protocol using MPGhave been proposed in [20]. In [16] authors have proposed a cluster-based trust aware routing protocol for MANets. Trust being time-based information, can be analyzed by pheromone updating strategy known from ant algorithms [3].

The proposed MPG_AbTR algorithm uses ant-based pheromone updating strategy for best forwarding node selection and uses the framework of Mobile Process Groups (MPG)to co-ordinate the mobile nodes.

3 MPG_AbTR: Ant Based Trusted Routing Using Mobile Process Groups

In this section, we describe our proposed Ant-based Trusted Routing (MPG_AbTR) approach that uses the framework of mobile process groups Mobile process groups are the groups of mobile nodes within a local transmission range forming group views using consensus.

Consider for example a network of six groups at time t (Fig. 1). In the example, node ids {n1, n2,……, n23} represent IP addresses. Some nodes may lie in two or more radio ranges thus forming overlapping groups. The nodes which are part of two or more overlapping groups act as routers or forwarding nodes. Thus each node maintains three tables: 1) view table, 2) Forwarding table and 3) Overlapping groups table. The algorithm uses FANT (Forward Ant) and BANT (Backward Ant) to find a path from a source to the destination. Each source sends out (multicasts) FANTs to different neighbors to explore the best path with high

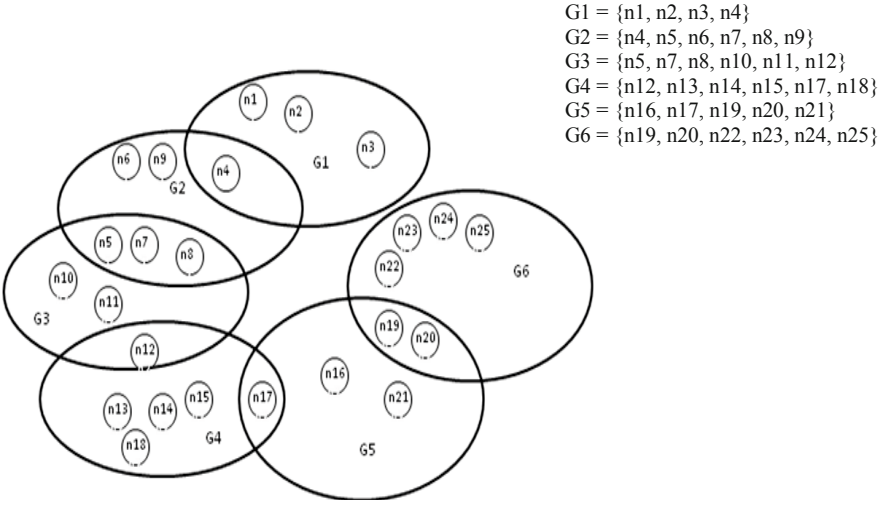


Fig. 1. An example ad-hoc network at time t with six groups

goodness value. The goodness value of the next neighbor (next hop) is calculated using some probability. The probability or goodness value is the likelihood of a data packet reaching its destination via the next hop. Backward Ants update the network routing tables as per the information collected by forward ants. The detailed operation of the algorithm is explained in the following subsections.

3.1 Maintaining the Group Views(View Table)

View table stores the information about all the members which are part of the current group view. The table entries store the local trust value (LTV) for each member of the group. Local trust value depends on the local interactions of node i with node j.

As illustrated in Fig. 1, n1 is a node in group G1. Table 1 shows the values of LTVs at node n1 at time t=0. Each entry in the table corresponds to the local trust value (LTV) for the neighbor nodes of n1. Trust of a node on itself is not defined. Initially when the network is started, LTV is initialized to 0.01 for all the nodes in a group. As local interactions take place, the values are updated periodically as given by eq. (1).

$$LTV_i = LTV_i + N_f^i \Big/ N_r^i \tag{1}$$

LTV is updated periodically by the ratio of total number of packets forwarded and received by node i ($N_f^i \Big/ N_r^i$). The updated view table at node n1 at time t=40 is shown in Table 2.

Table 1. View table at node n1 at time t=0

View No.	Node ID	Opinion(LTV)	Timer
1	n2	0.01	0
1	n3	0.01	0
1	n4	0.01	0

Table 2. View table at node n1 at time t=40

View No.	Node ID	Opinion(LTV)	Timer
1	n2	.20	40
1	n3	.18	40
1	n4	.19	40

3.2 Maintaining the Forwarding Table

Forwarding table stores the group reputations (aggregated trust of all nodes of a group) for each of the forwarding nodes of a group. Group reputations are

computed using equation (2). The forwarding table at node n1at time t=0 is shown in Table 3.

$$GrTV_f(g) = \sum_{\forall i \in g} (LTV_i) \tag{2}$$

Table 3. Forwarding table at node n1at t=0

Node ID	Group ID	Group Reputation(GrTV)
n4	G1	0.03

The group reputations are re-computed as and when required by a source node for forwarding a data packet and forwarding table is updated accordingly. The updated forwarding table at node n1 at time t = 40 is shown in Table 4.

Table 4. Forwarding table at node N1at t=40

Node ID	Group ID	Group Reputation(GrTV)
n4	G1	0.61

Only if the group reputation of a forwarding node is good (i.e. above some specified threshold), the trustworthiness of that forwarding node is computed using equation (3). We have taken the threshold to be 0.6 for group reputations.

$$\beta = \frac{N_f}{N_r} \tag{3}$$

This trustworthiness factor is the ratio of the total number of packets forwarded by total number of packets received by node in a group. This ratio may be lower if some of the packets have not been forwarded by a node and is an indicator of selfish or malicious behaviour.

3.3 Maintaining the Overlapping Groups Table

Overlapping groups table stores the confidence value of each forwarding node. This value is calculated by dividing the remaining battery power with the number of overlapping groups of a forwarding node. More the battery power and less the number of groups, more is the confidence and trust. An example overlapping groups table at node n7 at time t=0 is shown in Table 5 below. We assume an equal battery power for all the nodes and assign a constant value equal to 500mW initially.

Table 5. Overlapping groups table at node n7at t=0

Group ID	Forwarding node ID	Confidence
G1	n4	500/2
G3	n5	500/2
G3	n8	500/2
G4	n12	500/2

In the example above, all the nodes are part of two overlapping groups; therefore, confidence on all of them is initially same. As time progresses, the battery power depletes in accordance to the usage and node's participation in the network activity. Also the group memberships may change due to node movements. Hence the number of overlapping groups (which a node is part of) may also change thereby changing the confidence on that node. This value is updated in overlapping groups table. Table 6 shows the updated confidence values in overlapping groups table at node n7 for forwarding nodes n4, n5, n8 and n12. Node n5 has higher confidence value as there are three forwarding nodes in that group (n5, n7, n8) and n5 is less trusted and hence underutilized. Therefore the remaining battery power is more, leading to higher confidence value.

Table 6. Overlapping groups table at node n7 at t=40

Group ID	Forwarding node ID	Confidence
G1	n4	180
G3	n5	225
G3	n8	160
G4	n12	173

The confidence value is normalized between 0 and 1 to match the range of trust.

3.4 Trust Computation

The trust is computed by considering the route reliability, trustworthiness and confidence parameters of the next hop as discussed in subsections above. Local trust values, group reputations and confidence values for the next hop are computed and updated in the view table, forwarding table and overlapping groups table respectively. These values are then used to compute and update trust as given in (eq. 4).

$$T_f(g) = \alpha \cdot \delta + \beta \cdot \tau + \gamma / \eta \tag{4}$$

where α represents LTV_i and δ is the mobility rate which is inversely proportional to the speed of the i^{th} node. β is the route reliability factor given by eq. (3) and τ is the group membership time of node i in its group. γ/η is the confidence value which is the ratio of remaining battery power and the number of overlapping groups of the forwarding node.

3.5 Forwarding Node Selection

The probability with which an ant at node i chooses the next forwarding node in the group view g is given as follows:

$$p = \frac{T_f(g)^{\varepsilon}}{\sum_{f \in view_f(g)} T_f(g)^{\varepsilon}} \tag{5}$$

Where $view_f(g)$ is the set of all forwarding nodes in the current group view and ε is a parameter to control the path exploration using ants and is set to 1 (currently).

3.6 Pheromone Trail Updation

Each ant computes and updates the trust on a forwarding node $f \in view_f(g)$ using eq. (6).

$$T_f(g) = (1-\rho)T_f(g) + \rho \Delta T_f(g) \tag{6}$$

where

$$\Delta T_f(g) = \begin{cases} T_f(g) + GrTV_f(g) & \text{if f is selected forwarding node} \\ 0 & \text{otherwise} \end{cases}$$

$T_f(g)$ is trust on forwarding node f in group g and $GrTV_f(g)$ is the group reputation of node f in group g. The local pheromone updation rule updates pheromone value on all the links connecting the forwarding nodes. ρ is the pheromone evaporation coefficient.

3.7 MPG_AbTR Description

MPG_AbTR uses a hybrid approach as it follows proactive strategy for initialization and maintenance phase and reactive strategy for route discovery phase. In the proactive strategy the three tables viz. view table, forwarding table and overlapping table are maintained and updated periodically. The reactive strategy is used to discover a route to the destination as and when requested by the source node.

Initialization and maintenance phase (Proactive strategy). In this phase, path maintenance and path improvement of the already existing paths is done. As the nodes move, they may leave or join the groups, changing the group memberships and hence the group views. Changes in the group views also have impact on the information available in forwarding table and overlapping groups table. Therefore, these tables along with group view table have to be updated

accordingly. Continuous topology changes are taken care of by the MPGs view change service, that keeps track of the group join and group leave operations.

Route discovery phase (Reactive strategy). Reactive strategy is used to establish a path between the source node s and the destination node d. A path is established reactively (i.e. on demand) when a source node wants to send some data. This path is set up by creating a forward ant $FANT_d^s$ at source node and then forwarding this ant to all the group members in the current group view. The forward ant traverses the best path to the destination by choosing the next hop with a probability P (eq. 5). Each forward ant keeps a list of nodes it has visited. A backward ant $BANT_s^d$ is created which travels from destination to source retracing the same set of nodes as visited by forward ant. While moving from one hop to the next hop, the backward ant updates pheromone as given by equation (6).

The pheromone updation and probability computation both depend upon trust of forwarding node. If the trust on the node is more pheromone deposition is more and the probability of choosing the trustworthy next hop increases but as time progresses trust and hence pheromone value decreases due to the confidence parameter in trust computation. The increased traffic through the same node depletes the battery power and increases the load on that node. This in turn decreases trust and hence pheromone deposition. Therefore, this ant-based scheme implicitly does load-balancing and always does routing by choosing the best, reliable and trustworthy path to the destination.

4 Experimental Study

The proposed scheme was implemented in JADE (Java Agent Development Environment).JADE was chosen for simulation as each node in the Mobile Process Groups is assumed to be hosting an agent. Also, parallel execution of all the agents can be done in JADE.

The example network, as given in Fig.1 was implemented. The topology of the network, entire information about neighbors, routing and forwarding information and number of overlapping groups were stored in the text files of the corresponding nodes (agents). Initially the local trust value for each of the neighbors was initialized to be 0.01. Periodically these values were updated in the corresponding tables as per the algorithm.

The algorithm was executed multiple times with different communication scenarios by varying the trust values, reputations and battery power. Trust graph of the example network at time t=40is shown in Fig. 2.

It can be seen observed from Fig. 2 that there are three forwarding nodes n5, n7, n8 for forwarding packets in group G3. The computed trust values for each of these nodes at node n4 are 0.48, 0.64 and 0.7 respectively. So, the next forwarding node to be selected by the ant would be node n8. Further, nodes n12 and n17 are the only forwarding nodes for group G4 and G5 with trust values 0.63 and 0.68 respectively. Out of the two nodes n19 and n20, only n20 would be selected for forwarding as trust of n20 is more and above threshold.

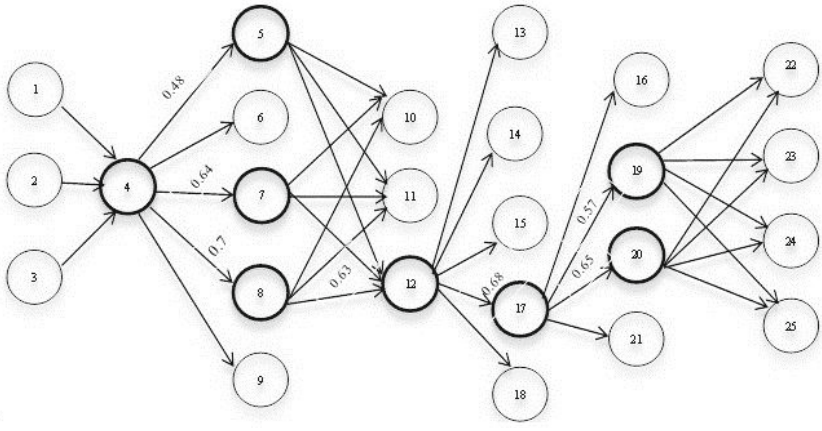


Fig. 2. Trust Graph for the example network at time t=40

The experiments prove the efficacy of the presented approach. It was observed that the most trusted node was the most loaded one and to balance the load, ant-based pheromone updation strategy explored other trusted paths and hence balanced the load on the most trusted forwarding node. Also, it minimized the effects of false negative trust.

5 Conclusion

In this paper, we have proposed an ant-based approach MPG_AbTR for performing routing on MANets. The routing is performed within the framework of Mobile Process groups (MPG). For finding the goodness of the routing path, the proposed approach uses trust on forwarding nodes. The trust is updated periodically and the pheromone is deposited and evaporated proportionately. The amount of artificial pheromone is directly proportional to the trust on the next forwarding node. Ant-based pheromone updating strategy exploits the most trustworthy node for forwarding and balances the load on that node.

References

[1] Assis Silva, F.M., Macedo, R.J.A.: Reliability requirements in Mobile Agent Systems. II WTF, SBC Curitiba Julho (2000)

[2] Bedi, P., Mediratta, N., Dhand, S., Sharma, R., Singhal, A.: Avoiding traffic jam using Ant Colony Optimization – A novel approach. In: Proc. of the Intl. Conf. on Computational Intelligence and Multimedia Applications, vol. 01, pp. 61–67 (2007)

[3] Bedi, P., Sharma, R.: Trust based recommender system using ant colony for trust computation. Expert Systems with Applications (2011) (article in press)

[4] Di Caro, G., Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research 9, 317–365 (1998)

[5] Di Caro, G.A., Ducatelle, F., Gambardella, L.M.: AntHocNet: An Ant-Based Hybrid Routing Algorithm for Mobile ad Hoc Networks. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 461–470. Springer, Heidelberg (2004)

[6] Di Caro, G., Ducatelle, F., Gambardella, L.M.: AntHocNet: an adaptive nature inspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications (Special Issue on Self-Organization in Mobile Networking), 16(2) (2005)

[7] Dorigo, M., Bonabeau, E., Theraulaz, G.: Swarm Intelligence: From Natural to Artifical Systems. Oxford University Press, New York (1999)

[8] Dorigo, M., Di Caro, G., Gambardella, L.: Ant colony optimization: a new meta-heuristic. In: Proceedings of the Congress on Evolutionary Computation, vol. 2, pp. 1470–1477. IEEE Press, Washington, DC (1999)

[9] Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004), ISBN 0-262-04219-3

[10] Macedo, R.J.A., Assis Silva, F.M.: Mobile groups. In: 19th Brazilian Symposium on Computer Networks, SBRC (2001)

[11] Macedo, R.J.A., Assis Silva, F.M.: Coordination of mobile processes with mobile groups. In: Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 177–186 (2002)

[12] Macedo, R.J.A., Assis Silva, F.M.: The mobile groups approach for the coordination of mobile agents. Journal of Parallel and Distributed Computing 65, 275–288 (2005)

[13] Mamei, M., Menezes, R., Tolksdorf, R., Zambonelli, F.: Case studies for self-organization in computer science. Journal of System Architecture 52(8), 443–460 (2006)

[14] Michlmayr, E.: Ant Algorithms for Self-Organization in Social Networks. Ph.D. Thesis, Vienna University of Technology (2007)

[15] Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D., Hortner, H.: Self-organising congestion evasion strategies using ant-based pheromones. Intelligent Transport Systems, IET 4(1), 93–102 (2010)

[16] Safa, H., Artail, H., Tabet, D.: A cluster-based trust-aware routing protocol for mobile ad hoc networks. Journal of Wireless Networks 16(4), 969–984 (2010)

[17] Schelfthout, K., Holvoet, T.: A Pheromone-Based Coordination Mechanism Applied in Peer-to-Peer. In: Moro, G., Sartori, C., Singh, M.P. (eds.) AP2PC 2003. LNCS (LNAI), vol. 2872, pp. 71–76. Springer, Heidelberg (2004)

[18] Sim, K.M., Sun, W.H.: Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 560–572 (2003)

[19] Vatsa, A., Bedi, P.: Load Balancing on Dynamic Network Using Mobile Process Groups. In: Intl. Conf. on Advanced Computing and Communications, pp. 553–558 (2007)

[20] Vatsa, A., Bedi, P.: A Self-organizing Self-healing On-demand Loop-free Path Routing Protocol Using Mobile Process Groups for Mobile Ad-hoc Networks. In: Advances in Recent Technologies in Communication and Computing (ARTCom), pp. 396–400 (2009)

[21] Vatsa, A., Bedi, P.: MPG-TAR: Mobile Process Groups Based Trust Aware Routing Protocol for MANETs. In: Intl. Conf. on Advances in Recent Technologies in Communication and Computing, pp. 131–135 (2010)

Uncertainty Analysis on Neural Network Based Hydrological Models Using Probabilistic Point Estimate Method

K.S. Kasiviswanathan and K.P. Sudheer

Department of Civil Engineering, Indian Institute of Technology,
Madras, Chennai – 600036, India
vishwaiitr@gmail.com, sudheer@iitm.ac.in

Abstract. Modeling hydrological processes are always a challenge due to incomplete understanding of the physics of the process. Therefore, various levels of simplification are essential during modeling, which are otherwise very complex. In addition, most of the hydrological processes being natural are random processes. Apart from the standard physics based models developed in hydrology, the artificial neural network (ANN) approach has been getting lot of attention plausibly due to the complexity associated with the system. However, in most of the application of ANN in hydrology the model is considered as deterministic despite a large amount uncertainty associated with the final ANN models. Very recently, there has been considerable interest to quantify the uncertainty associated with ANN models, and not much work is reported since application of standard methods for uncertainty quantification was difficult due to the parallel computing architecture of the ANN. This paper presents the application of probabilistic point estimate in quantifying the uncertainty of ANN river flow forecasting model. The method is demonstrated through a case study of L'Anguille watershed located in United States. The results show that the method effectively quantifies uncertainty in the model output by estimating the parameters in orthogonal domain. The study also suggests that the method can be employed for models with lesser number of simulations, and do not require much knowledge about the parametric distribution of the model.

Keywords: Orthogonal domain, parallel computing, physics based models, random processes.

1 Introduction

Artificial neural networks (ANN) belong to data driven models applied in various domains such as prediction and classification. Application of ANN in hydrological

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 377–384.
springerlink.com

models suggested promising solutions in the last two decades. A detailed review of ANN in water resources is available at [1, 2 and 7].

Despite a plethora of applications in hydrology, most of the studies reported in hydrology pertaining to ANN considers only point prediction and lacks in adequate representation of reality due its deterministic nature. This limits the reliable use of ANN in solving real life problems in hydrological application. The uncertainty associated with ANN based models occur at different level (i.e. input selection, parameter and structure of the model). Therefore, quantification of uncertainty is foremost important by constructing the prediction interval of the model output with respect to inputs and parameters.

In general, the uncertainty of a stochastic variable or parameter is represented by the probability distribution function (PDF). Quantifying the uncertainty is never a trivial task due to the detailed information requires about each variables and parameters. In this regard, variety of method has been introduced such as Monte Carlo simulation (MCS), Latin hypercube sampling (LHS), Direct integration method, and Taylor expansion approximation are only a few [11]. Most of the aforementioned methods explicitly require detailed information on the PDFs of each stochastic parameter. This is always computationally intensive and challenging task, as the complete knowledge of the PDFs is not available.

To circumvent all these issues, probabilistic point estimate (PPE) methods have been introduced where the lack of complete knowledge of the PDF of stochastic parameter exists. In PPE method, the parameters of the models are approximated using few statistical moments such as mean, standard deviation, skewness and kurtosis. The advantage of using PPE method does not require detailed information of the parameters and it can be performed with less computational burden. There are different PPE methods proposed by various researchers [4, 5, 6 and 9], in which their assumptions depend on whether the stochastic parameters are correlated or uncorrelated, symmetric or asymmetric.

This paper presents a method that combines the bootstrap technique with the Modified Harr's (PPE) method to evaluate the parametric and predictive uncertainty associated with the ANN models applied to river flow forecasting models. The earlier studies that used bootstrap techniques did not consider developing the uncertainty bounds of the predictions; rather most of them reported the simulated variability in predictions. Therefore the PPE based uncertainty analysis method is an alternate method without complex derivative of non linear functions involved in ANN.

2 Study Area and Data Used

The L'Anguille watershed is located in northeastern Arkansas. An index map of the watershed is given in Fig. 1. Portions of the watershed are in Craighead, Poinsett, Cross, St.Francis, Woodruff and Lee counties. The watershed area is about 2458.8 km². Mean elevation of this watershed is 71m. The average annual precipitation of this watershed is 45.8 inches and 8 inches of snowfall. The daily rainfall

runoff data was collected during 1998-2005 and used for the modeling. The average daily flow ranged from a minimum of 0m³/sec to a maximum of 634.297m³/sec with a standard deviation of 43.9457m³/sec.

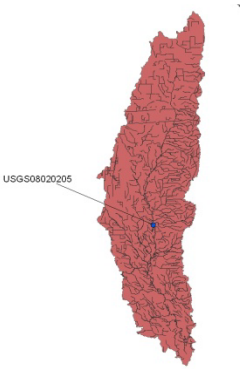


Fig. 1. L'Anguille River watershed, location of USGS gauging stations

3 Methodology

The proposed methodology is to couple the PPE method with ANN to quantify the predictive uncertainty and it can be seen in flowchart (Fig. 2). One of the most important steps in neural network based models requires appropriate input selection. Inputs are identified based on the catchments characteristics and it requires some priori knowledge about the system [3]. However in most of the cases, it is difficult to get the information of input variables to be included in the modeling exercise. In this study, the cross correlation and auto correlation analysis is used to identify the model inputs for L'Anguille watershed and the identified inputs are (R_t, R_{t-1}, R_{t-2}, R_{t-3}, Q_{t-1}, Q_{t-2}, Q_{t-3}), in which R represents rainfall and Q represents flow values at current and lag time.

The genetic algorithm based weight optimization neural network is applied to bring the non linear relationship between input and output variables. The optimum number of hidden node selection is required for the good generalization of developed model. This is carried out using trial and error method by training the ANN model with different number of hidden nodes. In accordance with principle of model parsimony, it is suggested that optimality is defined as the smallest network that adequately captures the relationships in the calibration data. In this study the number of hidden node was fixed as 2 initially and increased upto 10 with the step size of one and found that two hidden node is sufficient to bring the required relationship with single hidden layer.

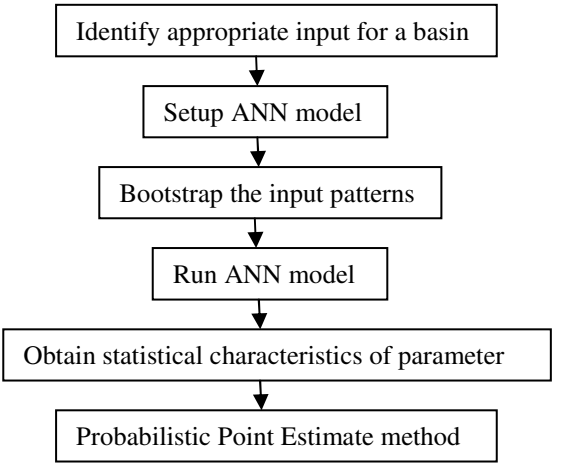


Fig. 2. Flowchart of proposed methodology

Bootstrap technique is a sampling technique which samples the pattern with replacement from the given data set to train ANN model. The technique quantifies the parametric uncertainty due to input variability by means of sampling. The approximated statistical moments of model parameters are obtained through bootstrap techniques and these are the primary inputs used in PPE method in order to quantify the predictive uncertainty.

The current study uses the Modified Harr's method proposed by [4]. The basic principle behind this method is that the area under the entire probability distribution of a chosen random variable or parameters can be redistributed at a finite number of discrete points, provided that the first few moments of the distribution are maintained [11].

The advantage of using this method does not require partial derivative of weights and biases. The basic assumption of the method is that all parameters of the ANN model follows normal distribution and correlated with each other. It is always a difficult issue and computationally intensive to perform uncertainty analysis in the correlated domain. However, methods are available through which the vector of multivariate normally distributed ANN parameters (i.e. weights and biases) can be expressed in terms of uncorrelated standard normal random vector. The method uses orthogonal transformation where the parameters are uncorrelated. This is carried out by calculating eigen vector from correlation coefficient of parameter values. Finally, the parameter values are converted into the original domain for the analysis as given in equation (1) through which the standard error can be estimated. These parameters are used to quantify the output uncertainty of ANN.

$$x_k \pm = \mu_K \pm \sqrt{K\lambda_k}\,\sigma_K v_k \tag{1}$$

In this method μ_K, σ_K represents mean and standard deviation of parameters obtained through bootstrap. A hypersphere with radius $\sqrt{K}$ centered at the origin in the K- dimensional standardized normal space is constructed and each estimate has been weighed based on the eigen values λ_K. The points at which the model output is to be computed are located at the intersections of the hypersphere and the K eigenvectors v_k of the correlation matrix [12]. Due to the normal distribution and the same scale on each component in the uncorrelated, standardized normal space, the 2K selected points in the original parameter space are located on a hyper surface with equal PDF values.

The standard error of the model is computed by calculating the first $E(Q)$ and second moment $E(Q^2)$ of model outputs Q_t with respect to all the parameters of the model involved and it can be seen in equations (2).

$$StdError = \sqrt{E(Q_t^2) - E(Q_t)^2} \tag{2}$$

4 Results and Discussion

4.1 Determination of Statistical Moments of Parameters Using Bootstrap Method

Bootstrap method employed in this study is to get the prior knowledge of the probability distribution of parameter. Out of 2880 total patterns, the initial 1880 patterns were selected for training and the remaining was used for validation. Further out of 1880 training sample patterns, 1000 samples were bootstrapped randomly with replacement. Note that the 1000 sets considered for validation were corresponding to a continuous hydrograph. Based on this approach, 100 models were trained to get variability among the set of parameters. The performance of the models (100 independent models) has been evaluated using Nash-Sutcliffe efficiency [8]. Their values during calibration and validation are reported in Table 1. and it is clear that the model has the average value of NSE as 0.8872 in calibration and in validation it has a value of 0.7902 shows the consistent performance of ANN model with different sampled patterns through bootstrap. The model has an average deviation of about $\pm$ 3.5% in calibration and $\pm$ 9.2% in validation. This suggests that more than 95% of simulated values are close to the observed values in calibration similarly more than 90% of values in validation. The result of NSE statistics shows that the performance is comparatively good in calibration as well as in validation stages.

Table 1. Nash Sutcliff Efficiency statistics for 100 models

Statistical Indices	Minimum	Maximum	Mean	Std deviation
Calibration	0.7937	0.9450	0.8872	0.0361
Validation	0.1528	0.9022	0.7902	0.0922

It is interesting to note that larger variance in weights obtained from the flow inputs compared to rainfall inputs. On an average, the weights connecting the flow inputs are ranges between -8 to 8 where as the weights connecting rainfall inputs ranges between -5 to 5. Therefore, it is observed that the tighter distribution of weights connecting rainfall inputs may not be having much influence compared to the weights connecting flow inputs. This information suggests that flow inputs contribute more in prediction and the similar results were observed [10].

4.2 Probabilistic Point Estimate Method in Uncertainty Estimate of Neural Network

The first step of Modified Harr's PPE is to construct the correlation coefficient matrix of the parameters obtained from the bootstrap. It is observed that almost all the weights connecting inputs are highly correlated with weights connecting hidden to output node except few (WI_5H_2 – WI_6H_2, WI_6H_1 – WI_7H_1). The weight (WI_5H_2 – WI_6H_2) which has the correlation value of 0.50 connecting the inputs Q_{t-3} and Q_{t-2}, similarly the weight (WI_6H_1 – WI_7H_1) has the correlation values of 0.63 connecting the inputs Q_{t-2} and Q_{t-1} respectively. This can be explained as Q_{t-3}, Q_{t-2} and Q_{t-2}, Q_{t-1} are highly correlated therefore their connection weights are also positively correlated in order to bring the required relationship with good prediction. The standard error varies between 38 m^3/s to 13 m^3/s where the flow ranges between 340 m^3/s to 12 m^3/s in case of high flow, where as in medium and low flow condition, the standard error varies between 13 m^3/s to 5 m^3/s where the flow ranges between 138 m^3/s to 17 m^3/s. From this, it is clearly understood that model creates on an average 10% of error. Figure 3 shows 95% prediction interval

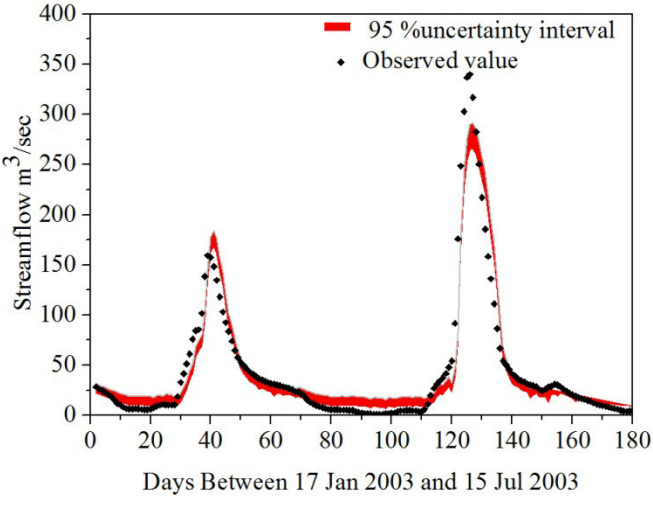


Fig. 3. The 95% predictive uncertainty interval of high flow simulation during validation period

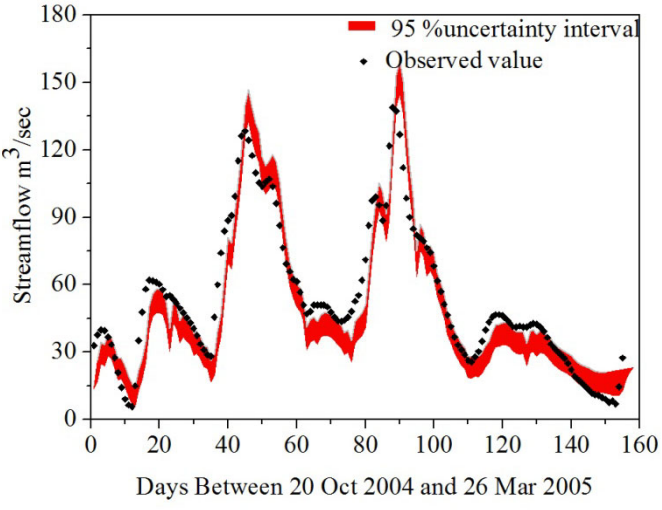


Fig. 4. The 95% predictive uncertainty interval of medium and low flow simulation during validation period

estimated using Modified Harr's PPE for high flow and Figure 4 shows 95% prediction interval estimated for medium and low flows during validation period.

From Figure 3, it can be noticed that 95% uncertainty interval covers most of the values except some of the peak flow observed values. In case of very low flows the quantification of uncertainty is high, this may be due to the model was consistently over estimating the low flow values. However in medium flow condition, the quantification is quit reasonable and the band covers most of the observed values.

There are several potential reasons for the inadequate performances of PPE based method to capture the uncertainty intervals at different flow components. One of the major reasons behind this is our level of understanding the hydrologic system still far from complete. This method considers the model parameters are normally distributed. However the actual distribution of parameter may not be normal distribution and the statistical information obtained from these distributions are only approximated values of prior distributions. Therefore, the efforts are to be made for the appropriate convergence to the true posterior distribution in order to get the accurate values of statistical moments. However, deriving the joint posterior distribution of parameter is beyond the scope of this paper.

5 Conclusion

This paper presents a probabilistic point estimate based method to conduct uncertainty analysis for ANN river flow forecasting models. The method uses statistical characteristics of parameters such as mean and standard deviation. These statistical moment values are obtained through bootstrap techniques. The selected

approach is demonstrated using case study of L'Anguille watershed located in United Nations. The analysis of results illustrate that the proposed method of uncertainty analysis of ANN models is very effective in case of detailed information about the parameters of the model is not available. The uncertainty band covers most of the medium flow values. The reason for such behavior may be during the entire simulation the model biased towards medium flow values. Monte Carlo simulation based method requires massive computation and hence, the PPE based method of quantifying the uncertainty of ANN model is relatively simple with less number of simulations.

References

[1] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology: Artificial neural networks in hydrology I: Preliminary concepts. J. Hydrol. Eng. 5, 115–123 (2000a)

[2] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology: Artificial neural networks in hydrology II: Hydrologic applications. J. Hydrol. Eng. 5, 124–137 (2000b)

[3] Campolo, M.P., Andreussi, Soldati, A.: River flood forecasting with a neural nework model. Water Resour. Res. 35(4), 1191–1197 (1999)

[4] Chang, C.H., Tung, Y.K., Yang, J.C.: Evaluation of probability point estimate methods. Appl. Math. Model. 19(2), 95–105 (1995)

[5] Harr, M.E.: Probabilistic estimates for multivariate analysis. Appl. Math. Modeling 13(5), 313–318 (1989)

[6] Li, K.S.: Point estimate method for calculating statistical moments. J. Eng. Mech. 118(7), 1506–1511 (1992)

[7] Maier, H.R., Jain, A., Dandy, G.C., Sudheer, K.P.: Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environmental Modeling and Software 25(8), 891–909 (2010)

[8] Nash, J.E., Sutcliffe, J.: VRiver flow forecasting through conceptual models: 1. A discussion of principles. J. Hydrol. 10, 282–290 (1970)

[9] Rosenblueth, E.: Point estimates for probability moments. Proc. US National Academy of Sciences 72, 3812–3814 (1975)

[10] Srivastav, R.K., Sudheer, K.P., Chaubey, I.: A simplified approach to quantifying predictive and parametric uncertainty in artificial neural network hydrologic models. Water Resour. Res. 43, W10407 (2007), doi:10.1029/2006WR005352

[11] Tsai, C.W., Franceschini, S.: Use of probabilistic point estimate methods in uncertainty analysis for environmental engineering applications. ASCE Journal of Environmental Engineering 131(3), 387–395 (2005)

[12] Tung, Y.K.: Uncertainty and Reliability Analysis. In: Water Resources Handbook. McGraw- Hill Book Company (1996)

Redesign of Wireless Sensor Actor Network due to the Insertion of Obstacles

Shaimaa Alrashed, Paulvanna Nayaki Marimuthu, and Sami J. Habib

Kuwait University
Computer Engineering Department
P.O. Box 5969 Safat 13060 Kuwait
sami.habib@ku.edu.kw

Abstract. Wireless sensor and actor network (WSAN) comprises of a set of tiny, stationary and low-powered sensor nodes deployed to collect environmental data, and WSAN also includes energy rich mobile actors to act as intermediate storage between the sensors and the service center. The actors aggregate the sensors' data, thereby saving the power of the sensors to transmit directly to the service center. Otherwise, the data transmission to a service center at a longer distance by the individual sensors consumes a lot of power. In this paper, we have extended our previous work on automatic placement and binding of actors to the sensors to examine the impact of the presence of obstacles within WSAN. We have proposed a redesign algorithm to the obstacle insertion problem within WSAN, which examines the position of the obstacles in view of actor-sensor line-of-sight (LOS) to identify the blocked sensors, and then the redesign algorithm relocates the actors to reestablish the communication with the blocked sensors. We have defined the redesign problem as an optimization problem, where the objective function is to minimize the energy consumption, while maximizing the actor-sensor binding ratio. We have utilized Simulated Annealing to search for the best possible position for the actors to be relocated within the network to cover all the sensors, which go out-of-sight after the insertion of obstacles. The simulation results demonstrate that our redesign algorithm maintains the actor-sensor binding ratio at 100% with the presence of obstacles.

Keywords: Obstacle, Redesign, Optimization, Simulated Annealing, Data analysis.

1 Introduction

In recent years, the advances in technological development have led to the widespread usage of wireless sensor networks. The low-power tiny sensors distributed within an area can collect and disseminate the data estimated from a physical environment; thereby the sensor nodes are periodically executing the three tasks; sense, process, and transmit. To increase the lifespan of the sensors, energy rich actors are deployed within the network to aggregate the data from the sensors and

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 385–396.
springerlink.com

transmit it to the service center. A WSAN is a distributed system of sensor and actor nodes, which are interconnected over wireless links. The nodes in a WSAN can be either stationary or mobile, depending on the type of the target application. In many situations, however, the sensor nodes are stationary whereas actor nodes are mobile and they are attached with mobile robots to change their location.

WSAN are deployed in difficult and hostile environment such as disaster rescue operations, home automation, smart spaces, pervasive computing systems, and cyber-physical systems and these applications are expected to be fault tolerant and to be lower in energy consumption. The presence of obstacles should be taken into serious consideration, whereby one or more sensors may abstain from the line-of-sight of the actors due to the inserted obstacles in the sensing field. The obstacles within the WSAN, blocking the communication between actors and sensors made the area under the surveillance goes unwatched. One of the approaches to reestablish connection between the actor and the sensors may be achieved through relocating the mobile actors.

In this paper, we have extended our previous work [1], where we have explored the automatic placement of actors within the wireless sensor-actor network to bind maximum number of sensors located in its communication radius in an obstacle free setting. In this paper, we have considered a WSAN network with obstacles in the sensing field and we have proposed a redesign algorithm to examine the sensors blocked by the obstacles and to relocate the actors to reestablish broken communication with the blocked sensors. We have employed Simulated Annealing to search for the best possible locations to relocate the actors, thereby covering all the sensors. The simulation results show a 100% sensor-actor binding ratio at all times.

2 Related Work

Most of the research work in WSAN focused on energy minimization techniques and localization techniques. Akyildiz and Kasimoglu [2] discussed the research challenges and requirements in communication protocol with the presence of actors in WSAN. Akkaya and Younis [2] formulated coverage and latency aware actor placement problem to find a set of head-clusters. The simulation considered the coverage of the actors and the data gathering latency, while determining the location of the actor nodes. The authors [3] also investigated the problem of actor placement for increased coverage while maintaining inter-actor connectivity. They presented C^2AP, an algorithm for Coverage-aware and Connectivity constrained actor positioning in WSANs. Evolutionary and heuristic algorithms were also employed to search for the optimal solution in localization problems [5][6].

Recent research works focused on various techniques to transmit the data in the presence of obstacles within the sensor network. Few researchers discussed about the routing algorithm to route the data in the presence of obstacles. There are also few works on description about the shape and material of the obstacles to examine the radio frequency wave reaching the sensors. Chatzigiannakis et al. [7] proposed a systematic obstacle model for precise simulation of real world phenomena, whereby the obstacles were classified as physical and communication obstacles.

The different type of obstacle shapes are described using geometric elements such as rectangular, circular, crescent etc.

An obstacle mobility model was proposed in paper [8] and it included four components. The first component modeled a terrain in which a user can define the positions, shapes and sizes of the objects and the second component, a movement graph, defined a set of pathways along which the mobile nodes move. The third component selected the shortest path (using Voronoi diagram) to move the nodes between two locations in the movement graph, and the fourth component was the signal propagation model, which computed the approximate signal attenuation experienced by the radio wave.

The papers above focused on routing techniques, which reestablish the broken link due to the presence of obstacles in the sensing field, actors' placement in an obstacles-free area and focused on modeling the obstacles. They also proposed a mobility model for obstacles-free sensing fields. In this paper, we have considered an obstacle insertion problem and we have proposed a redesign algorithm, whereby the mobile actors within WSAN are relocated to restore the broken communication links to bind maximum number of nearby sensors.

3 System Model

The wireless sensor and actor network model comprises of N sensors and M actors (M << N), randomly deployed within the given area A. The sensor field (A) is divided into H x W cells as illustrated in Figure 1, whereby the center-of-mass of each cell represents the candidate locations for the sensors; moreover, the intersection points of cells are considered as the candidate locations for the actors. The obstacles are defined with a square shape and distributed randomly to occupy a single and vacant cell. The obstacle totally blocks the propagation of the radio signals from the sensors to the actors. Here, we assume that a sensor is bound to a single actor.

3.1 Problem Formulation

The set of deployed sensors N, is given as an input to the redesign problem. Each element in the set N is a tuple, n_i, consisting of six ordered parameters, $n_i = <S_i, C_{KxL}, R_i, S_{Ci}, COM_i, B_i>$. The parameter S_i refers to the sensor identification, which was allocated from the sensor device library S. The parameter C_{KxL} represents the physical cell location of the placed sensor within the sensor field. The indices K and L refer the column and row numbers respectively of the floorplan of the sensor field. The parameter R_i indicates the radius of coverage in meters of the placed sensor S_i. The parameter COM_i refers to the communication radius of the placed sensor (S_i) and its unit is in meters. The range of COM_i varies with the consumption of power. The last parameter B_i indicates the current battery level of the placed sensor S_i. Each sensor node S_i has a data generation rate g_i, where g_i is the number of data packets generated by node S_i in unit time [1]. The

parameter S_{Ci} refers to the initial installation and deployment cost in Dollars ($) of the placed sensor S_i.

The objective function of this obstacle insertion problem is to minimize the energy consumption of the network while maximizing the actor-sensor binding ratio as represented as in Equation (1). Where, *E* is the energy consumption of an actor/sensor, *j* and *i* represent the actor and the sensor identification respectively, *N* is the set of sensors, and *M* is the set of actors.

$$\min\left(\sum_{i \in N} E_i + \sum_{j \in M} E_j\right) \tag{1}$$

Each actor/sensor node has a predefined residual energy that gets reduced by data transmission and the transmission energy is calculated using Equation (2). Where *l* is the message length in bits, *d* is the distance between the sensor and the actor. The electronics energy (E_{elec}) depends on many factors such as the digital coding, the modulation, the filtering, and the spreading of the signal, whereas the amplifier energy, $E_{amp} \times d^2$, depends on the distance to the receiver and the acceptable bit-error rate [9].

$$\text{Transmitted Energy } T_E \text{ (joules)} = l \times E_{elec} + l \times E_{amp} \times d^2 \tag{2}$$

The objective function is constrained by several constraints; here we highlight some of the core ones:

1. The number of allocated actors (j) cannot exceed the total number of intersection points (p) and there should be at least a minimum of one actor in the network as given by Equation (3).

$$1 \leq \sum_{j \in C} \beta_j \leq p \tag{3}$$

2. No two actors can have the same interaction point in the grid as stated in Equation (4), where *j* and *m* are two different actors in the set *M*.

$$\text{LOC(j)} \neq \text{LOC(m)} \text{ , where j} \neq \text{m, } \forall \text{ j, m} \in M \tag{4}$$

3. A sensor is either bounded to one actor or not bounded at all as in Equation (5).

$$\sum_{j \in M} \alpha_{i,j} \beta_j = \begin{cases} 0 \\ 1, \text{ for a given sensor } i \in N \end{cases} \tag{5}$$

4. Every actor must have at least one sensor bound to it, and the maximum bounded sensors are given by a threshold value (L_{max}) as in Equation (6).

$$1 \leq \sum_{i \in N} \alpha_{i,j} \leq L_{\max} \text{ for a given actor } j \in M \tag{6}$$

5. The total energy of the network must not go below a given threshold, E_{min} as stated in Equation (7). The E_{min} value is a predefined value, which may be set to 20% of the total initial energy of the nodes.

$$\sum_{i \in N} E_i + \sum_{j \in M} E_j > E_{\min} \tag{7}$$

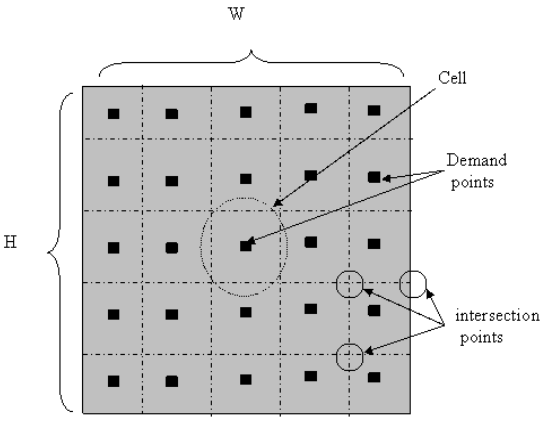


Fig. 1. A service area to be monitored by WSAN

The energy spent during the data transmission is calculated using the energy formula given by Equation (2). To minimize the transmission energy, the distance between a sensor and an actor must be as small as possible and without obstacles. Our proposed algorithm guarantees the binding of every sensor to an actor and also it ensures deploying minimum number of actors to save the networks' energy.

4 Simulated Annealing Overview

In general, the Simulated Annealing algorithm needs four basic components [10]; *Configuration, Move set, Cost function and Cooling schedule*. *Configuration* represents the set of possible solutions over which the algorithm search for a better solution. *Move set* provides a set of permitted moves, which result in the generation of all feasible configurations and it is generated by a neighbor function or by a modification function. As the annealing proceeds, the algorithm moves from configuration to configuration to generate a best solution. The *cost function* is defined to measure the goodness of the generated configuration. The *cooling schedule* anneals the problem from a random solution to a good, frozen placement. Specifically, we need to start at a high temperature (or a heuristic for determining a starting temperature for the current problem) and the cooling strategy determines when the current temperature should be lowered, and by how much the temperature should be lowered, and when annealing process should be terminated.

The Simulated Annealing is a generalization of iterative improvement optimization algorithm, in that it accepts with non-zero but gradually decreasing probability, deteriorations in the energy function. It will iteratively improve a value by moving it step by step through the function space. The idea of SA comes from the work published by Metropolis etc al in 1953 [11]. In 1982, Kirkpatrick et al [12] took the idea of the Metropolis algorithm and applied it to optimization problems. Figure 2 shows the algorithm for Simulated Annealing process, which is continuously attempting to transform the current configuration into one of its

neighbors. It can be viewed as a sequence of Metropolis algorithm as illustrated in Figure 3, which is evaluated at a sequence of decreasing values of control parameter. The process is continued until equilibrium is reached.

```
Simulated_Annealing (S0, T0, α, β, M, Maxtime)
1 begin
2   T = T0;
3   S = S0;
4   Time = 0;
5    repeat
6          call Metropolis(S ,T, M);
7          Time = Time + M;
8          T = α x T;
9          M = β x M;
10   until (Time ≥ MaxTime);
11   output the best solution found;
12 end
```

Fig. 2. An overview of Simulated Annealing

```
Metropolis(S, T, M)
1 begin
2    repeat
3       newS = move(S) or swap(S);
4       Δh = (Cost(newS) − (Cost(S));
5       If((Δh < 0) or (random < e^(−Δh/T) ))
6       then S = newS; {accept the solution}
7       M = M − 1;
8    until (M = 0)
9 end
```

Fig. 3. Metropolis procedure

There are a number of input parameters (S_0, T_0, α, β, M and Maxtime), which are to be utilized during the optimization process within Simulated annealing. Hereby S_0 is the initial solution, T_0 is the initial temperature, α is the cooling rate, β is a constant and it increases the value of M, M represents the number of modifications to the solution during a given temperature and also the time until the next parameters update and Maxtime is the total allowed time for annealing process.

The redesign problem considered in this work is formulated as an optimization problem, with objectives and constraints as presented in section IV. To solve this problem, we define the redesign algorithm within Simulated Annealing (SA), whereby the algorithm finds an optimal solution to cover the maximum number of sensors with obstacles present in the sensing field. At each temperature T, M configurations for a solution will be generated by the neighbor function. The neighbor

function will move the actor with the lowest workload to a random location in a round wrap fashion, in order to minimize the energy consumption.

5 Proposed Redesign Algorithm for the Obstacle Insertion Problem

For a given WSAN with N sensors, M actors (M<< N) and K inserted obstacles, our proposed redesign algorithm initially identifies the blocked sensors within the field.

5.1 Finding Blocked Sensors Methodology

Each obstacle in the sensing area is specified by its four corners as in Figure 4; top left corner, top right corner, bottom left corner, and bottom right corner. Figure 5 shows a sample wireless sensor actor network with obstacles. Here, the Labels S_1,S_2,…,S_7 represents the sensors and the labels A_1 and A_2 represents the actors.

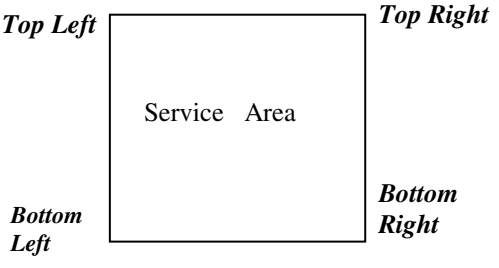


Fig. 4. Corners of an Obstacle.

An obstacle placed on the line-of-sight (LOS) between an actor and a sensor may affect the signal propagation [13]. Figure 6 illustrates the procedures to find out a broken communication link between an actor and a sensor:

1. Specify the obstacle's side(s) which is/are blocking the actor's signals propagation, by specifying the corners of the side(s) (corner *A* and corner *B* illustrated in Figure 6). The corners are determined based on actor's position with reference to the obstacle, and are represented in Table 1.
2. Find vector A <a_i ,a_j> which is the vector directed from the actor's location to corner a.
3. Find vector B <b_i, b_j> which is the vector directed from the actor's location to corner b.
4. Find vector L <s_i ,s_j> which is the vector directed from the actor's location to the senor's location.
5. Assume that the actor's location represent the origin point in the x-y plain (2D plain).

6. Find the angle θ_A between the x-axis and the vector A using Equation (8) [14], Where the value of the numerator is the dot product between A and the x-axis (represented by a vector X <1,0>). The value of the denominator is product of magnitude of A and magnitude of X.

$$\theta_A = \cos^{-1}\left(\frac{A \cdot X}{|A||X|}\right) \tag{8}$$

The magnitude of A is calculated by using the formula: $|A| = \sqrt{a_i{}^2 + a_j{}^2}$ similarly we can find the magnitude of vector X.

7. Find the angle θ_B between the x-axis and the vector B, using the formula in step 6.

8. Find the angle θ_L between the x-axis and the vector L, using the formula in step 6.

9. The sensor is blocked from the actor by an obstacle, if the angle θ_L is between θ_A and θ_B as shown in Figure 6.

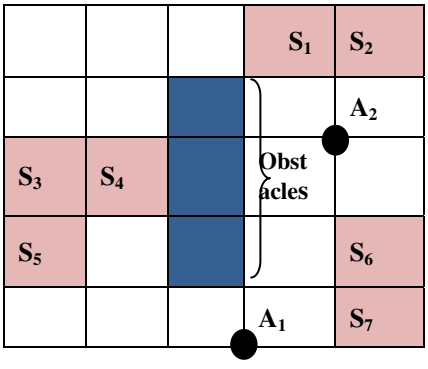


Fig. 5. Wireless sensor actor network with obstacles

5.2 Redesign of Actors' Location

With the set of blocked sensors identified through the procedures in section 5.1, the redesign algorithm analyzes the actors' new location with an objective to minimize the energy consumption of the mobile actors, which restricts the actor movement in a round-wrap fashion to the nearby grid intersection. The actors are relocated with an objective to bind maximum possible sensors located within its communication radii and the actors binding lowest number of sensors is being considered first in the relocation process.

Having defined the redesign problem as an optimization problem within SA, the neighbor function generates M configurations for a solution at each temperature T. The neighbor function moves the actor with lowest workload to a random location

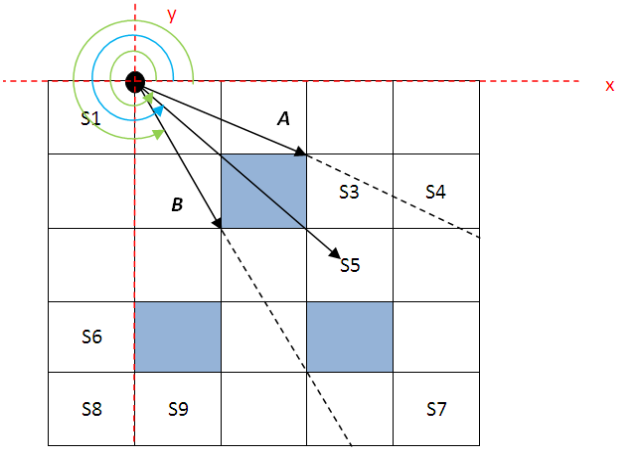


Fig. 6. Sensor 5 is blocked from the actor because θ_L lies between θ_B and θ_A.

in a round-wrap fashion, in order to minimize the energy consumption. With the cost function defined as maximum sensor-actor binding ratio within Simulated Annealing, the redesign algorithm relocates the actors to bind all the sensors.

6 Results and Discussion

We have coded the WSAN redesign experiments in JAVA programming language using Eclipse tool. In this experiment, we have considered a WSAN network of grid size 5 x 5 with 7 sensors, 7 actors, and 3 obstacles selected as initial parameters. The WSAN area is divided into a set of square cells, whereby the width and the height of each cell is taken as 1unit x 1unit. A sensor library file is added to provide sensors' parameters such as the sensor node's ID, their coordinates, radius of coverage of each of the sensors. Similarly, the actors' profile specifies the actors' coordinates, the coverage radius and the cost of the actors. The simulated annealing parameters for this experiment are selected as the initial temperature to be 1000° C, α, the cooling rate to be 1% and the β, the factor increasing the number of iterations to be 0.9%. The maximum time M is selected as 500 units. To start with, the redesign algorithm reads the initial network parameters and the Simulated Annealing parameters.

With the pre-defined initial network parameters, our algorithm computes the initial solution to find out the minimal number of actors necessary to bind the given set of sensors in an obstacles-free environment. For the given network of 7 sensors and 7 actors, our algorithm optimally binds 7 sensors with 2 actors (A_1 and A_2) as shown in Figure 7.

With this initial solution, the redesign algorithm randomly inserts the given 3 obstacles within the vacant cells of WSAN grid. The obstacles are square in shape and each of them occupies a cell in WSAN grid. The redesign algorithm runs its first phase to identify the blocked sensors. Figure 5 illustrates the WSAN network

with inserted obstacles. Since there are 4 corners for the given squared obstacle as illustrated in Figure 4, our algorithms checks all of the 16 possibilities starting from top left corner to bottom right corner as described in Table 1. After listing out the blocked sensors, our algorithm enters into the second phase to redesign the actors' position. The new position of each of the actors is analyzed, and the actors binding lowest number of sensors being considered first. In this experiment, the sensor S_4 is identified as the blocked sensor. The redesign algorithm relocates actor A1 to bind all sensors, thus satisfying 100% binding ratio, as shown in Figure 8. The behavior of the redesign algorithm within SA is shown in Figure 9(a).

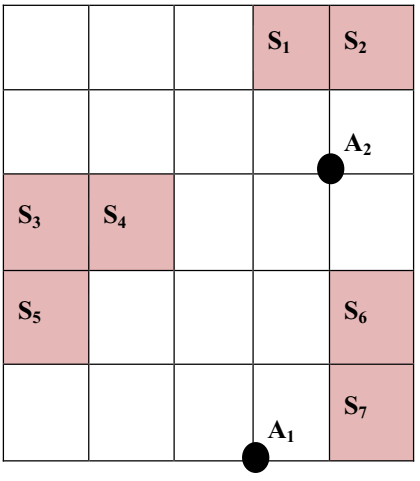


Fig. 7. Initial Solution

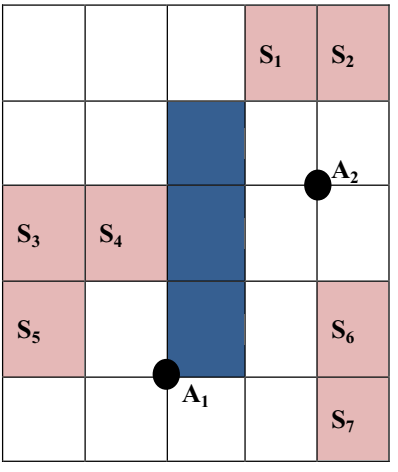


Fig. 8. The redesigned topology with inserted obstacles

Table 1. Actors' position description

side of the obstacle		corner of the obstacle
left	above	Top Right corner
right		Bottom right corner
below	along	Top left corner
above		Bottom right corner

The experiment is repeated for a WSAN network with increased area of 7m × 7m. The initial WSAN parameters are; 15 sensors and 10 actors. The initial solution for the obstacles-free grid generates 5 actors as optimal number to bind all

the 15 sensors. Then, the algorithm inserts the obstacles' within the network. After examining the actor-sensor link with respect to the obstacles, our experiment identifies a single unbounded sensor. Then, the redesign algorithm within Simulated Annealing relocates the actors; thereby the number of unbounded sensors has reached 0, as shown in Figure 9(b). This shows that the proposed redesign algorithm works fine with increased network area, and whereby, the neighborhood function defined within SA improves the cost of the solution.

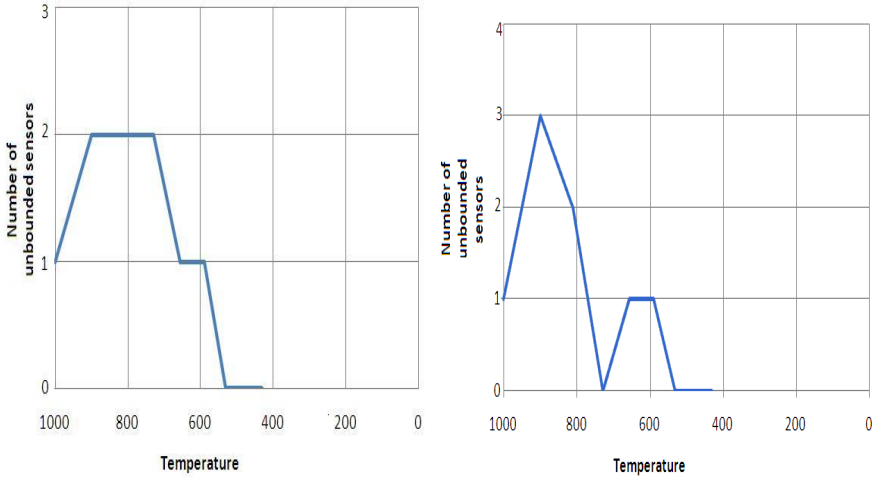


Fig. 9. Behavior of redesign algorithm within Simulated Annealing (a) network size of 5x5 and (b) network size of 7x7.

7 Conclusion

In this work, we have focused on redesign of actors' locations to reestablish the broken communication links between the sensors and the actors, which is due to the insertion of obstacles within the wireless sensor actor networks. The communication link between an actor and a sensor is considered to be broken if there is an obstacle on the line-of-sight of the sensors. We have formulated the obstacle insertion problem as an optimization problem within Simulated Annealing, where the objective is to minimize the energy consumption of the mobile actors in an obstructed sensing field by limiting the movement of the actor to a round-wrap fashion and also to bind maximum number of sensors in the obstructed sensing field. It is observed from the simulation results that our redesign algorithm attains 100% actor-sensor binding ratio with obstacle inserted sensing field with varying network area, and with varying number of actors and sensors.

References

[1] Alrashed, S., Marimuthu, P.N., Habib, S.J.: Optimal Deployment of Actors Using Simulated Annealing within WSAN. In: The Proceedings of the 17th IEEE International Conference on Telecommunications, Doha, Qatar, April 4-7 (2010)

[2] Akyildiz, I.F., Kasimoglu, I.H.: Wireless Sensor and Actor Networks: Research Challenges. Ad Hoc Networks 2(4), 351–367 (2004)

[3] Akkaya, K., Younis, M.: Equation of State Calculations by Fast Computing Machines. In: The IEEE Proceedings of Vehicular Technology Conference, pp. 1–5 (September 2006)

[4] Akkaya, K., Younis, M.: C2AP: Coverage Aware and Connectivity Constrained Actor Positioning in Wireless Sensor and Actor Networks. In: The IEEE Proceedings of IPCCC 2007, New Orleans, April 11-13, pp. 281–288 (2007)

[5] Jourdan, D.B., DeWeck, O.L.: Multi-objective Genetic Algorithm for the Automated Planning of a Wireless Sensor Network to Monitor a Critical Facility. In: The Proceedings of SPIE Defense and Security Symposium, pp. 565–575 (September 15, 2004)

[6] Kannan, A.A., Mao, G., Vucetic, B.: Simulated Annealing based Wireless Sensor Network Localization. Journal of Computers 1(2), 15–22 (2006)

[7] Chatzigiannakis, I., Mylonas, G., Nikoletseas, S.: A Model for Obstacles to be Used in Simulations of Wireless Sensor Networks and Its Application in Studying Routing Protocol Performance. SIMULATION 83(8), 587–608 (2007)

[8] Jardosh, A.P., Belding-Royer, E.M., Suri, S.: Real world Environment Models for Mobile Network Evaluation. IEEE Journal on Selected Areas in Communications 23(3), 622–632 (2005)

[9] Zytoune, O., Fakhri, Y., Aboutajdine, D.: Lifetime Optimization for Wireless Sensor Networks. In: The Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Morocco, May 10-13, pp. 816–820 (2009)

[10] Rutenbar, R.: Simulated Annealing Algorithms: An Overview. IEEE Circuits and Devices Magazine 5(1), 19–26 (1989)

[11] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of State Calculations by Fast Computing Machines. Journal of Chem. Phys. 21(6), 1087–1092 (1953)

[12] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)

[13] Jones, E.: The Basics of Radio Wave Propagation [Organization Website] (2011), `http://ecjones.org/propag.html`, (accessed June 30, 2011)

[14] Jordan, M.I.: An Introduction to Linear Algebra in Parallel Distributed Processing, vol. 1, pp. 365–422. MIT Press, Cambridge (1986)

A Hybrid CS/GA Algorithm for Global Optimization

Amirhossein Ghodrati[1] and Shahriar Lotfi[2]

[1] Computer Engineering Department, College of Nabi Akram, Tabriz, Iran
a.h.ghodrati@gmail.com
[2] Computer Science Department, University of Tabriz, Tabriz, Iran
shahriar_lotfi@tabrizu.ac.ir

Abstract. This paper presents the hybrid approach of Cuckoo Search (CS) and Genetic Algorithm (GA) algorithms for solving optimization problems. In standard CS, each cuckoo lays one egg at a time, but in the proposed hybrid algorithm, in order to lay more eggs we used the genetic algorithms' strategy (Crossover) for their reproduction. According to the cuckoos breeding style, each nest will have one cuckoo at a time. Since there is limitation in number of nests we will have a selection for all cuckoos. Furthermore, we added mutation in order to reduce the chance of eggs to be discovered, because cuckoo birds are specialized in mimicry in color and pattern of the host birds. This theory gets us closer to their real living style. Experimental results are examined with some standard benchmark functions and the results are reported.

Keywords: Cuckoo Search, Global optimization, GA and Hybrid evolutionary algorithm.

1 Introduction

Finding optimal solutions for many problems is very difficult to deal with. The complexity of such problems makes it impossible to look for every possible solution or combination [1]. However, because of their complexity the use of approximation algorithms in order to find approximate solutions is getting more popular in the past few years [2]. Among these algorithms, modern metaheuristics are becoming popular, which leads to a new branch of optimization, named metaheuristic optimization.

Yang and Deb formulated a new metaheuristic algorithm, called Cuckoo Search in 2009 [3]. This algorithm is inspired by life of cuckoo bird in combination with Lévy Flight behavior of some birds and fruit flies. The studies show the CS algorithm is very promising and could outperform some known algorithms, such as PSO and GA.

Genetic algorithm [4] was first introduced by Holland in early 1970's which is inspired by evolution. The process of GA utilizes an optimization technique which

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 397–404.
springerlink.com

allows a global search of the cost surface via the mechanism of the statistical random processes dictated by the Darwinian evolutionary concept [5].

CS and GA are nature inspired metaheuristic algorithms and work well for optimization problems. In this paper, we have tried to get closer to cuckoo birds real life and for this purpose we added GA's operators such as selection, reproduction and mutation into the CS algorithm.

Layeb introduced a new hybridization between quantum inspired and cuckoo search for knapsack problem [6]. This hybrid algorithm achieves better balance between exploration and exploitation and experimental results shows convincing results.

A new hybrid algorithm of cuckoo search and Particle Swarm Intelligence (PSO) was introduced by F. Wang, Lou, He and Y. Wang [7]. The hybrid algorithm seems to remedy the defect of PSO.

Standard CS is discussed in section 2. Section 3 provides an overview of GA. Section 4 gives a hybrid approach of CS with GA and Section 5 presents the detailed experimental results to compare the performance of the proposed algorithm with other algorithms. Conclusion and future works are discussed in section 6.

2 Cuckoo Search

This algorithm is inspired by the special lifestyle of a bird called 'cuckoo'. Cuckoo birds never build their own nests and instead lay their eggs in the nest of other host birds. If host birds discover the eggs are not their own, they will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. On the other hand cuckoo birds carefully mimic the color and pattern of the eggs of host birds. In general, the cuckoo eggs hatch slightly earlier than their host eggs. Once the first cuckoo chick is hatched, the first instinct action it will take is to evict the host eggs by blindly propelling the eggs out of the nest, which increases the cuckoo chick's share of food provided by its host bird. Studies also show that a cuckoo chick can also mimic the call of host chicks to gain access to more feeding opportunity [3].

In nature, animals' path for searching food is in a random way which effectively is a random walk because the next move is based on the current location and the transition probability to the next location. Various studies have shown that fruit flies or Drosophila melanogaster; explore their landscape using a series of straight flight path punctuated by a sudden 90° turn, leading to a Lévy-flight-style pattern. Such behavior has been applied to optimization and optimal search, and preliminary results show its promising capability [3], [8].

Cuckoo search is introduced in three idealized rules: 1) Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest. 2) The best nest with high quality of eggs (solutions) will carry over to the next generation. 3) The number of available host nests is fixed, and a host can discover an alien egg with a probability $P_a \in [0,1]$.

For maximization problem the fitness of a solution can be proportional to the value of its objective function. Other forms of fitness can be defined in a similar way to the fitness function in other evolutionary algorithm. A simple representation

where one egg in a nest represents a solution and a cuckoo egg represents a new solution is used here. The aim is to use the new and potentially better solutions (cuckoos) to replace worse solutions that are in the nests. When generating new solutions $x^{(t+1)}$ for, say cuckoo i, a Lévy flight is performed using the following equation:

$$x_i^{(t+1)} = x_i^t + \alpha \oplus Lévy(\lambda) . \tag{1}$$

Where $\alpha > 0$ is the step size which should be related to the scales of the problem of interest. The product $\oplus$ means entry-wise multiplication. The above equation is essentially the stochastic equation for random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term).

The Lévy flight essentially provides a random walk while the random step length is drawn from a Lévy distribution which has an infinite variance with an infinite mean.

$$Lévy \sim u = t^{-\lambda}, \qquad -1 < \lambda < 3 . \tag{2}$$

Studies show that Lévy flights can maximize the efficiency of resource searches in uncertain environments [9]. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail.

3 Genetic Algorithm

Genetic algorithm is inspired by evolution and is based on the genetic process of biological organisms. Genetic algorithms are ubiquitous nowadays, having been successfully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems [10].

This algorithm, like other algorithms starts with some initial random solutions. In each generation, algorithm starts with a current population which in the first generation, the initial population forms the first current population. Each solution in the population is named chromosome and should be evaluated by the *fitness function*. Henceforth, parents must be selected for the *crossover* operator to generate offspring and constitute the intermediate population. In order to guarantee the diversity of the population, *mutation* operator is applied. Afterwards, a suitable *selection* strategy is performed on the intermediate population to determine which solutions should remain for the next generation's population.

4 Hybrid CS/GA Algorithm

In this section, we explore the details of the proposed hybrid algorithm. In standard CS, each cuckoo lays one egg at a time. According to the life style of cuckoo birds, we see this is not the reality and each cuckoo will lay more than one egg at a time in different nests. Therefore, we used GA crossover operator to

reproduce more eggs for each cuckoo. We select two parents and generate two offspring for them. As mentioned in section 2, cuckoo birds mimic the color and pattern of the host bird in order to protect their eggs from being recognized by the host birds. In the standard CS implementation this theorem has not been considered. Therefore, cuckoo birds will have a mutation in their genes to supply this need. Thus, we added GAs' mutation operator for all cuckoo birds. Cuckoo birds in their real life immigrate to a better environment for their lifetime. In addition, the destination place has limit capacity for them to build their nests. Therefore, we use a GAs' selection strategy to select the survived cuckoos among all the produced cuckoos and eggs. The pseudo-code of the CS/GA is presented as bellow:

```
begin
  Objective function f(x), x = (x_1, ..., x_d)^T;
  Initial a population of n host nests x_i (i = 1,2, ..., n);
  While (t<MaxGeneration) or (stop criterion);
    Generate new eggs by a probability with Crossover
    operation;
    Mutation for all cuckoos by a probability;
    Select cuckoos by a Selection operator;
    Get a cuckoo (say i) randomly by Lévy flights;
    Evaluate its quality/fitness F_i;
    Choose a nest among n (say j) randomly;
    if (F_i > F_j),
      Replace j by the new solution;
    end
    Abandon a fraction (P_a) of worse nests
      [and build new ones at new locations via Lévy
      flights];
    Keep the best solutions (or nests with quality
    solutions);
    Rank the solutions and find the current best;
  end while
  Post process results and visualization;
end
```

5 Evaluation and Experimental Results

In this section, we show experimental results which evaluate our hybrid CS/GA proposed algorithm. We applied the proposed algorithm for some classical benchmark functions which is shown in table 1. Initial range, formulation, characteristics and the dimensions of these problems are listed in table 1. If a function has more than one local optimum, this function is called multimodal. Multimodal functions are used to test the ability of algorithms escaping from local minima. A p-variable separable function can be expressed as the sum of p functions

Table 1. The Benchmark functions

Function	Formulation	Type	DIM	Range
F1 (Ackley)	$-20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	MN	30	[-32,32]
F2 (Dixon-Price)	$(x_1-1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	UN	30	[-10,10]
F3 (Easom)	$-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$	UN	2	[-100,100]
F4 (Griewank)	$1 + \sum_{i=1}^{D}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{2}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	MN	30	[-600,600]
F5 (Powell)	$\sum_{i=1}^{n/k}(x_{4i-3}+10x_{4i-2})^2 + 5(x_{4i-1}-x_{4i})^2 + (x_{4i-2}-x_{4i-1})^4 + 10(x_{4i-3}-x_{4i})^4$	UN	24	[-4,5]
F6 (Rastrigin)	$\sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$	MS	30	[-5.12,5.12]
F7 (Schwefel)	$\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	MS	30	[-500,500]
F8 (Schwefel 1.2)	$\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	UN	30	[-100,100]
F9 (SumSquares)	$\sum_{i=1}^{D} i^2 x_i^2$	US	30	[-10,10]

of one variable. Non-separable functions have interrelation among their variables. Therefore, non-separable functions are more difficult than the separable functions. In table 1, characteristics of each function are given in the third column. In this column, *M* means that the function is multimodal, while *U* means that the function is unimodal. If the function is separable, abbreviation *S* is used to indicate this specification. Letter *N* refers that the function is non-separable.

The simulation results are compared with results of Cuckoo Search [11], a modernized implementation of PSO, which is known as PSO-2007 [12], Differential Evolution (DE) [13] and Artificial Bee Colony (ABC) algorithm [14]. We used 9 benchmark functions in [11] in order to test the performance of the CK, PSO, DE, ABC and CS/GA algorithms. The best values, the mean best values and standard deviation are given in table 2. Convergence diagram for function F5 is

Table 2. Experimental results with CS/GA

		CK	PSO	DE	ABC	CS/GA
F1	Min	4.40E-15	8.00E-15	4.40E-15	2.22E-14	0.00E+00
	Mean	4.40E-15	8.00E-15	4.40E-15	3.00E-14	0.00E+00
	StdDev	0.00E+00	0.00E+00	0.00E+00	2.50E-15	0.00E+00
F2	Min	6.67E-01	6.67E-01	6.67E-01	1.40E-15	6.67E-01
	Mean	6.67E-01	3.27E+01	6.67E-01	2.30E-15	6.98E-01
	StdDev	5.00E-16	9.86E+01	2.00E-16	5.00E-16	7.47E-02
F3	Min	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-9.77E-01
	Mean	-3.00E-01	-1.00E+00	-1.00E+00	-1.00E+00	-9.12E-01
	StdDev	4.70E-01	0.00E+00	0.00E+00	0.00E+00	2.21E-01
F4	Min	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	0.00E+00	9.23E-03	1.11E-03	0.00E+00	0.00E+00
	StdDev	0.00E+00	1.06E-02	3.44E-03	0.00E+00	0.00E+00
F5	Min	7.98E-08	2.11E-05	0.00E+00	2.72E-04	0.00E+00
	Mean	1.51E-07	5.18E-05	0.00E+00	4.49E-04	1.05E-13
	StdDev	5.45E-08	2.16E-05	0.00E+00	6.66E-05	3.60E-13
F6	Min	3.81E-04	1.39E+01	7.96E+00	0.00E+00	0.00E+00
	Mean	1.28E+00	2.80E+01	1.54E+01	0.00E+00	0.00E+00
	StdDev	1.04E+00	7.87E+00	4.96E+00	0.00E+00	0.00E+00
F7	Min	-1.26E+04	-1.04E+04	-1.22E+04	-1.26E+04	-8.27E+03
	Mean	-1.25E+04	-8.93E+03	-1.19E+04	-1.26E+04	-8.01E+03
	StdDev	7.64E+01	9.12E+02	2.04E+02	1.87E-12	9.72E+02
F8	Min	0.00E+00	1.28E-13	0.00E+00	1.08E+01	0.00E+00
	Mean	0.00E+00	7.39E-10	0.00E+00	4.75E+01	0.00E+00
	StdDev	0.00E+00	2.43E-09	0.00E+00	2.32E+01	0.00E+00
F9	Min	0.00E+00	0.00E+00	0.00E+00	3.00E-16	0.00E+00
	Mean	0.00E+00	0.00E+00	0.00E+00	4.00E-16	0.00E+00
	StdDev	0.00E+00	0.00E+00	0.00E+00	1.00E-16	0.00E+00

presented in Figure 1. The average line and best line in convergence diagram are very close to each other. Figure 2 shows the stability diagram for F3. The diagrams in Figure 3 show the functions best value until the algorithm reach to the global minimum. The global minimum values below 10^{-16} are assumed to be 0. 20 runs of algorithm were needed for each function and the maximum evaluation number was 2,000,000 for all functions. The parameter settings of CS/GA are described as follows: The population size is set to 50 and the probability of discovery, P_a is set

to 0.25. Selection method is Tournament with real representation. Crossover method is one-point with the rate of 0.75. Mutation method is set to gene flip with the rate of 0.1. Settings of the algorithms CK, PSO, DE and ABC can be found in [11].

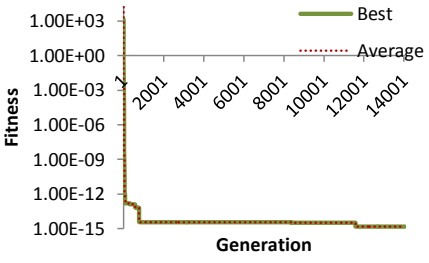


Fig. 1. Convergence diagram for function *F5*

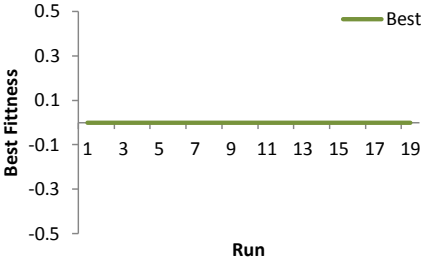
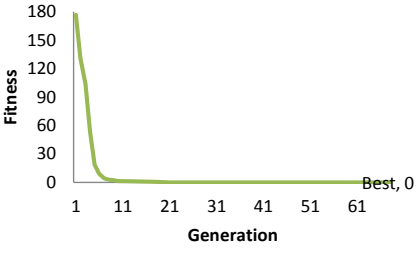
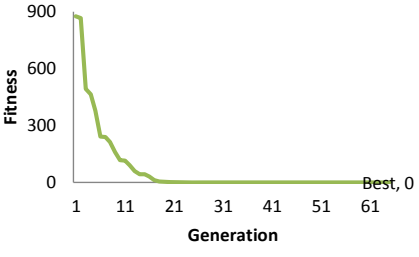


Fig. 2. Stability diagram for function *F3*



(a)



(b)

Fig. 3. Best values of each generation till the achievement of the algorithm to the global minimum for functions: (a) *F4* and (b) *F6*.

6 Conclusion and Future Works

In this paper we combined two nature inspired algorithms and introduced the CS/GA algorithm. With a profound look into the cuckoo birds' life style, we can observe the standard CS algorithm can be extended. Thus, we added crossover, mutation and selection operators to the original CS algorithm.

As seen in the simulation, the genetic algorithm improves the algorithm by the reduction of fitness average in the population. Although, the cuckoo search improves the best solution but have less effect on the fitness average of the population. By adding crossover and mutation more search space are observed during the algorithms' performance.

CS/GA was evaluated on some benchmark functions and the result show that the proposed hybrid algorithm outperforms the CK with more than 65%, PSO-2007 with 75%, DE and ABC with more than 55% of the total time in comparison. As it can be seen in figure 2 the algorithm can successfully reach to the global minimum of the Griewank and Rastrigin functions in generation 69 and 66 respectively.

Our future research would include the hybridization of the CS algorithm with other nature inspired algorithms. Also the future works can include the parallelization of the cuckoo search algorithm to provide considerable gains in term of performance.

References

1. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley Publishing, New Jersey (2010)
2. Coello Coello, C.A., Dhaenens, C., Jourdan, L.: Advances in Multi-Objective Nature Inspired Computing. Springer, Ann Arbor (2010)
3. Yang, X.S., Deb, S.: Cuckoo Search via Lévy Flights. In: Proceedings of World Congress on Nature & Biologically Inspired Computing, pp. 210–214. IEEE Press, Coimbatore (2009)
4. Holland, J.H.: Adoption in Natural and Artificial Systems. University of Michigan, Ann Arbor (1975)
5. Rahmat-Samii, Y., Michielssen, E.: Electromagnetic Optimization by Genetic Algorithms. Wiley Publishing, New York (1999)
6. Layeb, A.: A novel quantum inspired cuckoo search for knapsack problems. International Journal of Bio-Inspired Computation 3, 297–305 (2011)
7. Wang, F., Lou, L., He, X., Wang, Y.: Hybrid Optimization Algorithm of PSO and Cuckoo Search. In: Proc. of 2nd Int. Conference on Artificial Intelligence, Management Science and Electronic, pp. 1172–1175. IEEE Press, Deng Feng (2011)
8. Yang, X.S., Deb, S.: Engineering Optimisation by Cuckoo Search. Int. J. Mathematical Modelling and Numerical Optimisation 1, 330–334 (2010)
9. Yang, X.: Nature-Inspired Metaheuristic Algorithms, 2nd edn. Luniver Press (2010)
10. Kim, D.H., Abraham, A., Cho, J.H.: A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization. Information Sciences 177, 3918–3937 (2007)
11. Civicioglu, P., Besdok, E.: A Conceptual Comparison of the Cuckoo Search, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms. Artificial Intelligence Review (2011), doi:10.1007/s10462-011-9276-0
12. Xin, B., Chen, J., Peng, Z., Pan, F.: An Adaptive Hybrid Optimizer Based on Particle Swarm and Differential Evolution for Global Optimization. Science China Information Science 53, 980–989 (2010)
13. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
14. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)

Minimum Spanning Tree Based *k*-Anonymity

K. Venkata Ramana, V. Valli Kumari, and K.V.S.V.N. Raju

Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India, 530003
{kvramana.auce,vallikumari,kvsvn.raju}@gmail.com

Abstract. Protecting data privacy is a vital problem in micro data distribution when disclosing individual's information. *k-A*nonymization is a popular approach in protecting individuals from being identified when linked to the external data like voters list. A view of k-anonymity is grouping the data values with constraint of minimum number of data points in every group. In this paper we propose Minimum Spanning Tree (MST) approach to achieve k-anonymity. We define hierarchical distance between two tuples as a metric for constructing the edges of MST. Our approach yields an efficient way to achieve k-anonymity and it balances both privacy protection and data utility.

Keywords: Privacy, Inflexion point, Minimum Spanning Tree.

1 Introduction

Several data holders such as hospitals, financial corporations and other organizations publish their microdata for the purpose of data mining, statistical analysis, and other public benefits. This may cause privacy violations for the person specific data. The attributes of the table are divided into three categories, namely identifying attributes, quasi identifiers and sensitive attributes. The attributes like "Name", "SSN" etc., from which we can identify the individuals directly are termed as identifying attributes. The attributes like "Age", "Gender" etc., from which we can potentially identify the individuals are termed as Quasi Identifiers. The attributes like "Salary", "disease" are defined as sensitive attributes. The traditional approach is to de-identify the micro data by removing identifying attributes such as name, SSN etc [9]. When we link the data with publicly available databases like voter list one can re-identify the identity of the individuals. This type of attack is called linking attack. Data generalization and suppression are the most popular techniques being used to avoid the linking attacks [2]. Sweeney proposed *k*-anonymity model [1][2], in which the domain of each quasi identifier attribute is divided into set of equivalence classes and each equivalence class contains at least *k* elements with the same value.

The objective of this paper is to anonymize the data by using the Minimum Spanning Tree (MST). In this approach, first we construct minimum spanning tree (MST), where the vertices of MST are quasi identifier tuples and the edges of the

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 405–413.
springerlink.com

MST represent the hierarchical distance between two tuples. This paper is divided into six sections. Section 2 deals with the related work. In section 3 discusses basic preliminaries which are being used throughout the paper. Section 4 shows the MST k-anonymization and how we achieve anonymity with an example. Section 5 deals with algorithm and performance analysis. Section 6 shows the complexity measure of our approach and finally we conclude in section 7.

Table 1(a). Original Data

ID	Age	Sex	Zip-code	Disease
R1	21	M	13051	Anemia
R2	26	M	13053	Flu
R3	33	F	13063	Cancer
R4	37	F	13068	HIV
R5	45	M	14851	Flu
R6	48	F	14856	HIV
R7	50	M	14865	Cancer

Table 1(b). 2-Anonymized Table

ID	Age	Sex	Zip-code	Disease
R1	[20-30]	M	1305*	Anemia
R2	[20-30]	M	1305*	Flu
R3	[30-40]	F	1306*	Cancer
R4	[30-40]	F	1306*	HIV
R5	[40-50]	Person	148**	Flu
R6	[40-50]	Person	148**	HIV
R7	[40-50]	Person	148**	Cancer

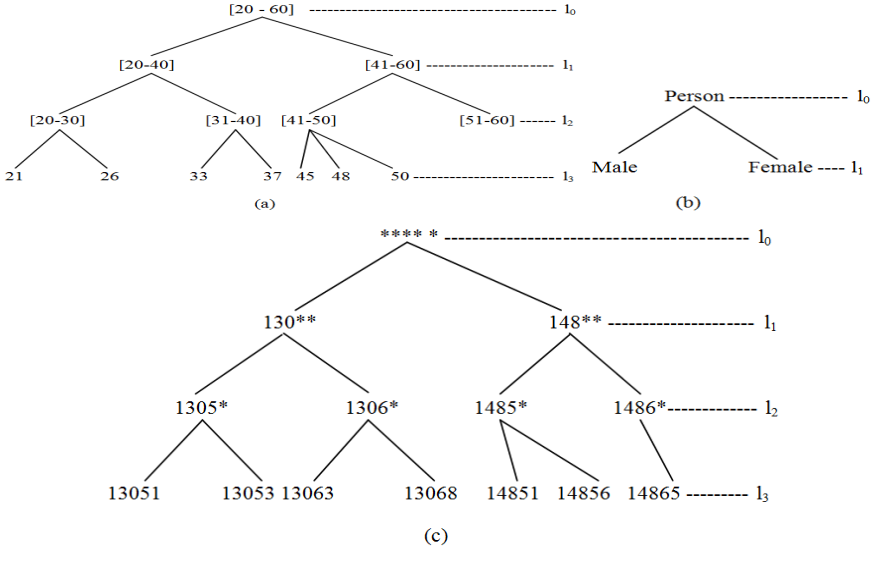


Fig. 1. Domain generalization hierarchies for a) Age, b) Sex, and c) Zip-code

2 Related Work

Sweeney and Samarati introduced k-anonymity [1], in which any tuple in published table is indistinguishable from at least k-1 other tuples with respect to their quasi identifiers. Gagan Agarwal et.al introduced clustering as a technique to anonymize data before publishing them. They considered r-GATHER as a metric to form clusters [3]. Hua Zhu et.al proposed method called density based clustering for

achieving k-anonymity by using *k*-Nearest-Neighbor (KNN) distance to form a set of equivalence classes [5]. Jiuyong Li et.al develop a frame work called KACA to accomplish k-anonymity in which grouping of tuples is done by attribute hierarchical structures [4].

3 Preliminaries

The basic concepts and terminology used in the paper are discussed in this section. Let T be the microdata table that needs to be published. The table contains collection of tuples from domain D = $I \times QI \times SA$, where I be the identifying attributes, QI be the quasi identifiers and SA be the sensitive attributes. For each tuple t ∈ T, the attribute value is denoted as t [A].

Definition 1 (Generalization Hierarchical Distance (GHD)): Let H be the height of the taxonomy tree of an attribute and its levels are 1, 2, 3,... H-1, H. Root level and leaf nodes represent more general and most specific values respectively in the taxonomy tree. Let x and y be the values at two different levels in generalization hierarchy, where y > x. The generalization hierarchical distance between x and y is defined as

$$\text{GHD (x, y)} = \frac{|y|-|x|}{H} \quad \text{where } |y|, |x| \text{ are the height of y and x}$$

For example the value of an attribute Zip code in hierarchy are {13051, 1305*, 130**, *****} as shown in Fig1 (a). Distance between 13051 and 130** is 2/3.The GHD is zero if both values are at same level or at the same leaf nodes and GHD is one if the value lies at the root.

Definition 2 (Generalization Effort of tuple (GE)): Let t = $\{a_1, a_2 \ldots a_m\}$ be the tuple and $t^l = \{a^l_1, a^l_2, \ldots, a^l_m\}$be a generalized tuple. The Generalization Effort is defined as the amount of effort needed to shift a tuple from one level to other level in domain generalization hierarchies of an attributes of tuple. i.e

$$\text{GE (t, t}^l\text{)} = \sum_{i=1}^{m} GHD(a_1, a_1^l)$$

For example, consider the tuple t_2 in table1 (a) and its generalized values t^l_2 in table1 (b). For attributes age, sex and zip-code the GHD values are 0.333, 0 and 0.333 respectively. Therefore, GE (t, t^l) = 0.666.

Definition 3 (Hierarchical Distance between Two Tuples (HDist)): Let t_1 and t_2 be two tuples and their closest common ancestor be the t_{12}. The hierarchical distance between two tuples define as HDist (t_1, t_2) = GE (t_1, t_{12}) + GE (t_2, t_{12}). For example, in table 1(a) the common ancestor of the tuples t_1 and t_2 is {[20-30], M, 1305*}. Therefore, Hierarchical Distance of t_1 and t_2 is 0.666 + 0.666 = 1.332.

Definition 4 (Information Loss): Information loss is a metric which represents the quality of anonymity. In general, the information loss is a ratio between current range of domain and whole domain range. In this paper since we use taxonomy tree for each quasi identifier attribute. The information loss for numerical and categorical dataset is defined in [10].

Definition 5 (Distance between two equivalence classes): Let C_1 be an equivalence classes containing n_1 identical tuples t_1 and C_2 be an equivalence class containing n_2 identical tuples t_2. t_{12} is the closest common generalization of t_1 and t_2 the distance between two equivalence classes is defined as follows [5]

$$\text{Dist}(C_1, C_2) = n_1 \times \text{HDist}(t_1, t_{12}) + n_2 \times \text{HDist}(t_2, t_{12})$$

4 MST *k*-Anonymization

Given a connected, undirected, weighted graph G = (V, E) together with a weight function w: E →R, where V is the set of nodes, E is the set of edges between pair of nodes and weight w (e) specifying weight of each edge e ϵ E. A spanning tree is an acyclic sub graph of a graph G, which contains all vertices from G. The minimum spanning tree (MST) of a weighted graph is minimum weight spanning tree of that graph. Two popular algorithms for finding an MST of weighted graph are Prim's and Kruskal algorithms [6][7][8]. The cost for minimum spanning tree is *O (|E| log |V|)*, where |E| is number of edges and |V| is number of nodes.

The Hierarchical distance based Minimum Spanning Tree (HMST) is spanning tree in which all QI tuples of micro table are treated as vertices and hierarchical distances between two tuples as weighted edges.

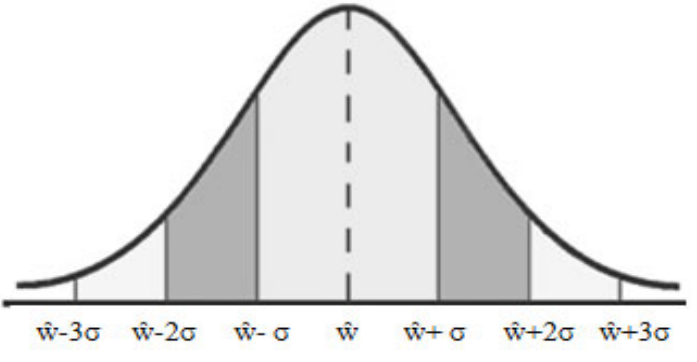


Fig. 2. Normal Distribution curve

Let us consider the weights of the edges of a spanning tree to be a value of a random variable $\mathcal{W}$. We assume that the weights of the spanning tree are normally distributed ($\mathcal{W} \sim N(\hat{w},\sigma)$), where ŵ be the mean and σ be the standard deviation. The normal distribution curve as shown in fig.2 has inflexion points at (ŵ+σ) and (ŵ-σ) i.e., concavity changes to convexity . Therefore the tuples whose weights are below (ŵ+σ) will likely to be close to each other. All such sets of close points tend to form a set of equivalence classes. Hence we acomplish the formation of equivalence classes by removing the edges from MST whose weights are more than threshold (ŵ+σ).

The approach is divided into four sections.

i) Computing the hierarchical distance matrix among all the data points (QI tuples).
ii) Generate MST and sub-trees formation.
iii) Checking k-anonymity in each sub-tree.
iv) Generating equivalence classes.

4.1 Finding Hierarchical Distance Matrix

Given a microdata, we initially identify the quasi-identifiers and construct the hierarchical distance matrix. The distance matrix is constructed in order to generate minimum spanning tree. We construct the distance matrix calculating the distance between the quasi identifier tuples according to definition 3. The hierarchical distance matrix for table 1(a) is shown as in figure 3.

1	0						
2	1.333	0					
3	4.667	4.667	0				
4	4.667	4.667	1.333	0			
5	4.000	4.000	6.000	6.00	0		
6	6.000	6.000	4.000	4.00	1.667	0	
7	4.000	4.000	6.000	6.00	2.000	2.00	0
	1	**2**	**3**	**4**	**5**	**6**	**7**

Fig. 3. Hierarchical distance matrix

4.2 Generate MST and Sub Trees Formation

By using the above distance matrix the minimum spanning tree is constructed [6][8]. For example, in fig 5 the edges of MST and their respective weights are as given below:

Table 2. Edges and their respective weights of MST

Edge	1-2	3-4	3-6	5-6	5-7	1-5
Weight	1.33	1.33	4	1.67	1	4

Mean of all the weights of MST=2.22
Standard deviation of all the weights of MST= 1.27
Inflexion point =2.22+1.27=3.49

Using Table 2 and based on inflection point ($\hat{w}+\sigma$) the inconsistent edges are removed. The inconsistent edges for the above example are 3-6 and 1-5 as shown

in Fig 6. This is because the weights of these two edges are 4 and 4 respectively and are above the inflexion point (3.49). Thus, the three disjoint sub-trees are formed as shown in fig 7.

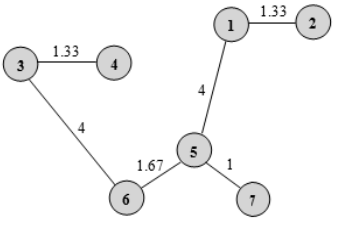


Fig. 5. Minimum spanning tree for Table 1(a)

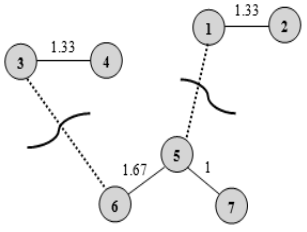


Fig. 6. Subtrees formation for MST in Fig 5

Fig. 7. Finding equivalence classes

4.3 Checking k-Anonymity in Each Sub Tree

For each sub tree generated by above step, test whether it contain *k* data points or not. For example, the required anonymity constraint *k*=2 for the table 1(a). Each sub tree contains at least 2 data points as shown in fig 7.

4.4 Generating Equivalence Classes

To find the equivalence class we initially find the minimum and maximum value of each attribute of a tuple in sub tree and then find the appropriate data range. For example, the data points of the sub-tree 1-2 are {21, M, 13051}, {26, M, 13053} and its minimum common generalized value {[20-30], M, 1305*}.

5 Algorithm

In this Section, we present a Minimum Spanning Tree based k-anonymity (MSTK) algorithm for protecting the data privacy. Given micro data D, our goal is to split data into set of equivalence classes to satisfy *k*-anonymity requirement with minimum information loss. First we construct the Minimum Spanning Tree (MST) of the graph G which is formed from the dataset D (line 3).The weights of

each edge *Weight(e)* in tree is the hierarchical distance between two tuples (nodes). Further we compute mean ($\hat{w}$) and standard deviation (σ) of all the weights of MST (line 6-7). The MST is then partitioned by removing all edges whose weight is greater than inflection point($\hat{w} + \sigma$) (line 9-14), resulting into a set of subtrees ST={$st_1, st_2, ..., st_i$}, each representing an equivalence class. If the

Algorithm Minimum Spanning Tree based *k*-Anonymity(MSTK)

Function MSTK (S, *k*) // S is Dataset
Input: Quasi Identifier data points QI, and required anonymity constraint *k*
Output: Set of equivalence classes, C
Method:

1. **Begin**
2. Transform all n data points into graph with n number of nodes.
3. T ←CreateMST(QI)
4. C ←Φ
5. C^1← Φ
6. $\hat{w}$ ←AvgWeight (T)
7. σ ← StdDev (T)
8. E ← Φ // E is set of edges
9. **for** each edge e ∈ EdgeSet (T) **do**
10. **if** (Weight (e) > $\hat{w}$+ σ) then
11. E ← E ∪ {e}
12. **end if**
13. **end for**
14. ST← PartMST(T, E) // ST is Set of subtrees trees
15. **for** each subtree *st* ∈ ST **do**
16. **if**(|*st*| ≥ *k* and |*st*| ≤ 2*k*-1)
17. C ← C ∪ {*st*} // C is Set of sub trees follows k-anonymity
18. **else if** (|*st*| < *k*)
19. $C^1 = C^1 \cup \{st\}$ // C is Set of sub trees not follows k-anonymity
20. **else if** (|*st*| ≥ 2*k*)
21. MSTK(|*st*|, *k*)
22. **end for**
23. **for** each subtree *st* ∈ C^1
24. evaluate pairwise distance between st_i and all other subtrees of C^1
25. find the equivalence subtree st^1 with smallest distance to st_i contain *k* points
26. Forming modified C^1 contain set of subtress with atleast *k* points
27. **end for**
28. $C_{new} = C \cup C^1$ // k-anonymized equivalence classes
29. Forming equivalence classes of C_{new} using generalization hierarchies
30. Finding the onformation loss
31. **Return** the *k*-anonymized table
32. **end Begin**

number of data points in each subtree lies between *k* and *2k*-1 then these sub trees satisfy optimal k-anonymity property (line 16-17). If the number of data points in the subtree ≥ *2k,* again the sub tree is splitted. Merging of the sub trees whose size is smaller than *k* is done based on distance metric (line 23-27). We then find the minimum and maximum values of each attribute in equivalence class. Finally using domain generalization hierarchies each value is substituted with appropriate ranges and then the information loss is computed.

6 Complexity Measure

Let n be the number of QI tuples. The total time complexity for MSTK involves three stages. First stage determines the complexity for constructing the distance matrix for n QI tuples i.e., $O(n^2)$. In second stage minimum spanning tree is constructed. The time complexity for constructing MST with n vertices and (n-1) edges is O (n log n). Finally after constructing MST, we partition it into set of subtrees whose complexity is O (n). Hence the total time complexity of our approach is $O(n^2)+ O(n \log n)+ O(n) \approx O(n^2)$. Our proposed approach has low information loss than generic approach as we consider closest tuples to form equivalence classes. The proposed algorithm is more practically feasible.

7 Conclusions and Future Work

In this paper we studied k-anonymization as graph based problem and proposed an algorithm based on Minimum Spanning Tree (MST). This approach automatically generates the k-anonymized dataset. However during the anonymization, outliers may be encountered since the process of anonymization is automatic in our approach. In future we would like to study the impact of outliers on the anonymized dataset.

References

1. Sweeney, L.: K-anonymity: a model for protecting privacy. International Journal on Uncertainty 10(5), 557–570 (2002)
2. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and Suppression. International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 571–588 (2002)
3. Gagan, A., Thomas, F., Krishnam, K., Zhu, A.: Achieving Anonymity via Clustering. In: Proc. of PODS 2006 (2006)
4. Li, J., Wong, R.C.-W., Fu, A.W.-C., Pei, J.: Achieving k- Anonymity by clustering in Attribute Hierarchical Structures. In: 8th International Conference on Data Warehousing and Knowledge Discovery, Krakow, Poland, pp. 405–416 (2006)
5. Zhu, H., Ye, X.: Achieving *k*-Anonymity Via a Density-Based Clustering Method. In: Dong, G., Lin, X., Wang, W., Yang, Y., Yu, J.X. (eds.) APWeb/WAIM 2007. LNCS, vol. 4505, pp. 745–752. Springer, Heidelberg (2007)

6. Kruskal, J.: On the shortest spanning sub tree and the travelling sales problem. In: Proceedings of the American Mathematical Society, pp. 48–50 (1956)
7. Nesetril, J., Milkova, E., Nesetrilova, H.: On minimum spanning tree problem. In: DMATH, p. 233 (2001)
8. Prim, R.: Shortest connection networks and some generalization. Bell Systems Technical Journal, 1389–1401 (1957)
9. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, pp. 439–450 (2000)
10. Adusumalli, S.K., Kumari, V.V.: Attribute Based Anonymity for Preserving Privacy. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. Communications in Computer and Information Science, vol. 193, pp. 572–579. Springer, Heidelberg (2011)

Quantum Concepts in Neural Computation

Sitakanta Nayak[1], Shaktikanta Nayak[1], and J.P. Singh[2]

[1] PhD Student, Department of Management Studies, Indian Institute of Technology, Roorkee
[2] Professors, Department of Management Studies, Indian Institute of Technology, Roorkee, Uttarakhand, India-247667
purieskay@gmail.com, {shakti302,jpsiitr}@gmail.com

Abstract. Quantum computation uses quantum mechanical concepts to perform computational tasks and in some cases its results are exponentially faster than their classical counterparts. The main practical reason to investigate the quantum concepts on artificial neural computation is motivated by the success of some quantum algorithms like Grover's search algorithm and Shore's factoring algorithm. The neural networks can be advanced more by the application of quantum computing on artificial neural network (ANN). Since quantum physics is the natural generalization of classical physics, the classical ANN can be generalized to its quantum domain by the combination of classical neural computation with quantum computation. In this paper we have tried to introduce basic quantum mechanical concepts in artificial neural computation.

Keywords: Quantum Computation, Artificial Neural Network, Quantum Neuron.

1 Introduction

Though classical computer is very much powerful in many cases still it has so many shortcomings. Classical computers efficiently process numbers and symbols with relatively short bit registers d < 128.But it has two major limitations to process Patterns, which are wide-band signals having d>100 bits. The first shortcoming is due to the hardware implementation of it i.e. in case of pattern processing, classical computers require enormous number of gates $\propto d^{4.8}$ (According to Rent's Law) to process d-bit registers [9].On the other hand a typical computer program able to perform universal calculations on patterns requires $\propto 2^d$ operators [10].This fact excludes the possibility to use algorithmic approach for pattern processing. ANN can solve this problem because it uses the novel architecture which is able to process long bit strings and learning by example not by programming.ANN can solve complex problems which have poor knowledge domains.ANN also has some other features like non linear parallel distributed processing and robustness. But ANN faces many problems like absence of rule for optimal architectures,

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 415–421.
springerlink.com

time consuming training, limited memory capacity. There is another powerful computation, called Quantum computation based on the quantum mechanical nature of physics [3, 4, 5], which is inherently a powerful parallel distributed processor having exponential memory capacity and easily trainable, but it has severe hardware limitations. Hence quantum mechanical concepts may be introduced in neural computation to overcome the drawbacks of classical computation, quantum computation and artificial neurocomputation [1-2]. In spite of all these, since quantum physics is the natural generalization of classical physics, the classical ANN can be generalized to its quantum domain by the combination of classical neural computation with quantum computation.

2 Quantum Concepts in Computation

The main concepts of quantum mechanics are wave function, superposition, coherence and decoherence, operators, measurement, entanglement, and unitary transformations.

2.1 Superposition

Linear superposition is the physical representation of mathematical principle of linear combination of vectors. The wave function Ψ, which describes the quantum system, exists in a complex Hilbert space .The Hilbert space has a set of states Φ that form a basis .Hence the quantum system can be described mathematically as follows:

$|\Psi\rangle = \sum_i C_i \quad |\Phi_i\rangle$, Where $|\Psi\rangle$ is the superposition of basis states $|\Phi_i\rangle$ and C_i is the complex coefficients.

2.2 Coherence and Decoherence

A quantum system is said to be coherent if it is a linear superposition of its basis states. If a quantum system is in a linear super position of states and interacts with its environment, the superposition is destroyed. This loss of coherence is called decoherence.

2.3 Operator

Operator transforms one wave function in to another .It is represented by capital letter with a hat and is represented by matrix. Mathematically $\hat{A}|\Phi_i\rangle = \mathrm{a}_i|\Phi_i\rangle$, where a_i is the Eigen value and the solution of this equation are called Eigen states, which are used to construct the Hilbert space.

2.4 Entanglement

It is a quantum mechanical phenomenon to establish correlation between two or more quantum states. Entanglement seems to be intuitive from computational point

of view but it is little understood from physical stand point of view. The correlation between states comes into picture when states exist as superpositions .When superposition is destroyed the proper correlation is somehow communicated between the qubits and this communication is achieved due to entanglement. The power of quantum computation derives from the exponential state spaces of multiple quantum bits: just as a single qubit can be in a superposition of 0 and 1, a register of n qubits can be in a superposition of all 2^n possible values. The "extra" states that have no classical analog and lead to the exponential size of the quantum state space are the entangled states. Mathematically entanglement is described using density matrix formalism [11-13].

3 How the Quantum Computer Works

Consider a classical register has 3 bits, then possible outcome will be 2^3=8 (it would be possible to use this register to represent any one of the numbers from 0 to 7 at any instance of time).If we will consider a register of 3 qubits, then the register can represent all the numbers from 0 to 7 simultaneously. A single processor having qubit registers will be able to perform calculations using all possible values of the input registers simultaneously. This phenomenon is called quantum parallelism. Unlike classical bits qubits can exist simultaneously as 0 and 1, with probability for each state given by numerical coefficients. A quantum computation [11, 12] is a collection of the following three elements:

1. A register or a set of registers.
2. A unitary matrix, which is used to execute the quantum algorithms.
3. Measurements to extract information.

Hence quantum computation is the set $\{H, U, \{M_m\}\}$, where H is the Hilbert space of n-qubit register. $U \in U(2^n)$ is the quantum algorithm. $\{M_m\}$ is the set of measurement operators .Actual quantum computation processes are very different from that of classical counterpart. In classical computer we give input data through the input devices .The input signal is stored in computer memory, then fed into the microprocessor and the result is stored in memory before it is displayed in the screen .Thus the information travels around the circuit .In contrast information in quantum computation is first stored in a register and then external fields such as oscillating magnetic fields, electric fields or laser beams are applied to produce gate operations on the register. These external fields are designed so that they produce desired gate operation, i.e. unitary matrix acting on a particular set of qubits .Hence information sits in the register and they are updated each time the gate operation acts on the register.

4 Physical Experiment to Realise Quantum Concepts on Neural Network

The architecture of double-slit experiment [15] provides the basis for quantum analog of artificial neural network. A photon gun sends photons to the Double slit

apparatus. The experiment can be explained in quantum mechanical terms through the wave-particle duality. The photons travel as waves but collapses to a particle when the detection screen is reached.

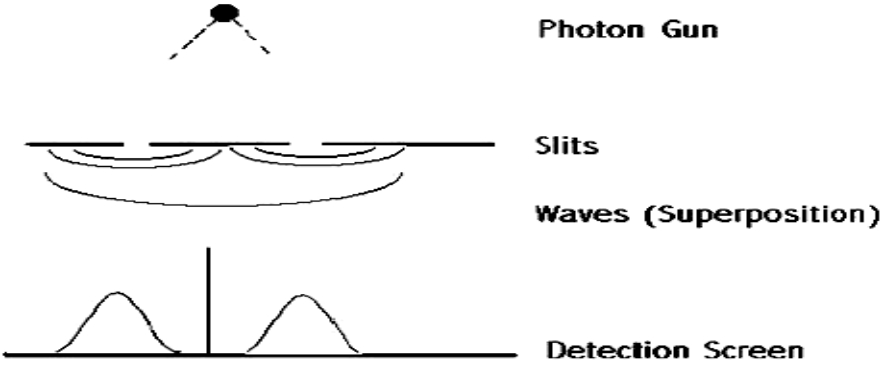


Fig. 1. Double slit experiment

The photon gun is equivalent to the input pattern, the slit is equivalent to input neuron, the waves between the slits and the screen are equivalent to the connections between the input and output neurons and the screen is equivalent to the output neurons. Refer Table-1 for better understanding.

Table 1. Quantum equivalent of Artificial Neural Network

Artificial Neural network	Double slit Expt.	Quantum Correspondence
Pattern	Photon gun	Quantum Register holds input Pattern
Input Neuron	Slits	Entanglement/unitary evolution of input pattern
Connection	Waves	Superposition of waves created by input pattern
Output neuron	Detector screen	Pattern Interference
Weight	Link	Phase shift

5 Necessary Functionalities Required for a Quantum Neuron

While the classical artificial neuron has its natural analog, the nervous cell, there is no natural analog of quantum artificial neuron .Therefore in order to understand the effect of quantum concepts on neural computation, it is necessary to imagine the functionalities of a quantum neuron.

A quantum neuron should be a unit which can be characterised both as a neural and also as a quantum to satisfy both neural technology and quantum technology [14]. Neural computing can process both binary signals (bit strings) and analog signals (patterns having continuous components).Qubit is the quantum analog of a bit .Hence the quantum neuron should process not only qubits but analog signals too.

The development in classical neural computing has been partly connected with the generalization of the form of input and output (real, complex, vector, binary etc.)..Hence it can be expected that a quantum neuron should also have arbitrary input and output and even new types of input and output. As in the quantum domain a superposition of states can be created, it is not obvious in advance if quantum neurons must be connected by networks. So the nature of inputs and outputs may be rather different.

6 Mathematical Representation of a Quantum Neuron

Here we will look how to quantise a neural network .The way we are taking is to replace each neuron with a qubit. Instead of having the neural state laying between in the range of $0 \leq Y \leq 1$ our qubit will be in the superposition state $|\Psi\rangle = \alpha\ |p\rangle + \beta\ |q\rangle$, where the complex coefficients α and β satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$.Hence $|\Psi\rangle$ is the superposition of the basis states $|p\rangle$ and $|q\rangle$ and can be represented in an infinite number of ways simply by varying values of the coefficients α and β subject to the normalization condition. The next common question is how to quantise the synaptic weights? The obvious answer is the amount of entanglement shared between two quantum neurons.

6.1 Artificial Neuron

A classical artificial neuron is an adjustable unit performing, in general, a non linear mapping of the set of many, N, input (perceptron) values x_1 ..... x_N to a single output value Y. The output value of a classical perceptron is [6]

$$Y = f\left(\sum_{j=1}^{N} w_j x_j\right) \tag{1}$$

Where f ( ) is the perceptron activation function and w_j are the weights tuning during learning process. The perceptron learning algorithm works as follows:

1. The weights w_j are initialized to small random numbers.
2. A *pattern* vector ($x_1 \ldots\ldots x_N$) is presented to the perceptron and the output Y is generated.
3. The weights are updated according to the rule

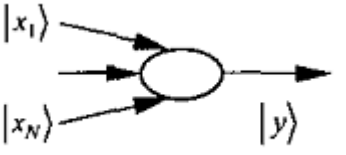
$$w_j(t+1) = w_j(t) + \eta(d - Y) \tag{2}$$

Where *t* is discreet time, *d* is the desired output provided for training and $0<\eta<1$ is the step size.

6.2 Quantum Neuron

Many researchers use their own analogies in establishing a connection between quantum mechanics and neural networks [7]. The power of ANN is due to their massive parallel, distributed processing of information and due to the nonlinearity of the transformation performed by the network nodes (neurons) .On the other hand; quantum mechanics offers the possibility of an even more powerful quantum parallelism which is expressed in the principle of superposition. This principle provides the advantage in processing large data sets.

Consider a neuron with N inputs$|x_1\rangle$.....$|x_N\rangle$ as shown in the figure.



$|x_j\rangle$ is a quantum bit of the form

$$|x_1\rangle = a_j|0\rangle + b_j|1\rangle = (a_j, b_j)^{\mathrm{T}} \tag{3}$$

Where $|a_j|^2 + |b_j|^2 = 1$.

The output $|y\rangle$ can be derived by the rule [8]

$$|y\rangle = \hat{F} \sum_{j=1}^{N} \widehat{w_j} |x_j\rangle \tag{4}$$

Where $\widehat{w_j}$ is a 2× 2 matrices acting on the basis $\{|0\rangle, |1\rangle\}$ and $\hat{F}$ is an unknown operator that can be implemented by the network of quantum gates. Let us consider $\hat{F} = \hat{I}$ be the identity operator .The output of the perceptron at the time t will be

$$|y\,(t)\rangle = \hat{F} \sum_{j=1}^{N} \widehat{w_j}(\mathrm{t}) |x_j\rangle \tag{5}$$

In analogy with classical case equation (2) we can update the weights as follows

$$\widehat{w_j}(\mathrm{t+1}) = \widehat{w_j}(\mathrm{t}) + \eta(|d\rangle - |y\,(t)\rangle)\langle x_j| \tag{6}$$

Where $|d\rangle$the desired output .It is is can be shown eq. (6) derives the quantum neuron in to desired state$|d\rangle$. Using rule (6) and taking modulo-square difference of real and desired outputs, we can get

$$\| |d\rangle - |y\,(t+1)\rangle \|^2 = \left\| |d\rangle - \sum_{j=1}^{N} \widehat{w_j}(\mathrm{t}+1) |x_j\rangle \right\|^2$$

$$= \left\| |d\rangle - \sum_{j=1}^{N} \widehat{w_j}(\mathrm{t}) |x_j\rangle + \eta \left(|d\rangle - |y\,(t)\rangle \langle x_j|x_j\rangle \right) \right\|^2$$

$$= \left\| |d\rangle - |y\,(t)\rangle - \sum_{\mathrm{j}=1}^{\mathrm{N}} \eta(|d\rangle - |y\,(t)) \right\|^2 = (1-N\eta)^2 \| |d\rangle - |y\,(t)\rangle \|^2 \tag{7}$$

For small η ($0< \eta< \frac{1}{N}$) and normalized input states the result of iteration converges to the desired state $|d\rangle$.The whole network can be then composed from the primitives elements using the standard rules of ANN architecture.

7 Conclusion

The introduction of quantum concepts to neural computation opens a new research direction in the field of neural computation. Here in this paper we have described the major limitations of classical computation, neural computation and quantum computation. We have analyzed a physical experiment-Double Slit Experiment-that can motivate to introduce quantum concepts in neural computing. Finally we have shown a mathematical model of quantum neuron equivalent of classical neuron.

References

[1] Kak, S.C.: On Quantum Neural Computing. Information Science 83, 143–160 (1995)
[2] Menneer, T., Narayanan, A.: Quantum-inspired Neural Networks. Tech. Rep. R329, Univ. of Exeter (1995)
[3] Benioff, P.A.: Quantum Mechanical Hamiltonian Model of Turing Machine. J. Stat. Phy., R. 29(3), 515–546 (1982)
[4] Feynman, R.P.: Simulating physics with computers. Inr. J. of Theo. Physics 21(6/7), 467–488 (1982)
[5] Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Pmc. of Roy. Soc. of London Ser. A 400, 97–117 (1985)
[6] Minsky, M., Papert, P.: Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge (1969)
[7] Ezhov, A., Ventura, D.: Quantum Neural Networks. In: Future Directions for lnielligent Systems and Information Sciences, pp. 213–234. Springer, Heidelberg (2000)
[8] Altaisky, M.V.: Quantum neural network, ArXiv: quant-ph/O l07012 (2001)
[9] Landman, B.S., Russo, R.L.: IEEE Transactions on Computers 20, 1469–1479 (1971)
[10] Ezhov, A.A., Shumsky, S.A.: Neurocomputing and its application in economics and business. Moscow Engineering Physics Institutes, Moscow (1999)
[11] Nakahara, M., Tetsuo, O.: Quantum Computing, From Linear Algebra to Physical Realizations. CRC Pres, Boca Raton (2008)
[12] Nielsen, M.A., Chuang, I.L.: Quantum computation and Quantum Information Chuang. Cambridge University press (2002)
[13] Rieffel, E., Polak, W.: An Introduction to Quantum Computing for Non- Physicists. ACM Computing Surveys (CSUR) 32(3), 300–335 (2000)
[14] Ezhov, A.A., Berman, G.P.: Introduction to Quantum Neural Technologies. Rinton Press, New Jersey (2003)
[15] Menneer, T., Narayanan, A.: Quantum artificial neural network architectures and components. Information Sciences (128), 231–255 (2000)

A Novel Constructive Neural Network Architecture Based on Improved Adaptive Learning Strategy for Pattern Classification

S.S. Sridhar[1] and M. Ponnavaikko[2]

[1] School of Computing, SRM University, Chennai, India
sridhar.srm@gmail.com
[2] Vice Chancellor, SRM University, Chennai, India
ponnav@gmail.com

Abstract. Constructive neural network algorithms provide optimal ways to determine the architecture of a multi layer perceptron network along with learning algorithms for determining appropriate weights for pattern classification problems. In this paper the possibility of developing a novel Constructive Neural Network architecture with improved adaptive learning strategy is proposed and analyzed. The new Multi category Tiling Constructive Neural Network architecture and the existing Tiling architecture are tested on machine learning datasets. The performance of the new learning strategy on Multi Category Tiling architecture was found to be comparatively better than when applied on the existing Tiling architecture.

Keywords: Constructive Neural Networks, Adaptive Learning, Multi-Category Tiling architecture, Pattern Classification.

1 Introduction

Artificial Neural Networks are biologically-inspired models of computation. They are networks with elementary processing units called neurons massively interconnected by trainable connections called weights, to perform computations for pattern classification tasks [1]. The Feedforward Neural Networks have been proven to be successful in the field of pattern recognition. They perform pattern recognition by learning an approximation of the underlying function that generates the pattern data. There are a number of practical problems that arise in the implementation of Feedforward Networks. The difficulty of learning algorithm like the back propagation algorithm is that the network architecture is fixed throughout the training process and the slow rate of convergence, since the weights are adjusted at each stage of training. The architecture of a neural network can have significant impact on its performance in any particular application [2].

In multilayer perceptrons, deciding the number of neurons in the hidden layers is a very important part of overall neural network architecture. Though these

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 423–433.
springerlink.com

layers do not directly interact with the external environment, they have a tremendous influence on the final output. The optimal nodes in a network for valid generalization depends on training samples, dimension and complexity of the function used [3]. Automated techniques to fix the number of hidden layers and neurons in those layers, so as to optimize the network architecture can be achieved by two ways, namely Network Pruning or Network Growing. A survey of pruning algorithms with their parameters are discussed in [4] and a comparative analysis of two pruning techniques and parameters are discussed in [5]. Growing algorithms on the other hand are called Constructive learning algorithms, start with minimal architecture and is build during the training process. They provide an optimal way to determine the architecture of a multilayer perceptron network trainable with learning algorithms to determine appropriate weights. The advantages of CoNN are the choice of network topology is dynamically determined during training, they provide guaranteed convergence to zero classification errors on non contradictory finite data sets, use of elementary threshold neurons for training, no extensive learning parameters to be fine tuned, and, by restricting its architectural size, it is less complex and easy to generalize.

A number of architectures and learning algorithms were found in literature which had its merits and demerits in addressing both the convergence rate and generalization capability. This paper attempts to explore the possibility of a novel architecture using an improved adaptive learning strategy to address the above said parameters specifically for selected pattern classification datasets as found in machine learning repository.

2 Literature Review

2.1 Review of Existing Constructive Architecture and Learning Algorithms

A number of Constructive algorithms that incrementally construct networks of threshold neurons for 2-category pattern classification tasks have been proposed in the literature. Ash [6] proposed *Dynamic Node Creation* which performed well on some simple classification tasks, but was not suitable for finding an optimal solution on complex tasks. Mezard & Nadal's [7] *Tiling* algorithm in which the convergence is guaranteed as it always produces a finite number of layers. The tradeoff between generalization capability and number of layers and the choice of weight training algorithms was considered to be the limitation of this method. Gallant [8] proposed the *Tower* and *Pyramid* algorithms which constructs a tower of Threshold Logic Units. Unfortunately the performance of the network degrades as number of instances increase for getting an optimal solution. The *Upstart* algorithm proposed by Frean [9] constructs a binary tree of threshold neurons imposes a set of inductive and representative biases on the algorithm [10].

Fahlman and Lebiere [11] proposed *Cascade Correlation* networks applicable to problems with continuous output variables uses Quick Prop algorithm for

achieving faster learning. It also performs poorly on regression tasks and function approximation tasks. Marchend et al [12] introduced *Sequential* algorithm, in which it trains neurons to sequentially exclude patterns belonging to one class from other. The limitation of this method was that of implementing 'the maximum cluster size strategy' in finding absolute minimal network. Burgess [13] proposed *Perceptron Cascade* algorithm which is similar to upstart algorithm and Cascade Correlation in freezing weights which puts limitation on learning more training samples. F.M.Frattale Mascioli and G. Martinelli [14] introduced *Oil-Spot* algorithm which uses only binary valued inputs. Yang.J et al [15] introduced *DistAI* which is based on inter-pattern distance and constructs a single hidden layer of spherical threshold neurons. The tradeoff of this technique is that of maintaining the inter-pattern distance matrix during learning and thereby memory size increases quadratically with size of the training set.

Leonardo Franco et al [16] introduced a new constructive architecture to learn specifically the breast cancer information and the extension of generalization complexity measure for real valued input datasets was found to be feasible [17].

Some of the famous algorithms for training individual Threshold Logic Units in constructive neural networks which appear in literature are

The *Perceptron Learning rule* and its variants like the *Pocket algorithm with Ratchet Modification (PRM)* proposed by Gallant [8], *Thermal Perceptron* Algorithm (TPA) proposed by Frean[18], *Barycentric Correction Procedure* (BCP) proposed by Poulard [19][20].

The pioneering work of Parekh.R et.al [21] with multicategory real valued inputs for constructive neural network learning algorithms like MTower, MPyramid, MTiling, MSequential, MPerceptron Cascade and MUpstart were converging to zero classification errors. The above architectures were used for classification problems' involving real-valued input attributes for multiple output classes. They analyzed the performance of the aforesaid algorithms on various parameters like convergence rate, network size, generalization performance and training speed. The Tiling and Sequential algorithms do not need preprocessing of pattern for convergence, rather in Tiling; the network itself is a vector quantizer. In addition, the Tiling architecture ensures faithful representation of training set which is a necessary condition for convergence. Parekh.R et.al [22] proved the convergence of MTiling and MPyramid algorithms on practical pattern classification problems. The convergences of the two algorithms were established by showing that each modification of the network topology guarantees the existence of weights for reducing the classification error. It was observed that the constructive learning algorithms take between 1.5 and five times as long as the perceptron algorithm to train on the datasets considered by them.

A New Constructive Algorithm (NCA) was proposed by Monirul Islam [23], which uses different training sets for hidden neurons to learn different aspects of the input, thereby freezing the neurons for functional adaptation. Neural Networks have been applied in Graphs such that the incremental approach eliminates the need to introduce cycle dependencies [24].

2.2 Issues in Constructive Architectures and Learning Algorithms

On surveying the various Constructive Neural Network topologies and their learning algorithms, the following important issues were found.

- There are no known efficient methods for determining the optimal network topology for a given problem. Too small networks are unable to adequately learn the problem well, while large networks tend to over fit the training data which results in poor generalization performance.
- Most constructive learning algorithms are designed for binary (or bipolar) valued inputs. Parekh et al [21] used real-valued attributes from smaller datasets for the existing constructive neural network topologies, which converge to fairly compact networks with zero classification errors. Both convergence and generalization for compact networks was a challenging task.
- Standard Learning algorithms like Perceptron Learning Rule, Pocket algorithm, Thermal Perceptron Algorithm, Barycentric Correction Procedure were found to be converging when the training set was linearly separable, but on non-linearly separable problems they performed reasonable better. So designing a learning algorithm for constructive neural networks which produce optimal set of weights and learning good number of training samples remain an open ended problem.
- The two important parameters like the Convergence and Generalization were found to be the key issues in designing constructive neural networks for pattern classification tasks. One parameter needs to be compromised for the other which is dependent on the dataset chosen.
- Multi-Category Tiling (MTiling) network topology offers a temporary solution for the above problems than other CNN topologies in reducing the network size and number of misclassifications. But, its performance was found to be deteriorating on some cases as reported by Parekh et al [22]. Further the MTiling network topology uses either the perceptron learning algorithm or its variants like the PRM, TPA and BCP on threshold neurons, which may be responsible for its poor performance.

3 Methodology

3.1 Design of New MTiling Constructive Neural Network

The new MTiling Constructive Neural Network architecture as given in Fig.1 is constructed using the strategy as given below

- The objective of the proposed architecture is to map the patterns present in the input dataset with N attributes, to M different classifications to achieve reduced network size and reduced misclassifications.

- The strategy adopted for achieving this objective is to initially design a network with two layers namely one input layer I=0 with N neurons for the number of input attributes, and an output layer I+1 with M neurons for the number of output categories, which are also called as master neurons.
- Two types of weights (connections) exist between the adjacent layers namely bottom-up weights and top-down weights.
- The bottom-up weights and top-down weights are initialized based on Improved Adaptive Learning Strategy for Master neurons namely, I-ALS[1] algorithm which will be discussed in the next section.
- Calculate the activations of master neurons in layer I.
- Choose the winner node in layer I using Winner-Take-All (WTA) strategy. The winner master neuron in layer I project its output through top-down weights to layer I-1 for performing a vigilance test as given in I-ALS.
- A gain control signal checks for input validity. If the vigilance test fails, which indicates misclassification, then it prevents the master neurons from being trained for current pattern.
- During the training process, if the current output layer with M master neurons does not classify the patterns correctly, then 'k' ancillary neurons are added to the current output layer, as given in the MTiling algorithm.
- The ancillary neurons are trained using misclassified patterns with the weight setting as given in Improved Adaptive Learning Strategy Algorithm for ancillary neurons namely I-ALS[2] algorithm
- If a layer has M+k neurons, which refers to the existence of misclassifications, then a new output layer is created with M master neurons and the previous output layer with M+k neurons will act as hidden layer.
- The new output layer and hidden layer are connected and trained
- The process of creating new output layer and training the weights is repeated until, either the output layer represents the desired classifications with M master neurons, or the maximum number of iterations specified is exhausted.

3.2 Design of Learning Algorithms

The design of learning algorithm for the New MTiling architecture follows the adaptive learning strategy. This strategy follows the principle of using the pattern itself as the weights during training. The bottom-up weights are initialized using the normalized version of the input pattern, then during the learning process; they are updated using Kullback-Leibler (KL) error measure. The KL error measures the divergence of actual outputs from desired outputs using cross entropy as proposed by Neelakanta [25]. In the context of datasets which are linearly not separable, the cross entropy error measure depicts the mutuality between two distributions of actual and target outputs. A Vigilance test is performed to locate the neuron which is closer, so that the pattern can be mapped to its respective classification.

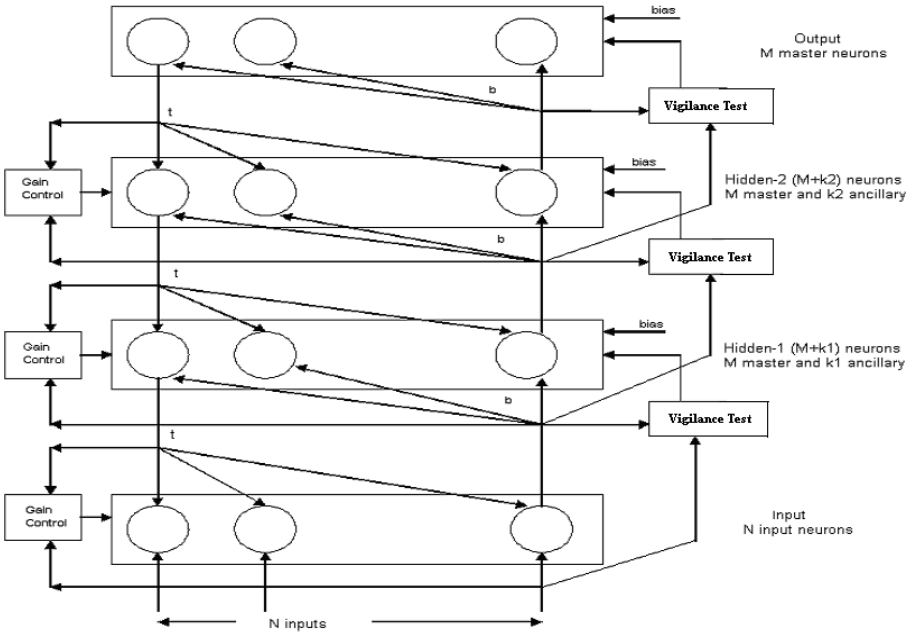


Fig. 1. New MTiling Constructive Neural Network Architecture

By using Adaptive learning strategy, the number of neurons in the network could be reduced, thereby reducing the number of weight connections, which addresses the generalization capability. The number of misclassifications occurring at every hidden layer could be minimized, since performing a vigilance test will add ancillary neurons of the respective classification. This strategy reduces the network size, thereby addressing the convergence rate.

Improved Adaptive Learning Strategy (I-ALS[1]) algorithm for training master neurons

Algorithm:

1. Initialize the bottom-up and top-down weights as follows:

Bottom-up weights:

$$b_{0,I_j} = \frac{-1}{\sum_{L=1}^{N}(P_L^r)^2} \text{ (for bias input)} \quad b_{I-1_i,I_j} = \frac{P_i^r}{\sum_{L=1}^{N}(P_L^r)^2} \text{ (for other inputs)}$$

Top-down weights: $t_{I_j,I-1_i} = 1$

2. Activations of master neurons are calculated as

$$net_{I_j}^r = \sum_{i=1}^{N} b_{I-1_i,I_j} P_{I-1_i}^r \qquad O_{I_j}^r = f(net_{I_j}^r) = \frac{1}{1+e^{-net_{I_j}^r}}$$

3. Error calculations according to Kullback-Leibler measure of cross entropy

$$E_1 = \sum R_i \log\left(R_i / Q_i\right) \quad \text{and} \quad E_2 = \sum Q_i \log\left(Q_i / R_i\right)$$

Where R_i refers to probability associated with actual outputs, and Q_i refers to probability associated with target outputs.

$E = (E_1 - E_2)$ Where E refers to divergence of actual output from their targets

4. Weight updation rule

$$b_{I-1_i,I_j} = \left[\frac{b_{I-1_i,I_j}(new) + b_{I-1_i,I_j}(old)}{2}\right] + \alpha E \quad \text{where } 0 < \alpha \leq 0.4$$

5. Vigilance test

a. Calculate Euclidean distance between actual output and each of the master neurons.

b. If distance from a cluster C_a is greater than 2/3rd distance from another cluster C_b

c. Then, Weights are updated for class C_b to adapt the pattern

d. Else, Collect the misclassified pattern of class C_b

e. Disable the master neuron C_b for further trails on that pattern.

6. Bottom-up weights between new master neurons in layer I and master neurons in layer I-1 are initialized as

$$b_{0,I_j} = A_{I-1} \quad \text{(for bias input)}$$

$$b_{I-1_k,I_j} = D_k \quad \text{(for k=1, 2,…., M except j)}$$

$$b_{I-1_i,I_j} = A_{I-1} \quad \text{(for } j^{th} \text{ neuron)}$$

where D_k is the output of k^{th} master neuron in layer I-1 and A_{I-1} is the number of neurons in layer I-1 which includes the ancillary neurons.

Improved Adaptive Learning Strategy (I-ALS^2) algorithm for training ancillary neurons

Algorithm:

Let the misclassified patterns be P^s

Bottom-up weights between input layer I-1 and layer I with ancillary neurons be

$$b_{0,I_j} = \frac{-1}{\sum_{L=1}^{N}(P_L^s)^2} \quad \text{(for bias input)}$$

$$b_{I-1_i,I_j} = 0 \quad \text{(for all neurons except k)}$$

$$b_{I-1_k,I_j} = \frac{P_k^s}{\sum_{L=1}^{N}(P_L^s)^2} \quad \text{(for maximum attribute k)}$$

Bottom-up weights between new master neurons in layer I and ancillary neurons in layer I-1 are trained as

$$b_{I-1_i, I_k} = D_k^r \quad (\text{where } D_k^r \text{ is the output of ancillary neuron for the respective class})$$

$$b_{I-1_i, I_j} = 0 \quad (\text{for other master neurons except k})$$

4 Applications

The new architecture and learning algorithms were tested on a selected real-valued Machine Learning Datasets available at the UCI Machine Learning Repository [26], namely the Pima Indian diabetic dataset, Wine dataset, Mamographic dataset and Teaching Assistant Evaluation datasets. Since, the number of instances in the datasets was found to be very small, bootstrapping methods were applied to generate more instances of the dataset. The datasets were separated as training set and testing set, and were divided into equal sized folds, of which 60% of instances were used for training, and remaining 40% of instances were used for testing. These dataset instances were applied separately to the new MTiling architecture as well as the existing Tiling architecture on different trails. The performance of the new architecture in comparison with the existing MTiling architecture were analyzed on the following parameters

- number of hidden layers and hidden neurons,
- total number of weight connections,
- total neurons in the network,
- number of misclassifications, and
- number of ancillary neurons.

On applying the Pima Indian Diabetic dataset the new MTiling architecture was found to produce less number of hidden neurons, total weight connections and total neurons when compared with existing Tiling architecture as given in Fig. 2a. The number of misclassifications increased, while number of ancillary reduced. In case of Wine dataset the new MTiling architecture was found to reduce the misclassifications very significantly with nearly one fifth of the ancillary neurons when compared to existing Tiling architecture as given in Fig. 2b. The behavior of existing Tiling architecture was found to be very poor in generating more weight connections, total neurons and misclassifications when compared to new MTiling architecture. On applying Mamographic dataset the new MTiling architecture was found to produce less number of hidden neurons, total weight connections and total neurons when compared with existing Tiling architecture as given in Fig. 2c. There were no misclassifications with reduced number of ancillary neurons in new MTiling architecture. In case of Teaching Assistant Evaluation dataset the new MTiling architecture was found to produce less number of hidden neurons, total weight connections and total neurons when compared with existing Tiling architecture as given in Fig. 2d. There was less number of misclassifications and ancillary neurons in new MTiling architecture when compared to existing architecture.

The results show that the new architecture generates less number of hidden neurons, less number total neurons with reduced weight connections in most of the datasets used for study. This may be due to the weight training strategy adopted during training,. This strategy improves the generalization capability of the network on unseen patterns in the dataset.

Significantly there were very less number of misclassifications which attributes to classifying more patterns at higher layers than layers below it. This strategy of reduction of misclassifications helps in making the network converge little faster reducing the training time.

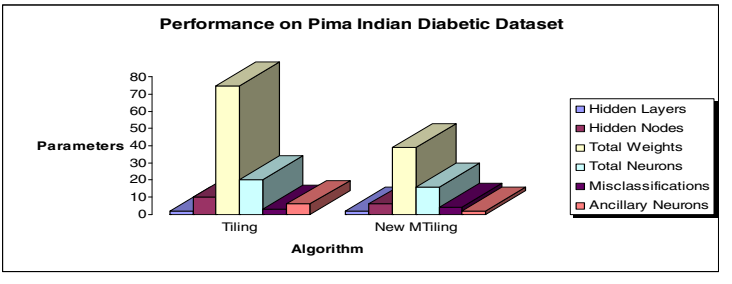


Fig. 2a. Performance on Pima Indian Diabetic Dataset

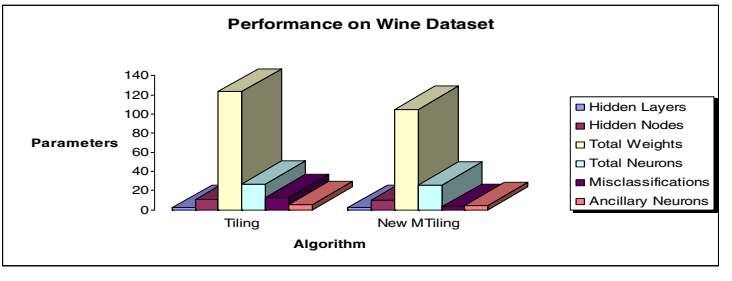


Fig. 2b. Performance on Wine dataset

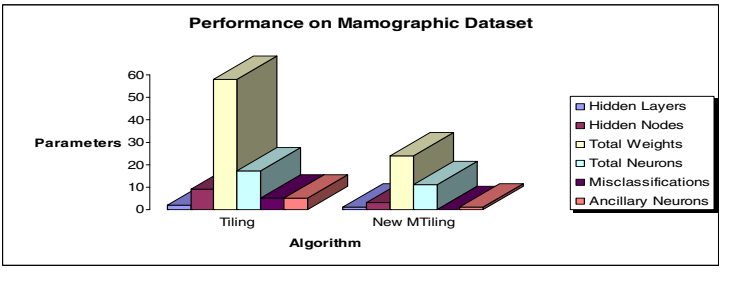


Fig. 2c. Performance on Mamographic dataset

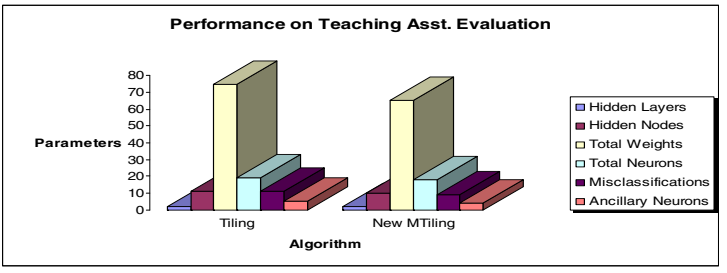


Fig. 2d. Performance on Teacher Evaluation dataset

5 Conclusion

Constructive Neural Networks offer an attractive framework for pattern classification problems. CoNN algorithms provide an optimal way to determine the architecture of a Multi Layer Perceptron network trainable with learning algorithms to determine appropriate weights. In this paper a novel Multi-Category Tiling architecture was designed with improved adaptive learning algorithm for reducing both the network size and misclassifications of patterns. The new MTiling architecture was tested on machine learning datasets namely Iris, Pima Indian Diabetic, Wine, Mamographic Mass and Teaching Assistant Evaluation. To analyze the performance of the new architecture, the datasets were also applied to existing Tiling architecture.

The results show that the new MTiling architecture outperforms the existing Tiling architecture in most of the parameters used for study. The improved adaptive learning strategy subsequently plays a significant role in improving the generalization capability, as well as the convergence rate of the network.

The New MTiling architecture along with improved adaptive learning strategy as proposed in this paper could be useful in many pattern classification tasks more effectively than the already existing MTilng architecture. The performance of the new architecture in terms of parameters like training time and improved generalization capability on various datasets as found in UCI repository is a challenging research direction.

References

[1] Haykin, S.: Neural Networks- A Comprehensive foundation. Pearson Education (2001)

[2] Bishop, C.A.: Neural Networks for Pattern Recognition. Oxford University Press (1995)

[3] Gomez, I., Franco, L., Jerez, J.M.: Neural Network architecture selection: can function complexity help? Kluwer Academy publishers (2009)

[4] Reed, R.: Pruning Algorithms – A survey. IEEE Transaction on Neural Networks 4(5) (1993)

[5] Thimm, G., Fiesler, E.: Pruning of Neural Networks, IDIAP Research Report (1997)

[6] Ash, T.: Dynamic node creation in back propagation networks. Connection Science 1, 365–375 (1989)
[7] Mezard, M., Nadal, J.P.: Learning in feed forward layered networks: the tiling algorithm. J. Phys. A: Math. Gen. 22, 2191–2203 (1989)
[8] Gallant, S.: Perceptron based learning algorithms. IEEE Transaction on Neural Networks 1, 179–191 (1990)
[9] Frean, M.: The Upstart Algorithm*: A* Method for Constructing and Training Feed forward Neural Networks. Neural Computation 4, 198–209 (1990)
[10] Parekh, R., Yang, J., Honavar, V.: MUpstart - A Constructive Neural Network Learning Algorithm for Multi-Category Pattern Classification. IEEE (1997)
[11] Fahlman, S., Leniere, C.: The Cascade correlation learning algorithm. In: Touretzhy (ed.) Information Systems. Margan-kauffman, Sanmateo (1990)
[12] Marchand, M., et al.: A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons. Europhys. Lett. 11, 487–492 (1990)
[13] Burgess, N.: Constructive Algorithm that converges for Real-valued input patterns. International Journal of Neural Systems 5(1), 59–66 (1994)
[14] Mascioli, F., Martinelli, G.: A constructive algorithm for binary neural networks: The Oil-Spot algorithm. IEEE Transactions on Neural Networks 6, 794–797 (1995)
[15] Yang, J., Parekh, R., Honavar, V.: DistAl: An inter-pattern distance Based constructive learning algorithm. Intelligent Data Analysis 3, 53–73 (1999)
[16] Franco, L., et al.: Early Breast cancer Prognosis prediction and Rule extraction using a New Constructive Neural Network algorithm. Springer, Heidelberg (2006)
[17] Franco, L., et al.: Extension of Generalization complexity measure to real valued input datasets. Springer, Heidelberg (2010)
[18] Frean, M.: A Thermal Perceptron learning rule. Neural Computation, 946–957 (1992)
[19] Poulard, H., Labreche, S.: A new threshold learning algorithm, Technical report 95504, LAAS (December 1995)
[20] Bertini, J.R., et al.: A Comparative Evaluation of Constructive Neural Networks Methods using PRM and BCP as TLU Training algorithms. In: IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan (2006)
[21] Parekh, R., Yang, J., Honavar, V.: Constructive Neural-Network learning algorithms for multicategory real-valued pattern classification, Dept. Computer. Science, Iowa State Univ., Tech. Rep. ISU-CS- TR97-06 (1997).
[22] Parekh, R., Yang, J., Honovar, V.: Constructive Neural Network Learning Algorithms for Pattern Classification. IEEE Transactions on Neural Networks 11(2), 436–451, (2000)
[23] Islam, M., et al.: A New Constructive algorithm for Architectural and Functional adaptation of Artificial Neural Networks. IEEE Transactions on Systems, Man and Cybernatics 39(6), 1590–1605 (2009)
[24] Micheli, A.: Neural Networks for Graphs: A contexual constructive approach. IEEE Transactions on Neural Networks 20(3) (2009)
[25] Perambur, N.S.: Information-Theoretic Aspects of Neural Networks. CRC Press (1999)
[26] Murphy, P., Aha, D.: Repository of machine learning databases. Dept. Inform. Comput. Sci., Univ., California, Irvine (1994), `http://www.ics.uci.edu/AI/ML/MLDBRepository.html`

A Novel Dual Band L-Slot Loaded Linearly Polarized Patch Antenna with Small Frequency-Ratio

Navneet Kumar[1], Monika[2], and N.S. Raghava[3]

[1] Department of ECE, Shree Ganpati Institute of Technology, GZB
nsitnavneet@gmail.com
[2] Department of ECE, Krishna Institute of Engg. & Technology, GZB
moniktronics@gmail.com
[3] Department of IT, Delhi Technological University, Delhi
nsraghava@dce.ac.in

Abstract. Microstrip Antennas are strongly used in wireless communication application. This paper proposed a novel patch antenna loaded with L-shaped slots and opposite corner of Square patch is trimmed to achieve dual band operation at 3.43 GHz and 4.75 GHz (the ratio of operating frequency is ≅1.38) with good matching (more than -20 dB) along with a single feed, radiate maximum power, VSWR, Return Loss etc. The antenna is designed based on extensive IE3D simulation studies. An antenna is simulated for the required performance.

Keywords: L-slot Loaded, Microstrip patch, Patch Antenna, Single Feed.

1 Introduction

A microstrip antenna consists of a metal patch radiator on an electrical grounded dielectric substrate. Microstrip patch antennas have many desirable features, including light weight, low cost and conformability, which make them suitable for a wide variety of applications at microwave frequency but it have a very low bandwidth characteristic. Dual-band operation is an important subject in microstrip antenna designs recently, several designs of the dual-band slot-loaded microstrip antennas have been reported [1, 2]. These dual-band designs are achieved by embedding a narrow L-shaped slot or an open-ring slot close to the boundary of the patch. However, the antennas designs proposed here have narrow impedance bandwidths of the two operating frequencies, of the order of 2% or less. In this paper, we present a novel dual-band L-slot loaded patch antenna printed on glass proxy material and the results of proposed antenna on its *S*11 characteristic along with the radiation patterns, VSWR, Axial ratio etc.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 435–440.
springerlink.com

2 Antenna Design and Analysis of Patch

2.1 Design

In this paper, parameters have been calculated using IE3D software version 14.65. For microstrip antennas, the width (W) and length (L) of the radiating patch and the effective permittivity of the microstrip structure ε_{re} which support the operation at the required resonant frequency (or the free-space wavelength λ_0) can be designed.

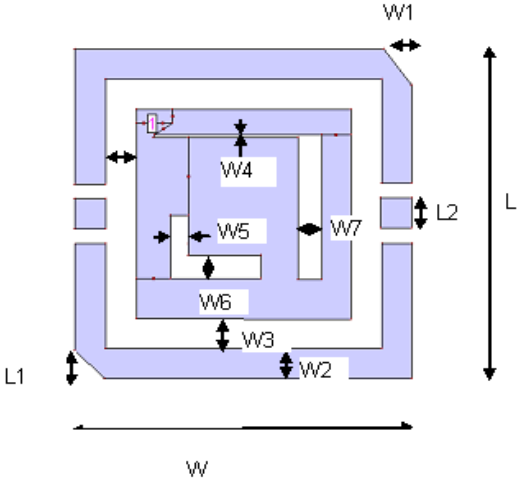


Fig. 1. The structure of the Ring and L slot loaded patch antenna 44x44mm

Proposed design was studied. Figure 1 shows the configuration of the proposed microstrip patch antenna. The selected substrate here is FR4, Glass Epoxy, (ε_r = 4.4, tanδ = 0.001) with thickness t = 1.6mm. the probe position is offset by 10mm x16 mm from the x – y axis .The dual L-slot are located as shown in figure-1 of microstrip patch antenna. By varying the length of one of the slot arms, the operating frequency- ratio of the two operating bands can be controlled. From simulation, it is found that the dimensions of the L-shaped slot significantly affect the performance of the antenna. The dimensions of the proposed patch antenna are given in table-1.

2.2 Analysis

Figure 2 shows the simulated return loss for the patch antenna shown in figure 1. The square radiating patch was of dimensions 44 mm x 44 mm (L x W). The distance “h” from top to ground plane of square patch was fixed to be 1.6mm.

Table 1. Dimensions of the design antenna

parameter	(mm)	parameter	(mm)
L	44	W3	4
W	44	W4	0.35
L1	4	W5	2.5
L2	4	W6	3
W1	4	Xf	10
W2	4	Yf	16

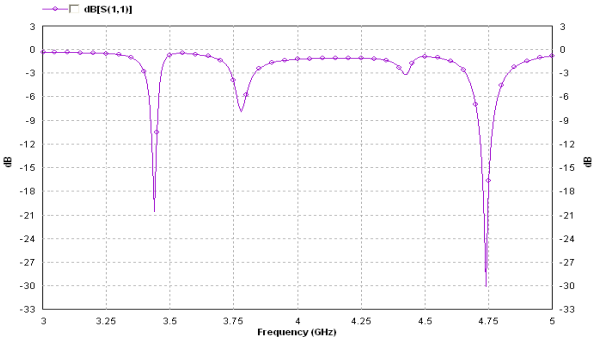


Fig. 2. Return loss of proposed patch antenna at 3.43GHz and 4.75GHz.

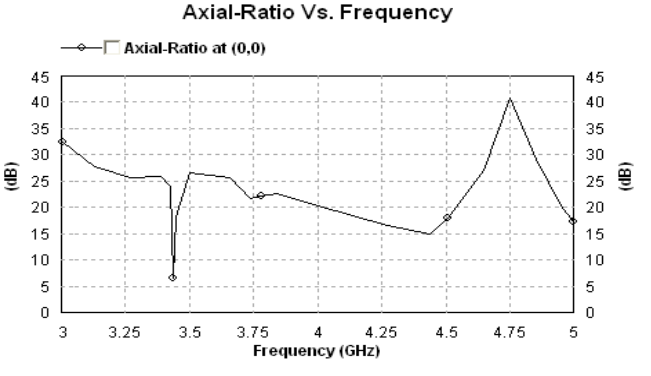


Fig. 3. Axial Ratio Vs Frequency of proposed patch antenna at 3.43GHz and 4.75GHz.

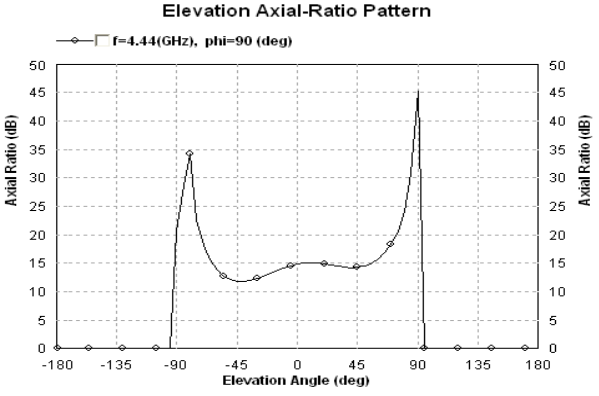


Fig. 4. Axial ratio of proposed patch antenna at $\phi=90^0$

From the result obtained in figure 2, it is seen that the patch antenna is radiating at a frequency of 3.43GHz up to 21 dB & 4.75 GHz up to 30dB. However it also generates harmonics say at 3.76GHz and 4.4GHz up to 7dB and 3dB respectively, which are unwanted.

The simulated axial ratio Vs frequency at $\varphi = 90^0$ in elevation plane are shown in figure 3 at frequency 3.4 GHz is near about 5dB.The simulated axial ratio at $\varphi = 90^0$ in elevation plane are also shown in Figure 4. Also radiation pattern in figure 5 shows that main lobe is up to 4.1dBi at frequency 3.4 GHz at $\phi=44.7^0$ and 3.8 dBi at 4.7GHz at $\phi=0^0$. Fig.6 shows the VSWR of the proposed antenna is 0.11 at the resonant frequency3.4 GHz and 0.11 at resonant frequency4.70 GHz.

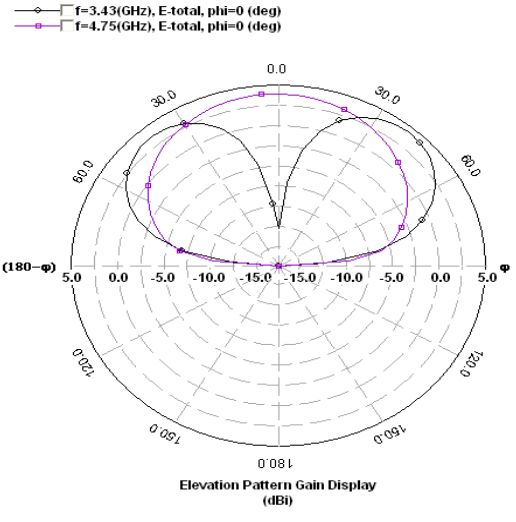


Fig. 5. Simulated radiation pattern of the of proposed patch antenna at frequency 3.43GHZ and 4.75 GHz at phase angle $\phi=00$.

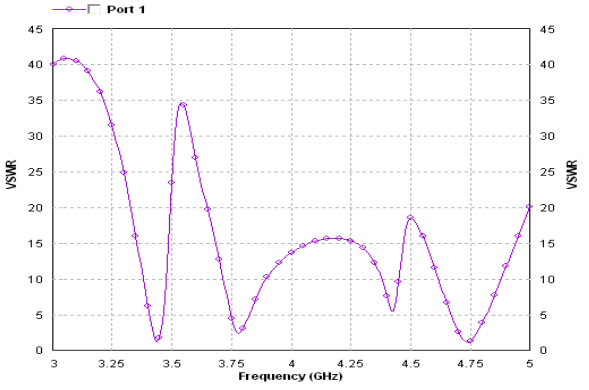


Fig. 6. VSWR of proposed patch antenna

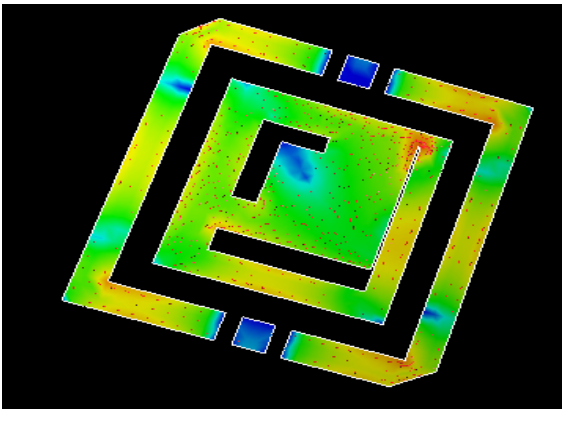


Fig. 7. Current distribution of proposed patch antenna at 3.4 GHz

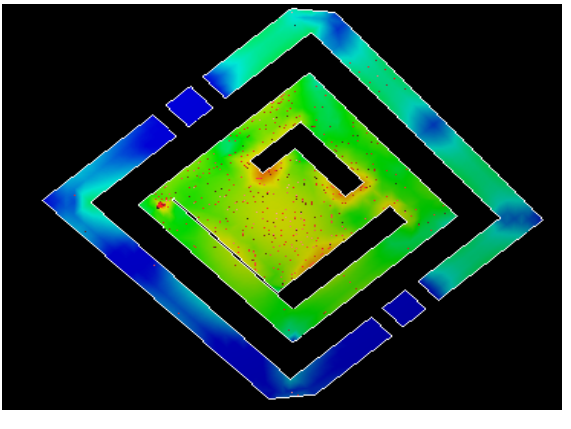


Fig. 8. Current distribution of proposed patch antenna at 4.75 GHz

3 Conclusion

A single-feed single-patch dual-band linearly polarized microstrip patch antenna with a small frequency-ratio has been investigated. The proposed antenna with an L-shaped slot has achieved good impedance matching, high gain. The antenna has been realized for a small dual-band frequency-ratio of 1.38. The proposed single-feed single-patch L-shaped slotted patch antenna is useful for small frequency-ratio, dual-band antenna and array designs.

Acknowledgments. The authors would like to thank authorities of Shree Ganpati Institute of Technology and Krishna Institute of Engineering and Technology, Ghaziabad for all the support provided.

References

[1] Svezhentsev, A.Y.: Some far field features of cylindrical microstrip antenna on an Electrically small cylinder. Progress In Electromagnetics Research B 7, 223–244 (2008)
[2] Abbaspour, M., Hassani, H.R.: Wideband Star-sharped microstrip patch antenna. Progress In Electromagnetics Research Letters 1, 61–68 (2008)
[3] Balanis, C.A.: Antenna theory, analysis and design, 3rd edn., pp. 843–865. John Wiley & Sons, Inc. (2005)
[4] Pozar, D.M.: Microstrip antennas. IEEE Proc. Antenna Propag. 80, 79–81 (1992)
[5] Garg, R., Bartia, P., Bhal, I., Ittipiboon, A.: Microstrip Antenna Design Hand Book. Artech House, Norwood (2001)
[6] Tariqul Islam, M., Shakib, M.N., Misran, N., Yatim, B.: Analysis of L-Probe Fed Slotted Microstrip Patch Antenna. In: Eleventh IEEE International Conference on Communication Systems (IEEE ICCS 2008), Guangzhou, China, November 19-21, pp. 380–383 (2008)

Group Social Learning in Artificial Bee Colony Optimization Algorithm

Harish Sharma[1], Abhishek Verma[2], and Jagdish Chand Bansal[3]

[1] ABV-Indian Institute of Information Technology and Management, Gwalior
harish0107@rediffmail.com
[2] ABV-Indian Institute of Information Technology and Management, Gwalior
abhishekverma.cs@gmail.com
[3] ABV-Indian Institute of Information Technology and Management, Gwalior
jcbansal@gmail.com

Abstract. Artificial Bee Colony (ABC) optimization algorithm is a powerful stochastic evolutionary algorithm that is used to find the global optimum solution in search space. In ABC each bee stores candidate solution; and stochastically modifies its candidate over time, based on the best solution found by neighboring bees,and based on the best solution found by the bee itself. When tested over various benchmark function and real life problems, it has performed better than a few evolutionary algorithms and other search heuristics . ABC, like other probabilistic optimization algorithms, has inherent drawback of premature convergence or stagnation that leads to loss of exploration and exploitation capability . Therefore, in order to balance between exploration and exploitation capability of ABC a new search strategy is proposed. In the proposed strategy, search process in ABC is performed by smaller group of independent swarms of bees. The experiments with 10 test functions of different complexities show that the proposed strategy has better diversity and faster convergence than the basic ABC.

Keywords: Numerical Optimization, Artificial Bee Colony Algorithm, Group Social Learning, Swarm intelligence, Meta-heuristics.

1 Introduction

Swarm Intelligence is a meta-heuristic approach in the field of nature inspired techniques that is used to solve optimization problems. It is based on the collective behavior of social creatures. Social creatures utilizes their ability of social learning to solve complex tasks. Researchers have analyzed such behaviors and designed algorithms that can be used to solve nonlinear, nonconvex or combinatorial optimization problems. Previous research [4, 12, 15, 17] have shown that algorithms based on Swarm Intelligence have great potential to

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 441–451.
springerlink.com

find a solution of real world optimization problem. The algorithms that have emerged in recent years include Ant Colony Optimization (ACO) [4], Particle Swarm Optimization (PSO) [12], Bacterial Foraging Optimization (BFO) [13], Artificial Bee Colony Optimization (ABC) [8] etc. Exploration and exploitation are the important mechanisms in a robust search process. While exploration process is related to the independent search for an optimal solution, exploitation uses existing knowledge to bias the search. In the recent years, there are few algorithms based on bee foraging behavior developed to improve both exploration and exploitation capability for solving the numerical optimization problems [5, 1, 6]. The Artificial Bee Colony (ABC) algorithm introduced by D. Karaboga [8] is one approach that has been used to find an optimal solution of numerical optimization problems. This algorithm is inspired by the behavior of honey bees when seeking a quality food source. Artificial Bee Colony (ABC) scheme is relatively a simple, fast and population based stochastic search technique.

A lot of developments, comparative studies and applications of ABC have been carried out in recent years. Performance of the ABC algorithm has been compared with Genetic Algorithm (GA) [7], Differential Evolution (DE) [16, 14, 15], Particle Swarm inspired Evolutionary Algorithm (PS-EA) [10], PSO and Evolutionary Algorithm (EA) [11]. ABC, DE and PSO algorithms were studied for measuring the effect of search space scaling in [2].

There are two fundamental processes which drive the swarm to update in ABC: the variation process, which enables exploring different areas of the search space, and the selection process, which ensures the exploitation of the previous experience. However, it has been shown that ABC may occasionally stop proceeding toward the global optimum even though the population has not converged to a local optimum [9]. Therefore, to maintain the proper balance between exploration and exploitation behavior of ABC a new search strategy is proposed in this paper. In the proposed methodology, the bees (onlooker and employed)are divided into smaller groups to search the food sources. Each group member exchanges the information within the group and the best information is shared among all the groups. In the proposed strategy, the groups size is kept very small comparative to the original swarm size. Therefore, the social learning process of the bees is limited to the group bees only. Hence, the proposed strategy is named Group Social Learning in Artificial Bee Colony optimization algorithm (GSLABC). Group Social Learning targeted to increase the algorithm's behavior to be more exploitive and diverse by increasing the number of local search in whole search space to basic ABC.

Rest of the paper is organized as follows: Section 2 describes brief overview of basic Artificial Bee Colony Algorithm. In section 3 The new search strategy is proposed. Experiments are presented and the results are discussed in section 4. Finally, in section 5, paper is concluded.

2 Brief Overview of Artificial Bee Colony Algorithm

Swarm based optimization algorithms find solution by collaborative trial and error. Peer to peer learning behavior of social colonies is the main driving force behind the development of many efficient swarm based optimization algorithms. Artificial Bee Colony (ABC)optimization algorithm is a recent addition in this category. Like any other population based optimization algorithm, ABC consists of a population of potential solutions. With reference to ABC, the potential solutions are food sources of honey bees. The fitness is determined in terms of the quality (nectar amount) of the food source. The total number of bees in the colony are divided into three groups: Onlooker Bees, Employed Bees and Scout Bees. Number of employed bees or onlooker bees are equal to the food sources. Employed bees are associated with food sources while onlooker bees are those bees that stay in the hive and use the information gathered from employed bees to decide the food source. Scout bee searches the new food sources randomly.

Similar to the other population-based algorithms, ABC is an iterative process. ABC process requires cycles of four phases: Initialization phase, Employed bees phase, Onlooker bees phase and Scout bee phase. Each of the phase is explained as follows:

2.1 Initialization of the Population

Initially, ABC generates a uniformly distributed initial population of SN solutions where each solution $x_i (i = 1, 2, ..., SN)$ is a D-dimensional vector. Here D is the number of variables in the optimization problem and x_i represent the i^{th} food source in the population. Each food source is generated as follows:.

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}) \tag{1}$$

where x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction and $rand[0, 1]$ is a uniformly distributed random number in the range [0, 1]

2.2 Employed Bee Phase

In employed bee phase, employed bees modify the current solution based on the information of individual experience and the fitness value of the new solution (nectar amount). If the fitness value of the new source is higher than that of the old source, the bee updates her position with the new one and discards the old one. The position update equation for i^{th} candidate in this phase is

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., D\}$ are randomly chosen indices. k must be different from i. ϕ_{ij} is a random number between [-1, 1].

2.3 Onlooker Bees Phase

After completion of the employed bees phase, the onlooker bees phase starts. In onlooker bees phase, all the employed bees share the new fitness information (nectar) of the new solutions (food sources) and their position information with the onlooker bees in the hive. Onlooker bees analyze the available information and select a solution with a probability, p_i, related to its fitness. The probability p_i may be calculated using following expression (there may be some other but must be a function of fitness):

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{3}$$

where fit_i is the fitness value of the solution i. As in the case of the employed bee, she produces a modification on the position in her memory and checks the fitness of the candidate source. If the fitness is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

2.4 Scout Bees Phase

If the position of a food source is not updated up to predetermined number of cycles, then the food source is assumed to be abandoned and scout bees phase starts. In this phase the bee associated with the abandoned food source becomes scout bee and the food source is replaced by a randomly chosen food source within the search space. In ABC, predetermined number of cycles is a crucial control parameter which is called *limit* for abandonment.

Assume that the abandoned source is x_i. The scout bee replaces this food source by a randomly chosen food source which is generated as follows

$$x_{ij} = x_{minj} + rand[0,1](x_{maxj} - x_{minj}), \text{ for } j \in \{1, 2, ..., D\} \tag{4}$$

where x_{minj} and x_{maxj} are bounds of x_i in j^{th} direction.

2.5 Main Steps of the ABC Algorithm

It is clear from the above discussion that there are three control parameters in ABC search process: The number of food sources SN (equal to number of onlooker or employed bees), the value of *limit* and the maximum number of cycles MCN.

In the ABC algorithm, the exploitation process is carried out by onlooker and employed bees and exploration process is carried out by scout bees in the search space. Following is the pseudo-code of the ABC algorithm [9]:

Artificial Bee Colony Algorithm
Initialize the population of solutions, $x_i (i = 1, 2, ...; SN)$ by using (1);
cycle = 1;
while cycle <> MCN **do**
Produce new solutions v_i for the employed bees using (2) and evaluate them;
Apply the greedy selection process for the employed bees;
Calculate the probability values p_i for the solutions x_i;
Produce the new solutions v_i for the onlookers from the solutions x_i selected depending on p_i using (3) and evaluate them;
Apply the greedy selection process for the onlookers;
Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution x_i using (4);
Memorize the best solution achieved so far;
cycle = cycle + 1;
end while

3 Group Social Learning Strategy

3.1 A Few Drawbacks of ABC

The inherent drawback with most of the population based stochastic algorithms is premature convergence. ABC is not an exception. Any population based algorithm is regarded as an efficient algorithm if it is fast in convergence and able to explore the maximum area of the search space. In other words, if a population based algorithm is capable of balancing between exploration and exploitation of the search space, then the algorithm is regarded an efficient algorithm. From this point of view, basic ABC is not an efficient algorithm [9]. Dervis Karaboga and Bahriye Akay [9] compared the different variants of ABC for global optimization and found that ABC shows a poor performance and remains inefficient in exploring the search space. The problems of premature convergence and stagnation is a matter of serious consideration for designing a comparatively efficient[1] ABC algorithm.

3.2 Motivation

In original ABC, all the bees are influenced from whole swarm therefore, there is a chance to get stuck in a local minima (stagnation). To set a trade-off between exploration and exploitation capability of ABC, the search capability of each bee (Onlooker and Employed) is guided by a set of bees (group). Therefore whole swarm size is divided into groups of smaller size in both the

[1] As it is not possible to design a fully efficient population based stochastic algorithm.

phases (Onlooker and Employed bees) of the algorithm. Now bees update their positions by learning within the group i.e. every bee now influence by the social behavior of its own group bees. This Phenomena improves local search capability i.e. exploitation in the search space. Next, each group search independently in the whole search space and share the local best information with each other, hence the exploration is achieved. So, it is clear from discussion that a good diversity and fast convergence can be achieved by the proposed strategy. To set the group size intensive experiments have been performed and found that the group size of 4 bees is best. Hence, by limiting the social learning capability in the search process, better exploration and exploitation can be achieved and so the proposed strategy is named "Group Social Learning in Artificial Bee Colony algorithm" ($GSLABC$).

4 Experimental Results and Discussion

4.1 Test Problems under Consideration

In order to see the effect of Group Social Learning strategy on ABC, 10 scalable (the number of decision variables may be varied as per user's choice) test problems of optimization are selected (listed in Table 1). These are continuous optimization problems and have different degrees of complexity and multimodality. For this study, number of decision variables is set to 50.

4.2 Experimental Setting

To test ABC and $GSLABC$ over test problems, following experimental setting is adopted:

- Colony size SN=100,
- $\phi_{ij} = rand[-1, 1]$,
- Number of food sources $SN/2$,
- $limit = 1500$,
- The stopping criteria is either maximum number of iterations (which is set to be 1000) is reached or the objective function value ≤ 0.0001,
- The number of simulations/run =100,
- The number of decision variables in scalable test problems $D = 50$.

4.3 Comparison of ABC with GSLABC

Numerical results with experimental setting of subsection 4.2 are given in Table 2. In Table 2, success rate (SR) (a simulation is said to be successful if the objective function value is ≤ 0.0001 in iterations up to 1000), mean objective function value ($MOFV$), average function evaluations (AFE) and

Table 1. Test problems

S. No.	Test Problem	Objective function	Search Space
1	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	[-5.12 5.12]
2	De Jong f4	$f(x) = \sum_{i=1}^{n} i.(x_i)^4$	[-5.12 5.12]
3	Griewank	$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$	[-600 600]
4	Rosenbrock	$f(x) = \sum_{i=1}^{n} (100(x_{i+1} - x^2)^2 + (x_i - 1)^2)$	[-30 30]
5	Rastrigin	$f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12 5.12]
6	Ackley	$f(x) = -20 + e + exp(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n} x_i{}^3}) - exp(\frac{1}{n}\sum_{i=1}^{n} \cos\ (2\pi.x_i)x_i)$	[-1 1]
7	Alpine	$f(x) = \sum_{i=1}^{n} \lvert x_i \sin\ x_i + 0.1x_i \rvert$	[-10 10]
8	Cigar	$f(x) = {x_0}^2 + 100000 \sum_{i=1}^{n} {x_i}^2$	[-10 10]
9	Schewel	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10 10]
10	Neumaier 3 Problem (NF3)	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	$[-n^2\ n^2]$

standard deviation (SD) are reported. Table 2 shows that most of the time inclusion of the proposed search strategy in ABC improves the reliability, efficiency and accuracy. Some more intensive statistical analyses based on performance index and boxplot have been carried out for results of basic ABC and $GSLABC$.

4.3.1 Statistical Analysis

Statistical comparisons have been carried out using boxplots and performance index [3]. For the purpose of comparison in terms of consolidated performance, boxplot analysis is carried out for all the considered algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [18]. The boxplot for average function evaluation of ABC and $GSLABC$ is shown in Figure 1. It is clear from this figure that $GSLABC$ is better than ABC as Interquartile Range and Median are low for this

Table 2. Comparison of the results of test problems for 30 Dimension

Test Problem	**Algorithm**	**MOFV**	**SD**	**AFE**	**SR**
Sphere	ABC	8.46E-05	1.51E-05	53934	100
	GSLABC	8.85E-05	1.33E-05	28869	100
De Jong	ABC	5.54E-05	3.09E-05	27640	100
	GSLABC	5.75E-05	2.91E-05	15131	100
Griewank	ABC	1.20E-03	3.21E-03	93795	54
	GSLABC	6.09E-03	1.10E-02	79073	61
Rosenbrock	ABC	2.31E+01	1.67E+01	100000	0
	GSLABC	1.40E+01	1.82E+01	100000	0
Rastrigin	ABC	4.48E+00	2.06E+00	100000	0
	GSLABC	9.30E-05	1.43E-04	70802	99
Ackley	ABC	6.80E-03	3.06E-03	100000	0
	GSLABC	1.02E-04	6.14E-05	92470	86
Alpine	ABC	3.65E-02	2.76E-02	100000	0
	GSLABC	5.06E-04	7.67E-04	99382	12
Cigar	ABC	9.05E-05	6.53E-05	90172	96
	GSLABC	8.37E-05	1.95E-05	81459	100
Schewel	ABC	9.41E-04	3.27E-04	100000	0
	GSLABC	9.37E-05	8.84E-06	80044	100
Neumaier 3	ABC	5.41E+03	5.98E+03	95459	29
	GSLABC	1.22E+03	2.68E+03	74371	68

strategy i.e. $LSABC$. In order to compare the consolidated performance of ABC with $GSLABC$, the value of a performance index PI is computed [3]. This index gives a weighted importance to the success rate, the mean objective function value as well as the average number of function evaluations. The PI is calculated by varying (i) success rate, (ii) function evaluation and (iii) Mean function value, for both the algorithms. The graphs corresponding to each of the cases (i), (ii) and (iii) are shown in Figures 2(a), 2(b), and 2(c) respectively. By analyzing the Figure 2, It is observed that for each case, PI of $GSLABC$ is significantly higher than ABC algorithm. Therefore, we can say that the effect of Group Social Learning strategy is significant on the performance of Artificial Bee Colony ALgorithm.

For analyzing the reliability of the proposed algorithm, Figure 3, shows a comparison of the success rates of all the benchmark functions which are used in experiments. It is clear from Figure 3 that most of the time high success

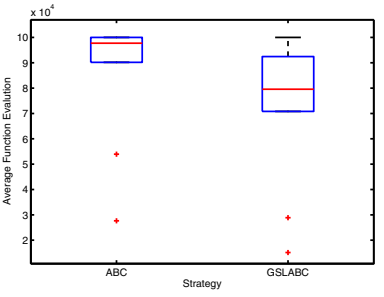


Fig. 1. Boxplot graph for Average Function Evaluation

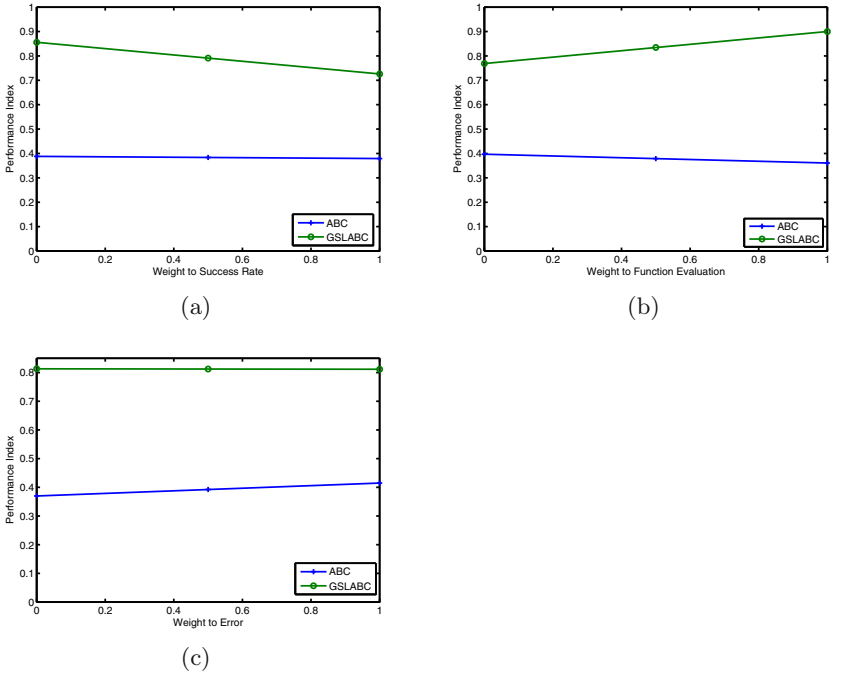


Fig. 2. Performance index; (a) for case (i), (b) for case (ii) and (c) for case (iii).

rate is achieved by $GSLABC$ as compare to the ABC. Therefore, it can be stated that $GSLABC$ is more reliable compared to the basic ABC.

Furthermore, Figure 4 shows a clear analysis of both the algorithm's efficiency for each of the benchmark optimization functions used in the experiments. It is obvious from Figure 4 that the average function evaluations of the proposed strategy are less than for all the test functions except function (4).

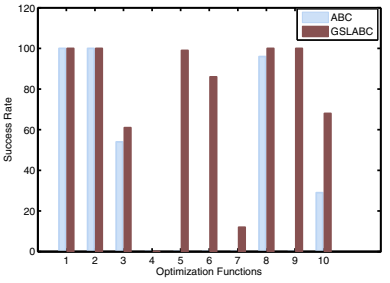


Fig. 3. Comparison of GSLABC and ABC based on success rate

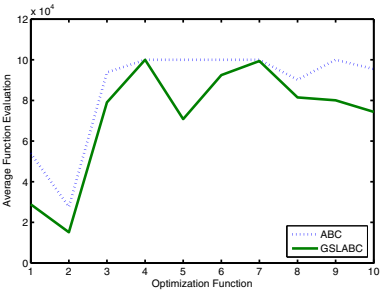


Fig. 4. Comparison of GSLABC and ABC based on Average Function Evaluation

Overall, by observing all the statistical tests and graphs, it may be concluded that *GSLABC* is better than the basic *ABC* algorithm.

5 Conclusion

In this paper, basic Artificial Bee Colony algorithm is improved by dividing the total population (employee and onlooker) in groups of smaller size. Numerical results indicate over benchmark functions that the reliability (due to success rate), efficiency (due to average number of function evaluations) and accuracy (due to mean objective function value) of GSLABC algorithm is outperformed the original ABC algorithm. The future work includes the studies on how to generalize the group size for different population size.

Acknowledgement. Harish Sharma and Abhishek Verma acknowledge ABV- Indian Institute of Information Technology and Management Gwalior, India for providing research grant to carry out this work.

References

[1] Banharnsakun, A., Achalakul, T., Sirinaovakul, B.: The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing (2010)

[2] Akay, B., Karaboga, D.: Effect of region scaling on the initialization of particle swarm optimization differential evolution and artificial bee colony algorithms on multimodal high dimensional problems. In: International Conference on Multivariate Statistical Modelling and High Dimensional Data Mining, Kayseri, Turkey, June 19-23, (2008)

[3] Thakur, M., Deep, K.: A new crossover operator for real coded genetic algorithms. Applied Mathematics and Computation 188(1), 895–911 (2007)

[4] Dorigo, M., Stützle, T.: Ant colony optimization. The MIT Press (2004)

[5] Haijun, D., Qingxian, F.: Bee colony algorithm for the function optimization. Science Paper Online (August 2008)

[6] Gao, W., Liu, S.: A modified artificial bee colony algorithm. Computers & Operations Research (2011)

[7] Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning (1989)

[8] Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06, Erciyes Univ. Press, Erciyes (2005)

[9] Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214(1), 108–132 (2009)

[10] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of Global Optimization 39(3), 459–471 (2007)

[11] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (abc) algorithm. Applied Soft Computing 8(1), 687–697 (2008)

[12] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, , vol. 4, pp. 1942–1948. IEEE (1995)

[13] Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine 22(3), 52–67 (2002)

[14] Price, K.V.: Differential evolution: a fast and simple numerical optimizer. In: Biennial Conference of the North American, Fuzzy Information Processing Society, NAFIPS 1996, pp. 524–527. IEEE (1996)

[15] Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)

[16] Storn, R., Price, K.: Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute-Publications-TR (1995)

[17] Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Congress on Evolutionary Computation, CEC 2004, vol. 2, pp. 1980–1987. IEEE (2004)

[18] Williamson, D.F., Parker, R.A., Kendrick, J.S.: The box plot: a simple visual method to interpret data. Annals of Internal Medicine 110(11), 916 (1989)

Multiple Objective Optimization of Reconfigurable Manufacturing System

Kapil Kumar Goyal[1], P.K. Jain[2], and Madhu Jain[3]

[1] Mechanical and Industrial Engineering Department, IIT Roorkee, Roorkee-247667, India
kapilacad@gmail.com
[2] Mechanical and Industrial Engineering Department, IIT Roorkee, Roorkee-247667, India
pjainfme@iitr.ernet.in
[3] Department of Mathematics, IIT Roorkee, Roorkee-247667, India
madhufma@iitr.ernet.in

Abstract. Reconfigurable Manufacturing System (RMS) is the state of art technology offering the functionality and capacity that is needed, when it is needed. The Reconfigurable Machine Tools (RMT) the sole of RMS, can perform variety of operations and can further be reconfigured to change its operational capabilities. In such environment a part can be processed by many feasible configurations of the RMS. In the present work a methodology is proposed for multiple objective optimization of RMS configuration based on convertibility, machine utilization and cost by applying nondominated sorting genetic algorithm II.

Keywords: Reconfigurable Manufacturing Systems (RMS), Reconfigurable Machine Tool (RMT), Machine selection, NSGA II.

1 Introduction

Modern challenges have paved the way for the new manufacturing paradigm known as Reconfigurable Manufacturing System (RMS) which can be reconfigured efficiently and economically to handle the ever changing product variety and demand [1, 2].

The reconfigurability is defined as ability of a system to adapt to expected or unexpected demand changes through the changes in the system or system component's structure guaranteeing the efficient use of functionalities [3]. Pattanaik *et al.* [4] have considered the cellular layout problem with reconfigurable machines to optimize the production. Maier-Speredelozzi *et al.* [5] presented the system convertibility as the capability of a system to adjust production functionality along with capacity and derived the system convertibility measures based on the assessment of convertibility of the system configuration, machines and material handling equipments. Gumasta *et al.* [6] have suggested reconfigurability of the RMS on system level based on various RMS characteristics. Son *et al.* [7] and Youssef and ElMaraghy [8] have generated a multi horizon demand configuration in the

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 453–460.
springerlink.com

RMS by first recording the k best configurations for all the demand periods based on cost as single performance criterion by applying genetic algorithm and later on selected the configurations based on configuration similarity and reconfiguration smoothness. Spicer and Carlo [9] have designed the multiple periods RMS configurations by considering the cost and reconfiguration. Goyal *et al.* [10] have modeled the system based on operational capability and reconfigurability.

From the literature on RMS, it can be concluded that RMS has not been modeled based on multiple objectives and moreover the important parameters like configuration convertibility and machine utilization have not been given due consideration in the available RMS approaches.

2 Performance Indices

The RMS allows quick changes in its configuration through the modular structure over the entire life cycle in response to changes in the product mix and volume. Thus in the present work configuration convertibility along with the machine utilization and cost are considered as multiple criteria for RMS single product flow line optimization. For modeling the RMS following notations are used:

Notations

mc_i^j	machine i ($1< i < I$) in its j^{th} ($1< j < J_i$) configuration
n_i^j	number of machines required to satisfy the demand when machine i with j^{th} configuration is selected
D	demand rate
FS_k	a set of feasible alternative machine configurations to perform k^{th} ($1< k < K$) operation $\{(i_1, j_1), (i_2, j_2),\ldots.(i_f, j_f)\ldots\ldots (i_{F_k}, j_{F_k})\}$. Here each feasible alternative f ($1< f < F_k$) is defined as (i_f, j_f), where i_f specifies the feasible machine and j_f specifies the feasible machine configuration
CM_i^j	cost of machine i with j^{th} configuration (i.e. mc_i^j)
$P_{i,k}^j$	production rate of machine i with j^{th} configuration for performing k^{th} operation
$\delta_{i,k}^j$	1 if operation k can be performed with machine i having its j^{th} configuration, otherwise 0
$C_{p,q}$	cost of assigning p^{th} machine with q^{th} configuration from the feasible alternative machine configurations to perform an operation at specified demand rate
$CC_{p,q}$	configuration convertibility of assigning p^{th} machine with q^{th} configuration from the feasible alternative machine configurations to perform an operation at specified demand rate
$MU_{p,q}$	machine utilization of assigning p^{th} machine with q^{th} configuration from the feasible alternative machine configurations to perform an operation at specified demand rate

2.1 Cost (C)

Cost is the important performance parameter driving the selection of machine configuration for a particular operation. Thus satisfying the customer demands economically is very crucial. The cost ($C_{p,q}$) of a feasible alternative machine configuration for performing k^{th} operation at specified demand rate is computed using

$$C_{p,q} = n_p^q \times CM_p^q \tag{1}$$

$$where \qquad n_p^q = \frac{D}{P_{p,k}^q} \tag{2}$$

The ratio $\frac{D}{P_{p,k}^q}$ is rounded off to the higher integer as number of machines can't be a fraction.

2.2 Configuration Convertibility (CC)

Convertibility is defined as the capability of a system to adjust production functionality, or changes from one product to another [11].

In the present study flow line configuration with crossover connections has been considered. The configuration convertibility is dependent upon the minimum increment of conversion (I), the routing connections (R) and the minimum number of replicated machines (X) [11]. The configuration convertibility (C_C) can be expressed as:

$$C_c^{'} = \frac{R \times X}{I} \tag{3}$$

The equation given below normalizes the value of C_C relative to a pure serial system with the same number of machines and so that the value falls in the range of 1 to 10 for all the system configurations:

$$CC_{p,q} = 1 + \frac{log_{10}\left[\frac{C_c^{'}}{C_{c,Serail}^{'}}\right]}{log_{10}\left[\frac{C_{c,\ K-Parallel}^{'}}{C_{c,\ K-Serial}^{'}}\right] \times \left(\frac{1}{9}\right)} \tag{4}$$

System convertibility includes contributions due to machines, their arrangements or configuration, and material handling devices. These factors are mapped together for an overall intrinsic assessment of system convertibility.

2.3 Machine Utilization (MU)

In the present scenario industries are facing a stiff global competition and volatile markets. In such circumstances utilization of the manufacturing system capacity is very crucial for sustenance and growth of the concern and underutilization of machine capacity which in turn affects the economic functioning may pose a serious threat to the survival of industry. Therefore the system should be utilized to the maximum possible extent by optimizing the system configuration. In the reconfigurable manufacturing environment the availability of multifunctional machines which can further be reconfigured into various configurations turns the machine selection problem into combinatorial problem. Therefore, while selecting the system configuration for a part the machine utilization should be given due consideration.

$$MU_{p,q} = \frac{P_{p,k}^{q} \times n_{p}^{q}}{D} \tag{5}$$

3 Optimal Machine Assignment for Single Part Flow Line Using NSGA II

Deb *et al.* [12] proposed NSGA-II, which is one of the most efficient and famous multi-objective evolutionary algorithms. The algorithm applies the fast nondominated sorting technique and a crowding distance to rank and select the population fronts. The terminologies central to the concept (non dominated sorting procedure, crowded distance estimation, crowded comparison operator) of NSGA-II, may be referred from Deb *et al.* [12]. The developed performance indices are applied to optimize a single part production flow line allowing paralleling of similar machines. As shown in Fig. 1, each operation is assigned to a stage according to the precedence constraints of the operation sequence and each stage is further assigned a machine type and its configuration number. The optimal assignment of the machine and machine configurations is realized by NSGA II taking cost, configuration convertibility and machine utilization as the objectives. For applying NSGA II in the present study, the set of feasible alternative machine configurations (FS_k) for all the operations are recorded by referring Table 1. Each element f of set FS_k is a combinations of two parameters i.e. the machine and the machine configuration. The total number of feasible alternative machine configurations to perform the k^{th} operation is F_k. The recording of feasible alternative machine configurations for all the operations is necessary for the constraint handling by applying the real coded chromosomes.

Table 1. RMT operation capabilities and cost

Operation (k) → mc_i^j ↓	RMT Production rate in parts/hour for performing various operations																				Cost of RMT (in 10^3 $)
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
mc_1^1	-	-	-	14	-	-	-	12	-	-	-	8	-	-	-	18	-	-	-	-	750
mc_1^2	-	-	-	-	15	-	-	-	20	-	-	-	-	-	-	-	-	16	-	-	955
mc_1^3	-	-	20	-	-	-	15	-	-	-	-	-	-	-	-	25	-	-	-	-	1025
mc_2^1	14	-	-	-	-	15	-	-	-	-	-	12	-	-	-	-	-	-	-	20	1215
mc_2^2	-	15	-	-	-	-	-	-	-	-	-	-	14	-	15	-	-	-	-	-	910
mc_2^3	-	-	25	-	-	-	-	18	-	-	25	-	-	-	-	-	20	-	-	-	1140
mc_2^4	-	20	-	-	20	-	18	-	-	-	-	-	-	24	-	-	-	-	-	-	1350
mc_3^1	-	12	-	-	-	-	-	-	15	-	-	10	-	-	-	-	10	-	-	-	780
mc_3^2	30	-	-	26	-	-	-	24	-	-	24	-	-	-	20	-	35	-	15	-	1825
mc_4^1	-	-	-	-	-	25	-	-	-	30	-	-	-	-	-	-	-	25	-	-	1350
mc_4^2	25	-	-	-	-	-	-	-	-	-	-	22	-	-	-	-	30	-	-	26	1500
mc_4^3	-	18	-	25	-	-	-	16	-	-	-	-	22	-	-	28	-	-	20	-	1400
mc_5^1	16	-	-	-	-	-	15	-	-	-	15	-	-	18	-	-	-	18	-	-	900
mc_5^2	-	-	24	-	20	-	-	-	-	25	-	-	-	-	-	-	24	-	-	20	1175
mc_5^3	-	-	-	24	-	-	-	-	30	-	-	-	-	-	18	-	-	-	-	-	1230
mc_5^4	20	-	-	-	-	22	14	-	-	-	-	-	-	20	-	16	-	-	18	-	1175

4 Objective Function, Constraint Handling and Solution Mapping

The present study proposes the assignment of machines to all the operations allocated on production stages from the feasible alternative machine configurations based on the objective function:

$$\text{Minimize } F1 = \sum_{s=1}^{S} C_{p_s, q_s} \tag{6}$$

$$\text{Maximize } F2 = \sum_{s=1}^{S} MU_{p_s, q_s} \tag{7}$$

$$\text{Maximize } F3 = \sum_{s=1}^{S} CC_{p_s, q_s} \tag{8}$$

Here Eq. (6) to (8) represent the objectives cost, machine utilization and configuration convertibility respectively. The subscripts p_s and q_s represent the feasible alternative machine p with its configuration q assigned at the s^{th} stage. The real-encoding of chromosome is used to overcome the difficulties related to binary encoding of continuous parameter optimization problems [13]. The real coded chromosomes along with the NSGA II are employed in the present study for obtaining the nondominated solutions representing optimal machine assignment. But in the present case feasible solutions are rather sparse which will lead to infeasible population. Further the crossover and mutation of the chromosomes will give rise to the infeasibility. Therefore a real coded chromosome is proposed. The length of

chromosome is equal to the number of stages and on each stage an operation has to be performed according to the operation sequence. A set of feasible alternative machine configurations FS_k for each operation is already recorded as shown in Fig. 1. As shown in Fig. 2, now each stage is to be assigned with a feasible machine configuration which is mapped in the present study through the real coded chromosomes.

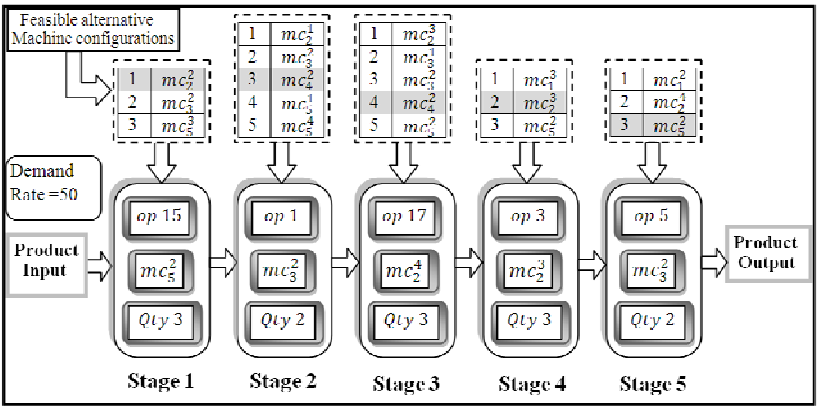


Fig. 1. RMS Flow line feasible machine configurations mapping

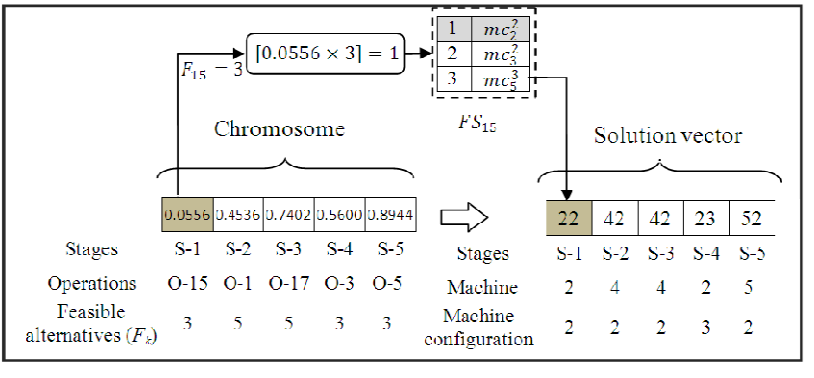


Fig. 2. Real coded chromosome solution mapping

5 Numerical Illustration

For illustrating the developed approach of optimal machine configuration assignment, a set of RMTs having the operational capabilities and cost as given in Table 1 is considered. The optimal machine assignment for a single part flow line allowing paralleling of similar machines for a single fixed demand period is illustrated. As shown in Fig. 1, the operation sequence of the product to be produced is

assumed to be 15→1→17→3→8 with a demand rate of 50. The number of stages is also assumed to be five. The nondominated solutions obtained by applying the NSGA II are compiled in Table 2.

The various parameters used in NSGA II are Population size 50, Number of generations 100, binary tournament selection operator, crossover probability 0.8 and mutation probability as 0.2. As the present problem lies in the domain of discrete discontinuous and non convex search space, thus predicting the exact number of pareto frontiers is not possible. In the present illustration 47 nondominated solutions were obtained, but due to space constraint only two best solutions for each objective have been listed in Table 2.

Table 2. Non-dominated solutions for the flow line

S. No.	Solutions (Stage wise) Machine assigned/No. of machines					Cost	Machine Util.	Config. Conv.	Solution No.
1	22/4	42/2	42/2	23/2	52/3	15445	0.9000	4.7210	26
2	53/3	42/2	42/2	23/2	52/3	15495	0.9185	4.6460	17
3	53/3	42/2	31/5	23/2	12/4	16690	0.9518	4.9869	5
4	22/4	42/2	31/5	23/2	12/4	16640	0.9333	5.0138	15
5	22/4	21/4	31/5	13/3	12/4	19295	0.8785	6.0223	11
6	22/4	21/4	31/5	13/3	52/3	19000	0.8785	6.0198	23

6 Conclusion and Future Scope

In the present study a novel approach has been proposed for the machine selection based on machine utilization, configuration convertibility along with the cost. The multiple objective problem of machine assignment is attempted through NSGA II. The real coded chromosomes helped in handling the sparse population of feasible solutions along with facilitating the crossover and mutation. Considering the ranking of the solutions the decision manager may choose a suitable candidate among the top ranking solutions to justify the objectives defined by the management along with the present market scenario. In future authors plan to study RMS for the multiple periods planning horizon.

References

[1] Koren, Y., Hiesel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van, B.H.: Reconfigurable manufacturing systems. Annals of the CIRP 48(2), 527–540 (1999)

[2] Mehrabi, M.G., Ulsoy, A.G., Koren, Y.: Reconfigurable manufacturing systems: Key to future manufacturing. Journal of Intelligent Manufacturing 11(4), 403–419 (2000)

[3] Urbani, A., Molinari-Tosatti, L., Pedrazzoli, P., Fassi, I., Boer, C.R.: Flexibility and reconfigurability: An analytical approach and some examples. In: CIRP 1st International Conference on Reconfigurable Manufacturing Systems, Ann Arbor, MI (2001)

[4] Pattanaik, L.N., Jain, P.K., Mehta, N.K.: Cell formation in the presence of reconfigurable machines. International Journal of Advanced Manufacturing Technology 34, 335–345 (2007)
[5] Maier-Speredelozzi, V., Koren, Y., Hu, S.J.: Convertibility measures for manufacturing systems. Annals of the CIRP 52, 367–370 (2003)
[6] Gumasta, K., Gupta, S.K., Benyoucef, L., Tiwari, M.K.: Developing a reconfigurability index using multi-attribute utility theory. International Journal of Production Research 49(6), 1669–1683 (2011)
[7] Son, S.Y., Olsen, T.L., Derek, Y.: An approach to scalability and line balancing for RMS. Integrated Manufacturing Systems 12(7), 500–511 (2001)
[8] Youssef, A.M.A., ElMaraghy, H.A.: Optimal configuration selection for Reconfigurable Manufacturing Systems. International Journal of Flexible Manufacturing Systems 19, 67–106 (2007)
[9] Spicer, P., Carlo, H.J.: Integrating reconfiguration cost into the design of multiperiod scalable reconfigurable manufacturing systems. Transactions of the ASME 129, 202–210 (2007)
[10] Goyal, K.K., Jain, P.K., Jain, M.: Optimal configuration selection in reconfigurable manufacturing system using NSGA II and TOPSIS. International Journal of Production Research (i first) (2011)
[11] Koren, Y., Katz, R.: Reconfigurable Apparatus for Inspection during a Manufacturing Process. US patent # 6,567,162 (2003) (Issue date: May 20, 2003)
[12] Deb, K., Samir, A., Amrit, P., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 103–112 (2002)
[13] Wright, A.H.: Genetic algorithms for real parameter optimization. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms I, pp. 205–218. Morgan Kaufmann, San Mateo (1991)

Design of Boolean Functions Satisfying Multiple Criteria by NSGA-II

Rajni Goyal, Shiv Prasad Yadav, and Amar Kishor

Indian Institute of Technology, Roorkee
rajni.aggarwal@yahoo.co.in, yadavfma@iitr.ernet.in,
amarsdma@gmail.com

Abstract. In this work we focus on non-linearity and resiliency related criteria and explore a multi-objective evolutionary approach aiming to find balanced Boolean functions satisfying these criteria. We have shown that the multi-objective approach is better to apply than single objective optimization approaches. We have applied NSGA-II to find desired Boolean functions.

Keywords: Crowding Distance, Pareto-Optimal Fronts, Rank.

1 Introduction

Many desirable multiple criteria of Boolean functions have been identified. Among them high nonlinearity, balancedness, high algebraic degree and low autocorrelation are important. It is well known that at the same time a function cannot be balanced and have maximum algebraic degree, maximum distance to linear functions and maximum distance to linear structures. Recently, the trade-offs between some of these criteria have received a lot of attention[6]. Some works have been sought to combine algebraic construction with deterministic computer search methods[6, 10, 11] and some authors have attempted using search heuristics such as genetic algorithms, hill climbers, and simulated annealing to generate Boolean functions[2, 7]. A clear trade-off has been shown for correlation immunity, algebraic degree and nonlinearity[1]. However, generating Boolean functions purely by constructive algebraic methods becomes increasingly difficult as the number of criteria to be satisfied is augmented. The more criteria to be taken into account, the more complected is to generate Boolean functions satisfying those property purely by constructive algebraic means. Many of the best functions of small numbers of variables have been obtained by single objective two stage optimization method[1]. We show that the multi-objective optimization technique(NSGA-II) approach is an efficient alternative to single objective optimization approaches presented so far. By using multi objective optimization, we can avoid weighting constant. We have constructed a high nonlinear, balanced, having high algebraic degree and resilient Boolean function by NSGA-II.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 461–468.
springerlink.com

Paper is organized as follows: In section 2, some definitions from literature are given. Section 3 gives a brief description of NSGA-II. Section 4 discusses the multi-objective optimization problem taken by us. Section 5 gives results and conclusion of our paper.

2 Some Definitions

2.1 Boolean Function

Let $\mathbf{F}_2^n$ be the prime field of characteristic 2, then $\mathbf{F}_2^n$ is an n-dimensional vector space over $\mathbf{F}_2$. An element of $\mathbf{F}_2^n$ can be represent by a binary vector of length n. Any function $f : \mathbf{F}_2^n \to \mathbf{F}_2$ is called a Boolean function on n-variables. The set of all Boolean functions of n-variables is denoted by $\mathscr{B}_n$. The Hamming weight of a binary vector $x = (x_1, \ldots, x_n) \in \mathbf{F}_2^n$ is denoted by $w_H(x)$ and is the size of its support $\{i \in N | x_i \neq 0\}$ where N=$\{1, 2, \ldots, n\}$. The Hamming weight $w_H(f)$ of a Boolean function f on F_2^n is also defined to be the size of its support $\{x \in \mathbf{F}_2^n | f(x) \neq 0\}$.

2.2 Balancednes

A function is said to be balanced if the number of 0's is the same as the number of 1's in output table. And this property is called balanced-ness.

2.3 Hamming Distance

The Hamming distance between two functions f and g, denoted by hd(f,g), is defined as the number of truth table positions in which the functions f and g disagree, i.e.,

$$hd(f,g) = |\{x | f(x) \neq g(x)\}|,$$

where $|.|$ stands for the cardinality of the set.

2.4 Walsh Hadamard Transform

The Walsh Hadamard Transform (WHT) provides another means of representing a Boolean function. The WHT expresses a Boolean function in term of its cross correlation to all the linear functions. Thus, for a given linear function L_λ specified by $\lambda \in \mathbf{F}_2^n$.

The WHT of a function f denoted by $W_f(\lambda)$ and is defined by

$$W_f(\lambda) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda . x},$$

where $\lambda \in \mathbf{F}_2^n$.

2.5 Bent Function

A Boolean function from $\mathbf{F}_2^n$ to $\mathbf{F}_2$ is called bent if $W_f(\lambda)$= c 2^{n-1}, where $c = 1$ or -1. In case n is even, the highest possible non-linearity is $(2^{n-1} - 2^{(n/2)-1})$. The Boolean function achieving this bound is called bent function.

2.6 Autocorrelation

The autocorrelation of a function gives an indication of the imbalance of all first order derivatives of a Boolean function and provides a measure of self similarity for Boolean function. The derivative of Boolean function f(x), taken with respect to a vector s, where x and $s \in \mathbf{F}_2^n$, is defined as $f(x) \bigoplus f(x+s)$. Similarly the derivative in polar form can be defined as $\widehat{f}(x)\widehat{f}(x+s)$. So the autocorrelation of a function f is denoted by $r_f(s)$ and is defined by

$$r_f(s) = \sum_{x \in \mathbf{F}_2^n} \hat{f}(x)\hat{f}(x+s),$$

where $\hat{f}(x) = (-1)^{f(x)}$.

A Boolean function f is considered to be good if r_f is small .

3 Non-dominated Sorting Genetic Algorithm (NSGA-II)

NSGA-II, doveloped by Srinivas and Deb[4] is a generational Multi-objective Optimization Evolutionary Algorithm(MOEA) that aims at approximating the Pareto optimal fronts for a given problem, while keeping high diversity in its result set.

It works on three main modules:

1. Non-dominated Sorting 2. Crowding distance assignment 3. Crowded comparison operator.

3.1 Non-dominated Sorting

At a certain generation t, it partitions the population P_t in front F_i with index i indicating the non-domination rank shared by all solutions in such a front. The first front F_1 is the actual non-dominated front, i.e., it consists of all non-dominated solutions in population P_t at a certain generation t. The second front F_2 consists of all solutions that are non-dominated in the set $|P_t| - |F_1|$, i.e., each member of F_2 is dominated by at least one member of F_1 as shown in Fig. 1. Generally, front F_k comprises all solutions that are non-dominated if the individuals in fronts F_j with $j < k$ were to be removed from P_t.

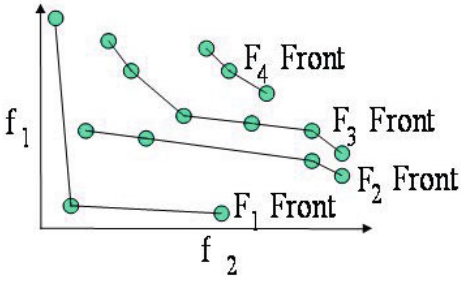


Fig. 1. Nondominated Sorting of a Population

3.2 Crowding Distance Assignment

To get an estimate of the density d_i of solutions surrounding a particular solution i in the population, we take the average distance of two solutions on either side of the solution i along each objective. This quantity d_i serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as vertices. It is called the crowding distance. The crowding distance assignment calculates a crowding distance value for each individual within a certain front F_i as the difference between objective function values in the nearest neighbors on each side of the individual, then summed up over all objectives. The crowding distance values for extreme solutions (i.e., solutions with the smallest and largest function values occurring within the front) are assigned an infinite value, which is motivated by the pursuit of diversity and which effectively preserves them into the next generation. The front in which they are contained should be partially discarded when a new population P_{t+1} is formed (Fig. 2).

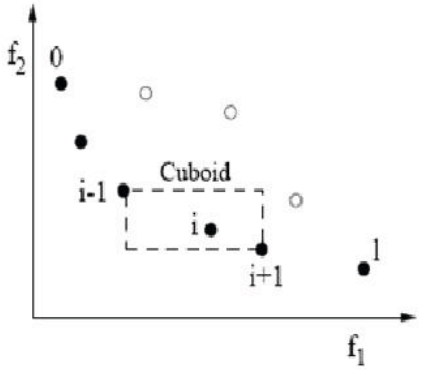


Fig. 2. Description of Crowding Distance

3.3 Crowded Comparison Operator $\prec_n$

It guides the selection process by defining an ordering on P_t. Each solution in the population has two attributes

1. non-domination rank(i_r);
2. crowding distance(i_d).
   The crowded comparison operator $\prec_n$ can be defined as
   i $\prec_n$ j if $i_r \leq j_r$ and $i_d > j_d$

Between two solutions with different non-domination ranks, solution having the lower rank is preferred. If both solutions belong to the same front(having the same rank), we prefer the solution that is located in the lesser crowded region (i.e., with higher crowding distance value).

The crowding comparison operator guides the selection process at the various stages of algorithm toward a uniformly spread-out Parieto-optimal fronts.

Algorithmically, NSGA-II is explained below:

step 1. Combine parent and offspring populations P_t and Q_t and create $R_t = P_t \bigcup Q_t$. Perform a non-dominated sorting to R_t and identify different fronts: F_i, $i = 1, 2, ..., etc.$

step 2. Set new population $P_{t+1} = \phi$. Set a counter $i = 1$. Until $|P_{t+1}| + |F_i| < N$, perform $P_{t+1} = P_{t+1} \bigcup F_i$ and set $i = i + 1$.

step 3. Perform the Crowding sort procedure and include the widely spread $N - |P_{t+1}|$ solution by using the crowding distance values in the sorted F_i to P_{t+1}.

step 4. Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection, crossover and mutation operators.

Constraint Handling in NSGA-II. Consider a constrained MOOP. The constraints divide the search space into two regions - feasible and infeasible regions. In MOOP, all Parieto-optimal solutions must be feasible solutions. The constrained problems in NSGA-II can be handled on the basis of dominance.

A solution i is said to constrained-dominate a solution j if any of the following conditions is true:

1. Solution i is feasible and solution j is not.
2. Both solutions i and j are infeasible, but solution i has a smaller overall constrained violation.
3. Both solutions i and j are feasible and solution i dominates solution j. When comparing two feasible solutions, the solution which dominates the other solution is considered a better solution. On the other hand, if both solutions are infeasible, the solution having a lesser number of constrained violations is a better solution.

4 Multiobjective Optimization

Here, we describe the evaluation criteria used in this work to evolve functions. The first evaluation criterion used is the non-linearity *nl* and the second is resiliency.

We know that the bent functions have the maximum nonlinearity and for bent functions, $|W_f(\lambda)| = 2^{n/2}$ for all $\lambda \in \mathbb{F}_2^n$. For balanced Boolean functions this ideal bound cannot be achieved but it does suggest that a objective function that seeks to minimize the spread of Walsh Hadamard values is well motivated. So we use first objective function

$$f_1 = \sum ||W_f(\lambda)| - 2^{n/2}|^R,$$

where R ia a constant, we can take any value between $[3, 10]$.

If a function has resiliency m, then $W_f(\lambda)$ will be zero all those $\lambda \in \mathbb{F}_2^n$ for which $w_H(\lambda) \leq m$. Hence, we take

$$f_1 = \sum ||W_f(\lambda)| - 2^{n/2}|^R,$$

for $w_H(\lambda) > m$.

Let us take the second objective f_2 as

$$f_2 = \sum_{\lambda} |W_f(\lambda)|,$$

for $w_H(\lambda) \leq m$.

Now,

$$f_1 = \sum_{\lambda} ||W_f(\lambda)| - 2^{n/2}|^R$$

for $w(\lambda) \geq m$,

(where $W_f(\lambda) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x}$ and n is the number of variables.)

So

$$= \sum_{\lambda} || \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x}| - 2^{n/2}|^R$$

$$= \sum_{\lambda} || \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}| - 2^{n/2}|^R$$

$$= \sum_{\lambda} || \sum_{x \in \mathbb{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}| - 2^{n/2}|^R$$

$$= \sum_{\lambda} ||M_\lambda - \sum_{x \in \mathbb{F}_2^n} 2f(x).a_{\lambda.x}| - 2^{n/2}|^R,$$

(where $M_\lambda = \sum_{x \in \mathbb{F}_2^n} a_{\lambda.x}$ and $a_{\lambda.x} = (-1)^{\lambda.x}$.)

$f(x)'s$ are binary variables(Boolean functions that have to be found.) We have taken R = 3 for all values of m.

Similarly,

$$f_2 = \sum_{\lambda} |W_f(\lambda)|$$

for $w_H(\lambda) \leq m$.

So, our problem as an MOOP is given below:

$$minimize F = (f_1, f_2)$$

subject to the constraints

$$|\sum_{x \in \mathbb{F}_2^n} f(x) - (2^n)/2| \leq 0,$$
$$|(2^n)/2 - \sum_{x \in \mathbb{F}_2^n} f(x)| \leq 0.$$

($\sum_{x \in \mathbb{F}_2^n} f(x)$ should be equal to $(2^n)/2$ for balanced function).

after applying NSGA-II to the above functions and constrained and got balanced Boolean functions with low autocorrelation for resiliency 1 and 2 for 4 and 5 variables.

5 Results and Discussion

After applying NSGA-II, we got Boolean functions having the best resiliency, nonlinearity, and algebraic degree for n = 4 and 5. These functions are given in Table 1. A Boolean function is denoted by (n, m, d, *nl*), where n is the number of variables, m is resiliency, d is algebraic degree, and *nl* is the nonlinearity.

Table 1. Results obtained by NSGA-II

(4,1,3,12)	0 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0
(5,1,3,12)	1 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 1
(5,2,3,12)	1 0 1 0 0 0 1 1 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1

6 Conclusion

We got the same best profile functions, as reported in literature [3, 5] by the technique NSGA-II. By using multi-objective optimization(NSGA-II), we got the Boolean functions having the same profiles as by single objective two stage optimization [3]. We have shown that the multi-objective approach is an efficient alternative to single objective optimization approaches presented so far.

References

1. Clark, J.A., Jacob, J.L., Stepney, S., Maitra, S., Millan, W.L.: Evolving Boolean Functions Satisfying Multiple Criteria. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 246–259. Springer, Heidelberg (2002)
2. Clark, J.A., Jacob, J.L.: Two-Sttage Optimization in the Design of Boolean Functions. In: Clark, A., Boyd, C., Dawson, E.P. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 242–254. Springer, Heidelberg (2000)
3. Clark, J.A., Jacob, J.L., Stepney, S., Maitra, S., Millan, W.L.: Evolving Boolean Functions Satisfying Multiple Criteria. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 246–259. Springer, Heidelberg (2002)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm. IEEE Transection on Evolutionary Computation 6(2) (April 2002)
5. Aguirre, H., Okazaki, H., Fuwa, Y.: An Evolutionary Multiobjective Approach to Design Highly Non-linear Boolean Functions. In: GECCO 2007 (2007)
6. Maitra, S., Pasalic, E.: Further Constructions of Resilient Boolean Functions with Very High Nonlinearity. IEEE Transactions on Information Theory 48(7), 1825–1834 (2002)
7. Millan, W., Clark, A., Dawson, E.: An Effective Genetic Algorithm for Finding Highly Non-linear Boolean Functions. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 149–158. Springer, Heidelberg (1997)
8. Millan, W., Clark, A., Dawson, E.: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 489–499. Springer, Heidelberg (1998)
9. Millan, W., Clark, A., Dawson, E.: Boolean Function Design Using Hill Climbing Methods. In: Pieprzyk, J.P., Safavi-Naini, R., Seberry, J. (eds.) ACISP 1999. LNCS, vol. 1587, pp. 1–11. Springer, Heidelberg (1999)
10. Pasalic, E., Maitra, S., Johansson, T., Sarkar, P.: New Constructions of Resilient and Correlation Immune Boolean Functions Achieving Upper Bound on Nonlinearity. In: Proc. Workshop on Coding and Cryptography - WCC 2001, Electronic Notes in Discrete Mathematics, vol. 6. Elsevier Science (2001)
11. Sarkar, P., Maitra, S.: Nonlinearity Bounds and Constructions of Resilient Boolean Functions. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 515–532. Springer, Heidelberg (2000)
12. Srinivas, N., Kalyanmoy, D.: Multiobjective optimization Using Nondominating Sorting Genetic Algorithms. Massachusetts Institute of Technology Evolutionary Computation 2(3), 221–248

Nature-Inspired Fault Tolerant Area Monitoring in Sensor Network

Rameshwar Nath Tripathi[1], Shekhar Verma[1], and S.V. Rao[2]

[1] Department of Information Technology, Indian Institute of Information Technology, Allahabad, 211012, UP, India
{rntripathi,sverma}@iiita.ac.in
[2] Department of Computer Science, Indian Institute of Technology - Guwahati, Guwahati 781039, Assam, India
svrao@iitg.ernet.in

Abstract. Effective network coverage and operational life span is key concern of randomly deployed Wireless Sensor Network (WSN) for performing monitoring function in designated region. Intrinsically, WSN consists of resource constraints sensor nodes. For effective coverage, it is undesirable activating all deployed set of nodes for getting the desired degree of coverage if the same result can be obtained by activating a small subset of deployed nodes for providing the sensing function in the concerned region. We study the problem of extending the life span of the sensor network for fault tolerant area coverage. The proposed genetic algorithm based approach aims to cover a sensed area with minimum number of active nodes and compute the maximum number of Sensing Cover Set (SCS) so that network life time can be prolonged by calculating the working schedule of cover set. Each SCS has been assigned the cover set ID. Each SCS works for the specified amount of time in an alternate fashion. Simulation experiment indicates that genetic algorithm based approach is able to optimally partition the nodes into different SCS.

Keywords: Coverage problem, Sensing cover set, Genetic algorithm, Fault tolerant.

1 Introduction

Wireless sensor nodes are equipped with limited resources which are power, communication and computation capabilities. Utilization of these resources in optimize fashion are key concerns in design of a sensor network. It is infeasible to provide continuous supply of these resources in all possible circumstances such as sensor nodes located in harsh terrain. So, the replacement of depleted battery is not viable option in every scenario. Hence, the resource optimization is one of the important objectives in designing of WSN protocols.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 469–480.
springerlink.com

A WSN consists of large numbers of autonomous sensor nodes which are geographically scattered in the Euclidian plane. These sensor nodes communicate among themselves using broadcast. These distributed sensor nodes cooperate, coordinate and collaborate to monitor the spatial temporal phenomena in physical world. Individually, the sensor nodes which are the member of the networks know little about the full task to be accomplished [1]. Once all the nodes work together, the task is accomplished. The principle aim of WSN is to sense ambient data from environment and send the report of the sensed data to base station for further processing. Effective and efficient monitoring needs good coverage of whole network area.

An important problem is to identify fundamental concepts of coverage that apply to a class of applications. The concept of coverage can be classified in three categories i.e. area coverage, target coverage [2] and barrier coverage. A wireless sensor network is said to provide area coverage in a region if every point in this region is within the monitoring range of some sensor. The network is said to provide k-area coverage, or simply k-coverage, if every point in the region is within the monitoring range of at least k distinct sensors. A wireless sensor network is said to provide k-barrier coverage in a region if every object of interest is guaranteed to be detected by at least k distinct sensors before it completely crosses the barrier of sensors. A wireless sensor network is said to provide k-target coverage in a region if every target in region of interest is guaranteed to be observed by at least k distinct sensors. WSNs are designed to perform monitoring function of an area or set of targets. Each sensor has a sensing area in which any desired event can be sensed. The monitoring quality is solely governed by considered application. The higher the coverage value, the network will be more fault tolerant against failure of the nodes. Some application requires strict monitoring of designated region. Strict monitoring applications require full coverage of the entire area at any point in time during the life span of WSN. The application may further require that each point must be monitored by more than one sensor. In order to guarantee higher level monitoring and reduce the possibility of skipping any interested event, every point of the designated area should be covered by at least one sensor at a time instant. In essence, the main concern is to prolong the network life time to its fullest capacity while meeting the full coverage requirement as per specification of concerned application.

Coverage in sensor networks has received significant research attention in the recent years. Various algorithms have been proposed by researchers for addressing the coverage issues. We briefly review a few well known and pioneer sensing coverage algorithms. Optimal Geographical Density control (OGDC) [3] is one of the popular coverage algorithm which aims to minimize active sensor count and conserve energy while providing the integrated coverage in highly dense sensor networks. Authors have explored the relationship between coverage and connectivity and prove that if the communication radius is at least twice the sensing radius, complete coverage of given convex area implies connectivity among the subset of working nodes. This is a distributed and localized solution of the 1-coverage problem. In [4], author has addressed the target coverage problem and aims to maximizing the network life time by scheduling the sensing activity of

the sensors. The simplified genetic operation and forwarding encoding scheme has been used to reduce the total number of operations. The effect of the sensor redundancy has been analyzed but this has not been incorporate in the design of fitness function. To exploit the redundancy within the cover set, paper proposed different kind of transition schedule. But this has not been in cooperated in the fitness function design.

In this paper, Genetic Algorithm (GA) based scheme is proposed, to address key concern to maximize the network life time with fault tolerant requirement specified by the application. The proposed solution gives the maximum SCS out of the deployed sensor nodes. This ensures the longer operational lifetime of the network. Each cover set contains near optimal number nodes which meet the desired coverage degree requirement.

The remainder of this paper is organized as follows: Section 2 describes problem definition, assumptions and contributions. Section 3 highlights some of the previous related works. Section 4 gives proposed solution; illustrate the mapping of sensor coverage problem to genetic algorithm. In Section 5 simulation results are analyzed. Finally, conclusion of our research and possible future directions are given in Section 6.

2 Problem Definitions and Contribution

2.1 Coverage in WSN

Coverage problem is one of the fundamental issues in a WSN. In any coverage problem, we address the following issues: (a) deployment problem (b) the selection of minimum number of nodes to cover ROI (c) finding the maximum number of disjoint cover set to maximize the network operational life time. Normally, density of deployment is high in WSN applications than required degree of density to monitor the region for user defined degree of coverage. In such a random deployment, there is high probability of presence of redundant nodes. The minimum number of nodes is selected out of deployed number of nodes using specific algorithm to carry out the sensing task and rest of the nodes go to sleep mode to preserve the energy. The sleep active switching of nodes should take place in such a fashion that connectivity among the nodes is ensured. This fulfills two requirements; monitoring of whole area and reporting of the sensed data.

2.2 Problem Definition

Given a convex rectangular region, it is required to partition deployed set of sensors into various sensing cover sets and schedule their working interval so that requested degree of coverage requirement can be fulfilled by these sensing cover set (SCS) and the operational life time of the network is maximized.

Definition 1: **(Convex Region of Interest):** If two considered points fall in the ROI and the straight line connecting theses two points completely falls inside the ROI, the ROI is called convex region.

Definition 2: **(Sensing Cover Set):** Each subset of deployed sensor nodes that can satisfy the desired coverage requirements needed by application is called SCS.

2.3 Contribution

The contributions of this work are as follows:

- We have designed grid based coverage scheme.
- We based our approach on the Euclidian distance of grid corners with centre the sensors. If all centre of a small grid fall inside of a grid, this means grid falls inside the considered sensor.
- We have proposed a GA based coverage approach for selecting the minimum number of sensors to cover the monitored region and finding the maximum number of sensing cover set to increase the network operational time with desired fault tolerance specified by application.
- We have proposed the switching criteria among the sensing cover sets.

3 SCS Determination

3.1 Assumptions

To address the coverage problem, we made some realistic assumption in our work.

- The communication radius of a sensor node is twice of the sensing radius. We have taken the assumption to maintain the integrated coverage.
- We have used the Boolean disk sensing model for simplicity. Any event which take place within the sensing radius of a specific sensor. The event is captured by that sensor with 100% probability and any event outside the sensing radius is not detected by the sensor.
- The node knows its own location in region of interest. Sensor nodes can use any localization techniques to know their relative locations.

Definition 3: **(Sensing Area):** The area which lies within the sensing circular radius of a sensor node is called sensing area of that sensor node. The sensor node can detect any event in this area when sensor node is in active state.

Definition 4: **(K-coverage):** A region of interest (ROI) is said to be k-covered if any location point in the region is monitored by k distinct sensors. Its coverage degree is not less than k, which is required degree of coverage by any specific application.

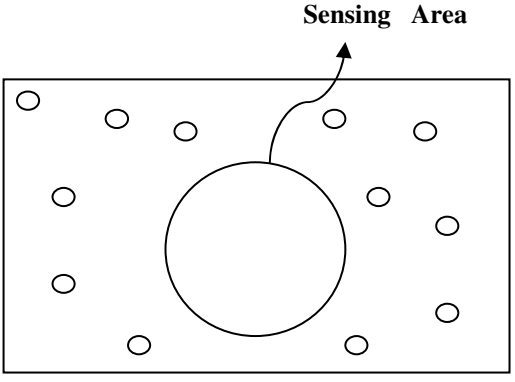


Fig. 1. Sensing area of a sensor node

Definition 5: **(Initial deployment computation):** The minimum number of the node required to cover the given rectangular region for requested degree of coverage by specific application is called the minimum deployment rate. The number of sensor required for k coverage in a rectangular region with dimensions L and W can be given as [5]

$$\frac{M\pi R^2}{L*W} = k \times \frac{2\pi}{\sqrt{27}} \quad (1)$$

Where, R denotes the sensing radius of a sensor.

Theorem: *The region of interest (ROI) can be represented by virtual grids. The initial deployment of sensor nodes provides desired k degree of coverage. This means every grid is fully covered by minimum k distinct sensors. If the area requires single degree of coverage, the maximum number of possible disjoint SCS's which can provide one degree of coverage is k. The maximum number of disjoint SCS's is one if coverage requirement in the area is k.*

3.2 Detailed Description of Algorithm

In the deployment phase, nodes are deployed randomly in the area to be covered. After completion of deployment process, all nodes learn their location coordinate and send their location information to the base station. ROI is represented by the virtual grid. If all grids in the region are covered by the k-sensors, the region is k-covered. If all deployed sensors are activated, the whole area (all grids) must be covered by k-sensors. If the area is not k-covered, initial deployment of nodes fail to achieve the deployment objective. Additional deployment is needed in this scenario to obtain desired coverage. The base station creates node location database and grid database. The base station executes the proposed algorithm. The algorithm takes the following inputs (a) the data base of the node co-ordinate (b) dimension of the rectangular region (c) size of the grid. The algorithm generates the output in the form of disjoint cover sets. The element of the SCS is mutually

exclusive and disjoint. Each node contains the identifier of sensing cover set number. The coverage problem is formulated and mapped in genetic space. Initial population size of chromosome is 10. The deployed set of sensor is represented in the form of gene. If the deployed number of nodes are N, the number of gene in a chromosome will be N.

3.3 Initialization of Population

The deployed sets of nodes cover the entire region successfully. This guarantees that all nodes altogether forming at least one SCS. Each gene in a chromosome is mapped to a sensor in the ROI. The gene values are initialized to one in each chromosome. This indicates that all deployed nodes are member of single SCS. Now partitioning process of initial SCS begins which divides SCS into different SCS which can provide k-coverage. The construction process of subsets takes place in iterative fashion. In first pass, a node which covers the most uncovered portion of ROI is added into SCS. At the end of first pass, a connected node subset is computed and this subset provides user specified coverage degree in monitored region. The value of the gene of this subset is set 2. First pass divides the CS in two subsets. All genes whose value is 2 belong to first subset and rest of genes whose value is 1 belongs to second subset. In the second pass, the subset whose gene value is 1 in first pass is explored to find another subset out of this subset. This leads to further partition of the current subset. This process is repeated after successful completion of second pass to explore the possibility of the further partition. This process continues until the termination criteria are not met. Following conditions form the termination criteria:

- $\mathrm{N} - |\mathrm{SCS}_1| - |\mathrm{SCS}_2| - |\mathrm{SCS}_3| - \cdots > Nopt$
- The coverage provided by number of nodes $\mathrm{N} - \sum_{i=0}^{n} |SCSi|$ is not sufficient to form a cover set.

$$Deployed\ nodes = \bigcup_{1}^{n+1} SCS_i$$

After completion of all passes, the algorithm gives a set of subsets as output. Each subset consists of nodes whose gene value is same. This set consists of the disjoint subsets which can provide the k-coverage in the ROI independently except the subset whose gene value is highest. The highest gene value subset may provide complete or incomplete coverage.

3.4 Evaluation of Fitness Function

The designing of the effective fitness function is key element because it governs the direction of exploration in the search space. An effective fitness function must provide the adequate information so that it can guide towards good candidate solution. The formulation of the fitness function is essentially problem specific. The fitness function of k-set cover problem [6] is normally defined as the total

covers count. This definition does not account the amount of redundancy within a cover set and does not give the precise clarity of fitness of a chromosome. This redundancy effectively reduces the cover count. One way of measuring the redundancy is to find the grid redundancy (GR) in a cover set. The fitness function of a chromosome Ci in the population is defined as

$$F_i = W_1 + W_2 + W_3 + \cdots + W_n + W_{n+1} \tag{2}$$

W_i is weight of the i^{th} subset in a chromosome C_i. The value of W_i is defined as follows:

$$W_i = \begin{cases} 1 & if\ Grid\ Redundancy < 50\% \\ 0.9 & if\ 50\% \leq Grid\ Redundancy \leq 75\% \\ 0.8 & if\ Grid\ Redndancy > 75\% \end{cases} \tag{3}$$

$$\%Grid\ Redundancy = \frac{Total\ grid\ count - No.of\ grid\ exactly\ covered\ by\ one\ sensor}{Total\ grid\ count} \times 100$$

The suitability of a chromosome is judged by its fitness value. The higher fitness value of a chromosome provides the indication that chromosomes contain higher number of subset. If the fitness values of two chromosomes are equal, the tie between chromosomes is resolved by getting better insight about incomplete subset. The chromosome with larger percentage coverage in the incomplete set is declared the winner.

3.5 Selection of Chromosomes and Cross Over

After the initialization phase is over, two chromosomes are randomly selected out of the given population. One to one mapping of two selected chromosome is done. One to one comparison of genes is carried out. The selection of gene which should be copied to offspring chromosome is decided by coin flipping. If heads come, the gene form first chromosome is copied to offspring chromosome otherwise first gene of second chromosome is copied. This process continues until genes in chromosome are exhausted. The output of cross over phase is offspring chromosome. Two chromosomes out of three chromosomes (two parents and one offspring) are selected based on fitness criteria to go into next generation. This ensures next generation population is better. This process is repeated for whole population and in end; new population of original size is obtained as output.

3.6 Mutation

We simply perform the bitwise mutation. The gene is mutated by flipping a coin with some mutation rate. The need of mutation is to create the string in the neighborhood of the current string, thereby achieving the local search around the current solution. The mutation is also used to maintain the diversity. After termination of GA algorithm, we get 10 chromosomes as output. We select best

chromosome according to fitness function criteria. From the best chromosome we can extract the output in the form of sensing cover set SCS_1, SCS_2, SCS_3, …, SCS_n, SCS_{n+1} are complete Sensing Cover Set. The GA gives final output as Cover Set (CS) which is the set of all sensing cover set (subsets). The element of CS is mutually exclusive and each set contain optimal number of elements.

$$CS = \langle SCS_1 | SCS_2 | SCS_3 | \ldots | SCS_n | \rangle$$

4 SCS Scheduling

Each SCS of a cover set has been assigned unique set identifier which is called Sensing Cover Set Identification (SCSID) number. All nodes of single SCS bears the same SCSID number. The base station takes the scheduling decision and informs the nodes of their SCSID number and its schedule through a broadcast message. In this manner nodes know their SCSID number and allotted period of its sensing activity. The span of time during which a SCS is scheduled to carry out its monitoring activity is called working time of SCS. The working time is denoted by $\mathbf{T_w}$ .After the timeout of current SCS $\mathbf{T_w}$, next scheduled SCS gets activated to monitor the area**.** The calculation of $\mathbf{T_w}$ is based on the battery power of a node. This dictates the working duration of each cover set without compromising the monitoring quality.

Table 1. SCSID number and SCS

SCS \ SCSID Number	A_1	A_2	A_3	…	A_n
	SCS_1	SCS_2	SCS_3	…	SCS_n

A cover set is discarded if the cover set is no longer able to provide the required degree of coverage. This happens when the residual energy of node falls below a certain threshold. To gauge this condition, the energy decay profile of the node must be monitored by the base station. These nodes send their energy status embedded in the sensed data forwarded to the base station.

The base station creates the residual energy map of the active sensing cover set. From this residual energy map, base station is able to predict the time to sleep which is periodically updated. If the time to sleep is less than a predefined time period, a broadcast message is triggered by the current active SCS to switch off and next designated SCS to switch on and take over the monitoring responsibility. This takeover time is finite and depends on the node wake up time etc. This scenario is shown in figure 2.

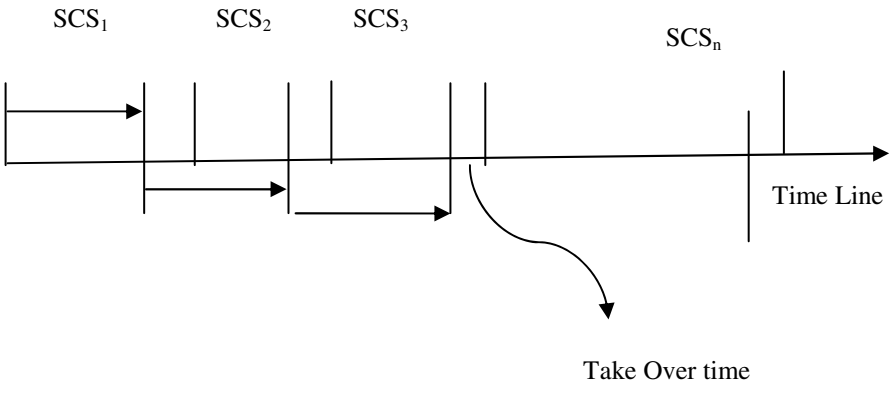


Fig. 2. Schedule of Sensing cover set and Switching

5 Simulation Results and Discussion

5.1 Simulation and Configuration

In our simulation, sensor field is represented by a grid. ROI is 400 m^2 squared area which is divided into 100 uniform square grids. Each grid has the same length of $2m$, and all nodes are equipped with equal sensing radius. The nodes are randomly deployed and their location coordinates are generated as a real number in [0, 20]. A fair distribution of nodes throughout the simulation area has been achieved. The sensing radius and the communication radius which is twice that of the sensing range, of nodes to ensure the network connectivity. Therefore, the communication range is set as 12 *m* and *16 m.*

GA setting: In our experiments, we employed the binary coded standards GA to represent the chromosome. The GA consists of the following components: the random number generation units, fitness unit, random selection, cross over and mutation unit. Single point cross over operation of the rate 1 was employed. Mutation operation is used to restore the genetic diversity lost during crossover and reproduction. We have employed the mutation rate 0.01.

5.2 Performance Metrics, Experiments and Results Discussion

Coverage Ratio: Coverage ratio is defined as the ratio of the number of grid covered by each SCS to the total number of grids in the monitored area.

$$Coverage\ Ratio = \frac{No.of\ Grids\ Covered}{Total\ \text{No.}of\ Grids}$$

The coverage ratio is one of the prime metric to evaluate the performance of any coverage algorithm. We have considered 100% as performance metrics.

Node Count in Cover Set: The lower node count in a cover set; lower the redundancy within the cover set. Redundancy within the cover set reflects the lower cover set count. This directly affect the network life time.

SCS Count: The higher the cover set count, the higher will be life time of the network. Thus, finding the high number of set cover is one of the desired characteristics of a good coverage.

Experiment 1: In this experiment, we evaluated the 1-coverage of the network achieved using the proposed algorithm. Since this algorithm divides the nodes into mutually exclusive and disjoint sets, we plotted the sensing cover set count for sensing radius 6 m and 8 m. The result is obtained by simulating the algorithm 15 times. It can be observed from Fig. 3 that it requires approximately 20 nodes with a sensing radius R = 6m to cover a 400 m^2 area whereas the same number of nodes with R=8m are able to provide a minimum 2-coverage resulting in two SCS. Moreover, as the node density increases, the number of SCS with higher sensing radius nodes increasing sharply as compared to nodes with lower sensing radius even though the increase in number of SCS is linear in both the cases. As the number of nodes increase, the number of SCS number increases steeply with R=8m. When the number of nodes is around 120, there is no further increase in SCS even though there seems to be no plausible reason for the same.

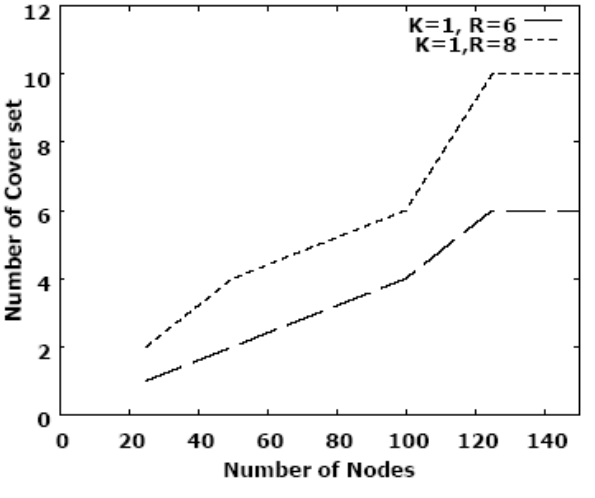


Fig. 3. Plot showing the cover set count for K=1

Experiment 2: In this experiment, we varied the number of deployed nodes in monitored area. We evaluated the number of cover set for fix coverage degree K = 2 for sensing radius value 6 m and 8 m.

Fig. 4 shows that it requires approximately 20 nodes to cover a 400 m2 area and no fault tolerance is achieved both nodes with R=8 or 6m. However, as the number of nodes increase, the nodes with greater sensing radius are able to form two SCS

while sensors with lower sensing radius form just one SCS. As expected, with further increase in number of nodes, the number of SCS increases monotonically for both types of sensor nodes.

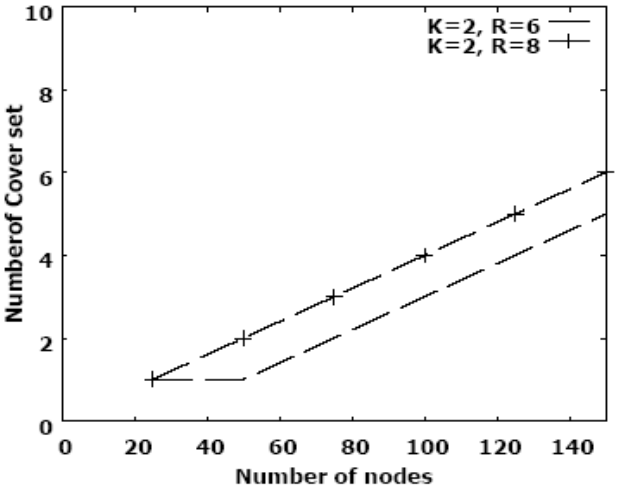


Fig. 4. Plot showing the cover set count for K=2

Experiment 3: In this experiment, we evaluated the number of sensing cover set for sensing radius 6 m and 8 m with coverage requirement K = 3. We varied the deployed number of nodes ranging from 25 to 150.

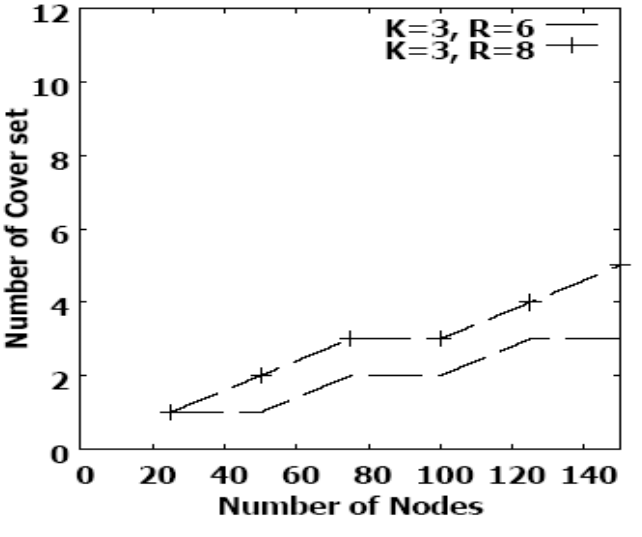


Fig. 5. Plot showing the cover set count for K=3

When 25 nodes were deployed, the nodes of both the sensing radii provided K= 3 coverage. This meant that nodes with R = 8m has inherently higher redundancy even though only one complete SCS was formed. Fig. 5 prove this observation to be correct as the number of SCS increases quickly with the increase in node density for nodes with R=8m. The SCS count does not, however, increase for R=6m. As expected, the SCS count for both the node types increases slowly as large number of nodes are required for k=3 coverage and the increase in SCS number is quite low even when the number of nodes increase beyond hundred.

6 Conclusion and Future Work

In this paper, we have considered grid based sensor network in which nodes are placed over 400 m^2 squared region. We have presented a genetic algorithm for area coverage with fault tolerance requirement. We have considered the redundancy within cover set for designing suitable and effective fitness function. This exploitation of redundancy within cover set improves the cover set count. This leads to prolonged system life time. One possible direction of future work would be to present fault tolerant area coverage problem with energy analysis at node level as well as network level.

References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine, 102–114 (August 2002)

[2] Slijepcevic, S., Potkonjak, M.: Power Efficient Organization of Wireless Sensor Networks. In: Proceedings of the IEEE International Conference Communication, Finland, vol. 2, pp. 472–476 (2001)

[3] Zhang, H., Hou, J.C.: Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. In: The Technical Report UIUC, UIUCDCS-R- 2003-2351 (2003)

[4] Hu, X.M., et al.: Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks. IEEE Transaction on Evolutionary Computation 14, 766–781 (2010)

[5] Williams, R.: The Geometrical Foundation of Natural Structure: A Source Book of Design, pp. 51–52. Dover, New York (1979)

[6] Abrams, Z., Goel, A., Plotkin, S.: Set k-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks. In: The Proceedings of the 3rd International Symposium Information Processing in Sensor Network, pp. 424–432 (2004)

Optimizing Supply Chain Management Using Gravitational Search Algorithm and Multi Agent System

Muneendra Ojha

Department of Information Technology, Indian Institute of Information Technology, Deoghat, Jhalwa, Allahabad 211012, UP, India
muneendra@iiita.ac.in

Abstract. Supply chain management is a very dynamic operation research problem where one has to quickly adapt according to the changes perceived in environment in order to maximize the benefit or minimize the loss. Therefore we require a system which changes as per the changing requirements. Multi agent system technology in recent times has emerged as a possible way of efficient solution implementation for many such complex problems. Our research here focuses on building a Multi Agent System (MAS), which implements a modified version of Gravitational Search swarm intelligence Algorithm (GSA) to find out an optimal strategy in managing the demand supply chain. We target the grains distribution system among various centers of Food Corporation of India (FCI) as application domain. We assume centers with larger stocks as objects of greater mass and vice versa. Applying Newtonian law of gravity as suggested in GSA, larger objects attract objects of smaller mass towards itself, creating a virtual grain supply source. As heavier object sheds its mass by supplying some to the one in demand, it loses its gravitational pull and thus keeps the whole system of supply chain perfectly in balance. The multi agent system helps in continuous updation of the whole system with the help of autonomous agents which react to the change in environment and act accordingly. This model also reduces the communication bottleneck to greater extents.

Keywords: Multi Agent Systems, Supply Chain management, Particle Swarm Optimization, Gravitational Search Algorithm, Food Corporation of India.

1 Introduction

Supply chain management can be defined as "a goal-oriented network of processes and stockpoints used to deliver goods and services to customers" [1]. Science of supply chains deals with an array of suppliers, plants, warehouses, customers, transportation networks and information systems that make up actual

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 481–491.
springerlink.com

supply chains. But to study it we must break the overall complexity to a level which is relevant to the task at hand. We must define the application domain and then address the issues which are most important. In the above definition processes represent activities involved in producing and distributing goods or services. On the other hand stockpoints in this definition refers to the locations in supply chain where inventories are held. Finally processes and stockpoints are connected by a network describing various ways in which goods can be supplied through the chain. Supply chain has four levels of decision classification hierarchy - strategic, tactical, operational, and real time level, where strategic level is the most important among these, dealing with the resource assignment, goods movement, location of facilities to make it cost effective etc [2]. Success rate at other levels depend heavily on the decisions taken at this levels. However, the decision making is a difficult process as it involves the resolution of complex resource allocation and scheduling problem involving selection of different available stakeholders. According to the findings of Beamon [3], and Amiri [4], finding the best solution for supply chain management is NP-hard problem, so it must be strategically dealt with, on case to case basis developing efficient methodology that can find out the optimal or near-optimal solution in minimum computational time.

Over the last decades, research community has shown greater interest in analyzing natural phenomena and mapping their models to find out optimized solution for high-dimensional search space cases. Scientists have adopted various meta-heuristic methods like Genetic Algorithm [5], Simulated Annealing [6], Ant Colony Search Algorithm [7], Particle Swarm Optimization [8] etc. and shown that these algorithms are able to find efficient solution of complex computational problems like pattern recognition [9], image processing [10], filter modeling [11] etc. As like any other heuristic method, these algorithms do not promise to provide the most accurate and best solution for the problems but try to give the most optimized one, again on the case to case basis. So applying a new optimization algorithm on an untested domain is always an open problem.

In this paper we discuss the results of applying a relatively new swarm optimization algorithm, the Gravitational Search Algorithm (GSA) [12] in supply chain management domain. We have modeled the software application using multi agent system where intelligent agents autonomously take decision according to the changing environment and adapt to the perceived needs. In the following section we describe the relevance of GSA and Multi agent system in finding the solution of supply chain problem. Later we explain the problem structure with a mathematical model for the proposed clustering algorithm. Finally we end with implementation details and further scope of the research work.

2 Combining Multi Agent Technology with Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computational technique used for optimization motivated by the social behavior of individuals in large groups in nature [13]. Scientists and technicians have used different approaches to

understand how such algorithms works and applied variations to optimize a large number of real life problems. They have been able to solve a variety of engineering problems, but fine tuning the PSO parameters as well as finding general criteria applied to all problems remains a major challenge. Thus we focus on a particular problem and try to find the best algorithm which is well suited to its requirements.

2.1 Why GSA Is Relevant in Supply Chain Management Problem?

Swarm based algorithms use a collection of agents like ants or honey bees and perform parallel search with multiple starting points. Each member carries out its intended operation with a target to maximize respective utility function, and shares its information with others. These operations are normally very simple, but their collective result produces astonishing results. This phenomenon in which local interaction between agents produce a global result facilitating the system to solve a problem without involving any central authority is known as swarm intelligence [14]. Gravitational Search Algorithm (GSA) is also a particle swarm optimization algorithm which works on the Newtonian laws of Gravity: "Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them" [12]. It has found a good number of interested audiences albeit being relatively new. Just like other swarm intelligence techniques, it also models the laws of nature to find out an optimized solution to a non linear and dynamic problem domain. In the GS algorithm, each entity is considered an object whose performance is measured by the mass it carries. All objects attract each other by gravitational force which causes global movement towards objects of heavier mass. As heavier masses move slower than lighter ones, this creates virtual clusters representing better solutions. For the case of supply chain management, we assume all stakeholders including producer, consumer, supplier, warehouses, strategic partners, vendors, and haulers etc, as objects with some respective masses. We divide these stakeholder in two groups i.e. suppliers and customers. One can be supplier at some point of time and customer at other. Although we do not consider the demand part of the SCM process, if we focus on calculating the total cost of moving goods from a supplier to a customer, we can simulate a major part of the whole process of supply chain management. Cost of moving might include purchase price, cost of quality, inventory losses, transportation costs and any other factor which directly or indirectly affects the amount of money the customer has to shed. Thus suppliers act as objects of heavier masses or stars and customers imitate objects with lower masses, creating planetary systems revolving around stars.

2.2 Multi Agent System Representing the Complete Supply Chain

A rational agent is defined as an agent, which has clear preferences, models uncertainty via expected values, and always chooses to perform the action that results in the optimal outcome for itself from among all feasible actions [15]. A coordinated

group of rational agents form the basis of multi agent system. Multi agent system is based on the assumption that agents coordinate among each other with the vision of achieving a common global goal. Referring the goal to be global is important here as individual agents have only local knowledge and thus local goals. But the agent system as a whole strives for one particular target to achieve. So whatever be the individual motives of agents, it should add an extra step towards the combined target of system and this is the raison d'être of these agents [16]. When an event happens each agent cooperates to organize a collaborative group to solve the problem. Agents communicate among the nearest of neighbors thus eliminating number of communications and form virtual clusters. In supply chain management (SCM) also, more and more consumers collect towards a prominent supplier thus forming some kind of demand supply cluster, where demands of consumers are fulfilled by nearest of supplier. Thus MAS presents an intuitive method of demand supply chain implementation in which each agent represents one or other real life entity participating in the whole process.

Many preceding studies have considered multi-agent system as the most suited way to solve complicated problems under the diverse environmental dynamisms [17][18]. A number of researchers have also successfully applied the multi agent technology for manufacturing, supply chain management, planning, scheduling, and control [19]. Their studies mainly focused on making better manufacturing system using an agent technology for automation and efficiency in the process planning and scheduling. Also, there are representative studies that have used MAS in developing intelligent manufacturing supply chain management systems e.g. the AARIA (Autonomous Agents for Rock Island Arsenal) project of Intelligent Automation company [20], ABCDE (Agent-Based Concurrent Design Environment) system developed by KSI (Knowledge Science Institute) of University of Calgary [21], a virtual manufacturing agent made by LIPS institute of the University of Texas at Austin [22] and MASCOT (Multi-Agent Supply Chain cOordination Tool) agent of Intelligent Coordination and Logistics team of Carnegie-Mellon University [23].

In the SCM problem, GSA with multi agents system is intuitively the most suited method of implementation. Every participating entity either a consumer or supplier is represented by an agent. Every agent carries some mass. All heavier agents are supplier and lighter ones are consumers. As the problem under consideration is the distribution among centers of Food Corporation of India (FCI), mass determines the total amount of food grains one center has in its stock. A center can be at the same time supplier for one type of agricultural produce (e.g. wheat) and consumer for some other type of grains (e.g. rice). Thus effective mass and gravitational pull is context dependent.

3 FCI Distribution System and Methodology

Food Corporation of India (FCI) was established to fulfill the objectives of procurement of agricultural products, its storage, movement, export if asked, disposal in case of excess amount, and quality control. FCI has five zones with each zone having its member state representing region.

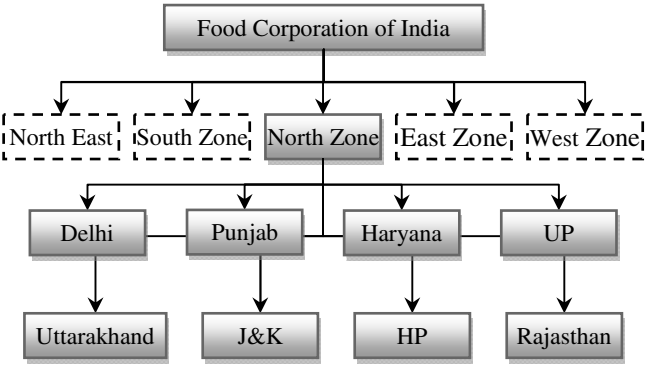


Fig. 1. Hierarchical structure of FCI collection and distribution centers. North zone itself has 54 district centers in total involved in the procurement and other food management activities.

For example, North zone of FCI has eight regions namely Delhi, Haryana, HP, J&K, Punjab, Rajasthan, UP and Uttarakhand regions as depicted in the Figure 1, which in turn have 54 district centers. Punjab & Haryana Regions are the major surplus states as far as production/procurement of food grains is concerned. More than 90% stocks of these supplier states are moved to other parts of the country [24]. Apart from FCI the other major agencies involved in storage and warehousing of food grains in public sector are Central Warehousing Corporation (CWC) and 17 State Warehousing Corporations (SWCs). For procuring food grains from farmers, FCI creates nodal centers where it purchases at a price fixed by the Government. It is then sent to respective region's storage centers or warehouses. Every warehouse has a minimum requirement as well as maximum capacity.

With the broad requirements clear about FCI and its supporting agencies, we can model the whole distribution system as MAS. An agent in our system represents a district center of FCI. As there are 54 district centers in North Zone, so there will be 54 agents corresponding to them. Each agent thus would have a minimum requirement, i.e. a minimum amount of particular food grain, which it has to maintain as reserve at any given instance. This reserve is decided on many factors like, Government of India policy on food security, type of food, population of the region, requirements for the region in near future, rate at which food is being consumed in public distribution system, state rules etc. Any amount of food above this reserve level is assumed to be surplus which can be sent to other agents in the system if required. Thus we can clearly differentiate between supplier and consumer at any point of time using following definition:

Supplier: *An agent which has food grains in excess of its set reserve amount and any point of time.*

Consumer: *An agent which has food grains below the decided reserve at an instance of time.*

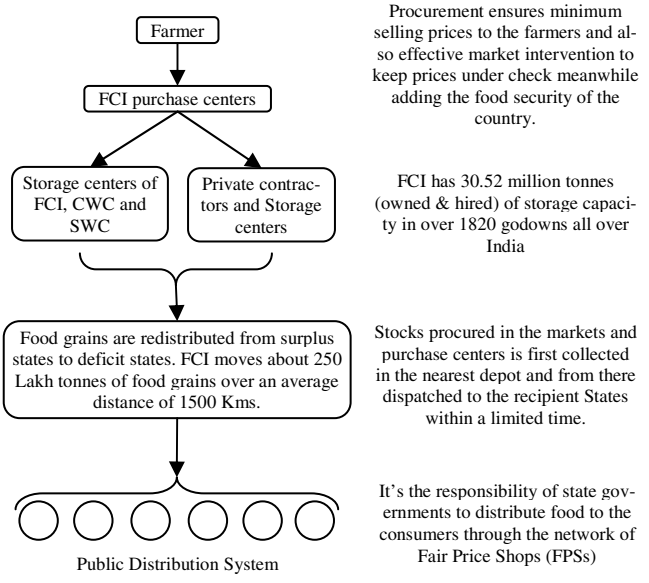


Fig. 2. Broad functionality and objectives of FCI and its supporting agencies like Central Warehousing Center, State Warehousing Center and Public Distribution Centers. Source: Food Corporation of India [24]

As this definition is dependent on time and type of food grain, any agent can be a supplier for one type and consumer for other type of food grains at the same point of time. Two reasons can be visualized for automatic redistribution of food grains between centers: first, when a center needs it because it has receded below its reserve or when a center has accrued more than what it can efficiently manage. Both of these cases prompt their representative agents to autonomously act in case such scenario is perceived in the environment. Though FCI has its structural limitations and boundaries, we assume it to be free from any such regulations. Although we do suggest a priority queue for suppliers in which higher priority must be given to consumers of the same zone or state rather than a consumer of different zone. Figure 2 gives a broad idea of the levels at which different agencies work and how food grains purchased from farmer reaches household.

4 The Mathematical Model

Each participating physical entity in this supply chain system is represented by an agent. There are agents for zonal offices, regional states, district centers, nodal centers, storage warehouses, and purchase centers. Out of these, any agent can be supplier or consumer depending on the resources it owns. A consumer is consumer till it has requires food grains to achieve a particular target level referred to as set reserved limit. It is decided by an array of parameters namely, type of food grain, near future requirement prediction, number of Public Distribution Centers

(PDCs) it caters, geographical area this agent covers, positional importance, and minimum food reserves it has to maintain (decided as a policy matter by government, hardcoded but can be changed if needed).

In GSA, initially the system is considered a randomly distributed isolated space of masses (agents). Each agent carries some mass and in gravitational system, every object gets attracted by gravitational pull of another object. In our system we do not assume an agent to move. Rather we assume that whichever supplier exerts greatest force of attraction on consumer will be the first one to attempt in fulfilling its requirements.

At a time 't' we define force acting on a consumer mass 'M_c' from a supplier mass 'M_s' to be

$$F_{cs}(t) = -G_c(t)\frac{M_c(t)\times M_s(t)}{R} \quad (1)$$

Where,

- F_{ij}(t) = Force of attraction by a supplier and consumer at time t.
- $M_c(t)$ = Mass of Consumer at time t
  = Amount it can supply
  = Reserve limit – Current Stock
- $M_s(t)$ = Mass of Supplier at time t
  = Amount it needs
  = Reserve limit – Current Stock
- $G_c(t)$ = Urgency factor, initially set randomly between 0 to 1. Increases with time.
- R = Possible cost incurred in supplying food grains per unit to the consumer.

An agent can be attracted by another agent by a positive force of attraction. Putting a minus sign in equation 1 ensures that no two consumers or suppliers are attracted towards each other.

Unlike the traditional gravitational search algorithm proposed by [12] we have made some changes in our algorithm. We consider the attracting force to be one dimensional only. We were thus able to eliminate the possibility in which multiple smaller suppliers could collaborate and exert greater force than a single large supplier. In this case combined force could take a consumer to cluster of smaller suppliers but do not consider the cost incurred in multiple smaller transportations. We have also set R to be the cost incurred per unit where,

$$F \propto 1/R \quad (2)$$

As cost increases attraction between supplier and consumer decreases though it never becomes zero as is the case in gravitational forces of nature as well. Gravitational constant G_c is initially randomly set as the urgency factor defining how urgent is the need of this particular customer. As food is supplied to it and need

decreases, value of G_c also changes with time. Thus it is a function of initial value G_0 and time t:

$$G(t) = f(G_0, t) \tag{3}$$

Masses of individual agents are also dependent on time t. As supply starts, the total available stock of both consumer and supplier change. This changes the respective masses and thus mutual force of attraction is also decreased. If there are N agents, position of an agent is defined in 2-dimensional space as:

$$X_i = (x_i^1, x_i^2) \; for \; i = 1,2, \ldots, N \tag{4}$$

When agents attract each other they tend to move towards each other in the virtual space. This movement is not reflected in the real scenario nor has any physical significance. Maximum closeness between two agents during clustering phase is restricted to sum of their radii.

5 Implementation and Result Discussion

Application simulation is divided in two phases. First one is the clustering phase when we form virtual clusters of supplier consumer agents depicting some kind of planetary system in which supplier is the star and consumer are the planets revolving around it. As the clustering phase starts, agents get attracted towards those of greater mass and their respective Euclidean distance decreases. When clustering ends each supplier is attached with many consumers as shown in Figure 3. If the supplier has enough capability to fulfill demands of more than one consumer it must decide which consumer to cater first. Thus every supplier creates a priority queue on the basis of first hierarchical boundaries of FCI and then in the decreasing order of consumers' urgency factor. Hierarchical boundaries are important here as we assume an agent will cater to the needs of its state prior to sending

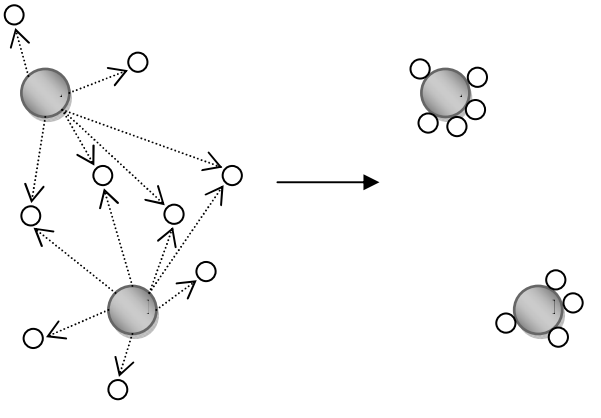


Fig. 3. Initially all agents are scattered in the search space and are attracted by more than one supplier. But they move closer to the one which has greater force of attraction forming virtual clusters.

grains to neighboring state. Providing autonomy to the agent system, we assume that if a consumer finds a supplier 'A' in neighboring state providing cheaper food than agent 'B' in the same state it will approach 'A' rather than 'B' as long as both states are in same zone of FCI.

Actual supply of goods can only be realized physically and once it is approved and done, agents' information must be updated manually. As the system itself notifies, how much food is to be provided by a supplier to the consumer; it can automatically update its inventory after approval. Once the inventory is updated, effective mass of both supplier and consumer changes. Agents that were customer for some time but their needs have been satiated will move out of the SCM process and similarly an agent that was a supplier for some time but its excess mass is already shed, will also exit the system. This behavior is perfectly represented by multi agent system as MAS gives autonomy and intelligence to its agents to participate in the process as long as required or leave the environment whenever deemed necessary.

There is no need of a centralized command to dictate the centers which must be catered. Agents automatically form collaborative supplier-consumer clusters and cater to the needs of each other thus removing the bottleneck of central node. Number of consumers and suppliers as a given time can be generated using Poisson distribution. As soon as a center becomes consumer the representative agent passes on its identification and requirement message to its immediate neighbors which in turn pass it on to their respective neighbors and so on. With the one hop communication channel, agents do not need to broadcast requirement to whole network thus reducing burden on the network. All the supplier agents respond back with their identification and availability. When consumer receives response of a supplier it refers to the database in order to fetch supplier information like location, state, zone, modes of transportation, price of grain and any other constraints based on which the final cost is calculated. It calculates the attraction force using equation 1 and updates this agent as the global best. A consumer also maintains list of suppliers for as soon as new supplier comes up, the queue is reshuffled and new global best is updated if required. In case one supplier is not able to fulfill the requirement, consumer will approach the next in queue without pronouncing its need again on the network.

6 Conclusion

The current system can be implemented for very large scale real life demand supply chain management. This system can be applied to observe the pattern of product usage, to find out high demand zones, or the consumption rate of an area. It can also draw out the reliability factor of a supplier and cost to benefit ratio for a product. This system can also help organizations in evaluating needs of customer and adapt accordingly.

Acknowledgments. We would like to thank Dr. M.D.Tiwari, Director, IIIT – Allahabad for his support and Prof. R.B.Mishra, Head, Department of Computer Engineering, IT BHU for his guidance as well as encouragement.

References

[1] Hopp, W.J.: Supply Chain Science. McGraw Hill, New York (2006)

[2] Poirier, C., Quinn, F.: How are we doing? A survey of supply chain progress. Supply Chain Management Review, 24–31 (2004)

[3] Beamon, B.M.: Supply chain design and analysis: models and methods. International Journal of Production Economics 71(1-3), 145–155 (1998)

[4] Amiri, A.: Designing a distribution network in a supply chain system: formulation and efficient solution procedure. European Journal of Operation Research 171(2), 567–576 (2006)

[5] Tang, K.S., Man, K.F., Kwong, S., He, Q.: Genetic algorithms and their applications. IEEE Signal Processing Magazine 13(6), 22–37 (1996)

[6] Kirkpatrick, S., Gelatto, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)

[7] Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B 26(1), 29–41 (1996)

[8] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)

[9] Tan, X., Bhanu, B.: Fingerprint matching by genetic algorithms. Pattern Recognition 39, 465–477 (2006)

[10] Nezamabadi-pour, H., Saryazdi, S., Rashedi, E.: Edge detection using ant algorithms. Soft Computing 10, 623–628 (2006)

[11] Lin, Y.L., Chang, W.D., Hsieh, J.G.: A particle swarm optimization approach to nonlinear rational filter modeling. Expert Systems with Applications 34, 1194–1199 (2008)

[12] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Information Science 179(13), 2232–2248 (2009)

[13] Fernández-Martínez, J.L., García-Gonzalo, E.: What Makes Particle Swarm Optimization a Very Interesting and Powerful Algorithm? In: Panigrahi, B.K., Shi, Y., Lim, M.-H. (eds.) Handbook of Swarm Intelligence. ALO, vol. 8, pp. 37–65 (2011)

[14] Tarasewich, P., McMullen, P.R.: Swarm intelligence: power in numbers. Communication of ACM 45, 62–67 (2002)

[15] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, New Jersey (2003) ISBN: 0-13-790395-2

[16] Nwana, H.S., Lee, L.C., Jennings, N.R.: Coordination in software agent systems. The British Telecom Technical Journal 14(4), 79–88 (1996)

[17] Choi, H.S., Kim, H.S., Park, B.J., Park, Y.S.: Multi-agent Based Integration Scheduling System Under Supply Chain Management Environment. In: Orchard, B., Yang, C., Ali, M. (eds.) IEA/AIE 2004. LNCS (LNAI), vol. 3029, pp. 249–263. Springer, Heidelberg (2004)

[18] Julka, N., Karimi, I., Srinivasan, R.: Agent-based supply chain management-2: A refinery application. Computers and Chemical Engineering 26, 1771–1781 (2002)

[19] Shen, W., Ulieru, M., Norrie, D.H., Kremer, R.: Implementing the Internet enabled supply chain through a collaborative agent system. In: Proceedings of Workshop on Agent Based Decision Support for Managing the Internet Enabled Supply-Chain, Seattle, pp. 55–62 (1999)

[20] Baker, A.D., Parunak, H.V.D., Erol, K.: Manufacturing over the Internet and into your living room: Perspectives from the AARIA project (Tech. Rep. TR208-08-97). ECECS Dept (1997)

[21] Balasubramanian, S., Norrie, D.H.: A multi-agent intelligent design system integrating manufacturing and ship-floor control. In: The Proceedings of the First International Conference on Multi-Agent Systems. The AAAI press/The MIT Press, San Francisco (1995)

[22] Chuter, C.J., Ramaswamy, S., Baber, K.S.: A virtual environment for construction and analysis of manufacturing prototypes (1995), http://ksi.cpsc.ucalgaly.ca/projects/mediator (retrieved)

[23] Norman, M.S., David, W.H., Dag, K., Allen, T.: MASCOT: an agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In: Proceedings of the Third International Conference on Autonomous Agent (Agents 1999), Seattle, WA (1999)

[24] Food Corporation of India, http://fciweb.nic.in

A Fuzzy Trust Model for Argumentation-Based Recommender Systems

Punam Bedi and Pooja Vashisth

Department of Computer Science, University of Delhi, Delhi, India
punambedi@gmail.com, poojavashisth@rediffmail.com

Abstract. With the enormous growth of the Internet and Agent-based E-commerce, online trust has become an increasingly important issue. The fact that multi-agent systems are vulnerable with respect to malicious agents poses a great challenge: the detection and the prevention of undesirable behaviors. That is the reason why techniques such as trust and reputation mechanisms have been used in literature. In this paper, we propose a fuzzy trust model for argumentation-based open multi-agent recommender systems. In an open Agent-based Recommender System, the goals of agents acting on behalf of their owners often conflict with each other. Therefore, a personal agent protecting the interest of a single user cannot always rely on them. Consequently, such a personal agent needs to determine whether to trust (information or services provided by) other agents or not. Lack of a trust computation mechanism may hinder the acceptance of agent-based technologies in sensitive applications where users need to rely on their personal agents. Against this background, we propose an extension of the basic argumentation framework in Agent-Based Recommender Systems to use the fuzzy trust within these models for trustworthy recommendations.

Keywords: Intelligent Agents, Recommendation Agents, Trust, Argumentation Framework, Fuzzy logic.

1 Introduction

Recommender systems (RS) have evolved in the last years as tools to assist users in a variety of computer mediated tasks by providing guidelines or hints. At the same time, the emergence of large E-commerce sites has highlighted several problems regarding issues of trust and deception in these markets. The lack of information about the background, especially the trustworthiness of the participants causes a lot of mistrust among participants [1]. The fact that open multi-agent systems are vulnerable with respect to malicious agents poses a great challenge. Therefore, trust is an especially important issue from the perspective of autonomous agents and multi-agent systems. The premise behind the multi-agent systems field is that of developing software agents that will work in the interests of their "owners",

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 493–502.
springerlink.com

carrying out their owners' wishes while interacting with other entities [3, 7]. In such interactions, agents will have to reason about the degree to which they should trust those other entities. Consequently, such a personal agent needs to be able to reason about trusting (information or services provided by) other agents [4, 9].

The paper is organized as follows. In Section 2, we discuss related work and section 3 presents the argumentation framework in IBR. In Section 4, the core idea of our approach, i.e., a combination of a fuzzy modeling technique is applied to an argumentation framework for Interest-Based Recommendation (IBR) is presented for making trustworthy decisions. Later section describes the concept with the help of a walk-through example to motivate our approach. Finally, section 6 concludes the paper and gives directions for future work.

2 Related Work

Trust is an approach for measuring and managing the uncertainty about autonomous entities and the information they deal with. As a result trust can play an important role in any decentralized system. Trust is especially an important issue from the perspective of autonomous agents. Several algorithms have been devised to confront the problem of estimating trustworthiness by capturing past experiences in one or two values to estimate future behavior. At any rate, these approaches do not provide human readable information about these decisions. Therefore, the core idea of the approach in [8] was to present a unique combination of a fuzzy rule opponent modeling technique and a solid argumentation framework was applied to the process of making trust decisions supported by rational explanations. Also, the work by [6] presents a first approach towards the integration of web-based recommender systems with a defeasible argumentation framework. The goal was to incorporate argument-based qualitative inference and trust in RS to enhance its reasoning capabilities. A novel recommendation-based trust model [10], utilizes the theory of collaborative filtering and ant colony optimization, and includes two indexes of trust scenario similarity and propensity to trust similarity. Recently, [5] proposed fuzzy computational models for both trust and reputation concepts. Reciprocity and experience were used for trust modeling while the reputation model is a fuzzy extension of beta reputation model. In this direction of research, the proposed work in this paper investigates the combination of trust measures on agents using argumentation for reasoning and interaction; by combining an existing system for reasoning about trust [4, 5] and an existing system of argumentation for IBR [2].

3 Argumentation Framework in IBR

In this section, we briefly present our argumentation-based framework for recommender systems known as Interest-Based Recommender or IBR [2]. These agents have a BDI architecture (Beliefs, Desires, and Intention) augmented with argumentation and logical reasoning. The agent architecture is composed of two models: the mental model, and the reasoning model. The mental model includes

beliefs, desires, goals, etc. The agents must use their reasoning capabilities to reason about their mental states before taking any decisions. The agent's reasoning capabilities are represented by the reasoning model using an argumentation system. To deal with the different nature of the arguments involved, we have developed three distinct argumentation frameworks: one for reasoning about beliefs, another for arguing about what desires should be pursued, and a third for arguing about the best plan to intend in order to achieve these desires. These beliefs, desires and the related, supporting arguments can be used to generate an interesting recommendation or even to defeat one. During recommendation seeking process, agents can establish a common knowledge of each other's likes (satisfaction) and dislikes (dissatisfaction), find compromises, and persuade to make decisions through argumentation. An argumentation system is simply a set of arguments and a binary relation representing the attack-relation between the arguments. The following definition, describe formally an argument. Here $\mathcal{K}_{\mathcal{B}}$ indicates a possibly inconsistent knowledge base. $\vdash$ stands for classical inference and $\equiv$ for logical equivalence.

Definition 1 (Argument). *An argument is a pair* (H, h) *where h is a formula of a logical language and H a sub-set of* $\mathcal{K}_{\mathcal{B}}$ *such that i*) *H is consistent, ii*) $H \vdash h$ *and iii*) *H is minimal, so no subset of H satisfying both i) and ii) exists. H is called the support of the argument and h its conclusion.*

The allowed communicative acts are: Inform, Recommend, Accept, Refuse, Challenge, Justify, Repair, and Attack.

4 Proposed Fuzzy Trust Model for an Argumentation-Based Recommender System

In recent years, several models of trust have been developed in the context of MAS and recommender systems. Various definitions have been given in the literature to both trust and reputation concepts. However, these models are not designed to trust argumentation-based recommendation agents. Their formulations do not take into account the elements we use in our recommendation approach like accepted and refused arguments, satisfied and dissatisfied with recommendations.

Also, trust and reputation concepts are subjective and fuzzy concepts thus fuzzy computational models for both the concepts would reflect their real value for improving the recommendation accuracy. Two main sources of information frequently used for trust and reputation modeling, are namely direct information and word of mouth information (witness information). The direct information is the experience based on the direct interactions with a specific agent while witness information is the information that comes from other members or neighbors in the community. In this section we present our fuzzy trust model for the argumentation based recommendation systems (Figure 1).

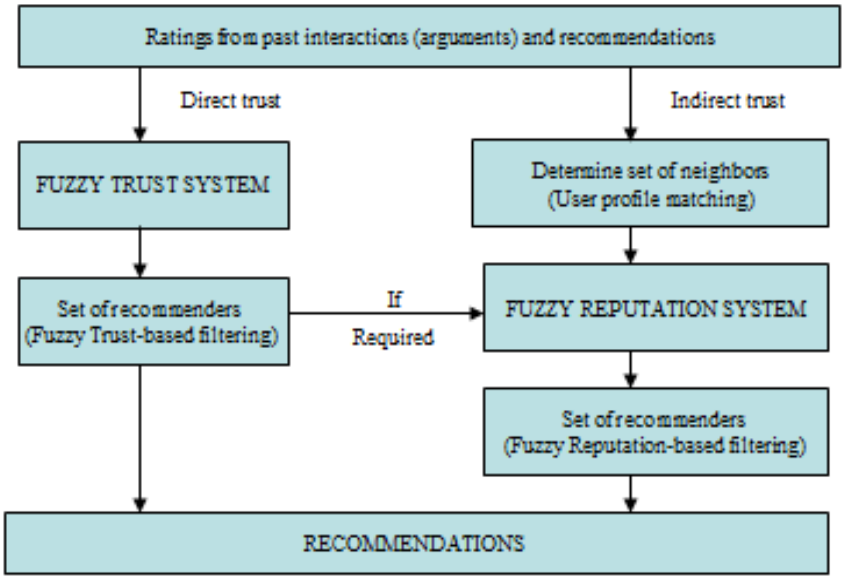


Fig. 1. The proposed methodology for incorporating trust and reputation in RS

4.1 Estimating Agent's Trustworthiness

Our fuzzy trust model for an argument-based recommender system is built on the following definitions.

Definition 2 (Trust). *Trust is a subjective expectation a partner has about another's future behavior based on the history of their encounters [5].*

These encounters in an argument-based recommendation system consist of acceptable, unacceptable arguments, satisfactory and unsatisfactory recommendations. These facts are computed from the agent's local beliefs.

Definition 3 (Reputation). *Reputation is what is generally said or believed about a person's or thing's character or standing.*

Thus reputation exists only in a community which is observing its members. Accordingly, reputation is the collected and processed information about one agent's past behavior as experienced by others. These facts are computed from the global beliefs. The score obtained is the aggregation and is visible to all members of the community. Thus the reputation score reflects a global view of each individual. Also, a high trust measure of an agent can override even high reputation scores based on the past direct experience with the other agents.

Trust and reputation systems are rating systems where each individual is asked to give his opinion after completion of each interaction in the form of ratings (it can be implicit or explicit). In our work, trust and reputation values are automatically inferred from the rating database of the RS and thereafter these values are used to enhance the accuracy of the recommendation process. These interactions

consist of arguments and the recommendations. More formally, let A = {a_1,a_2, . . ., a_M} be the set of all agents, where M is the number of agents in the system. We assume each agent will rate the other after completing the interaction. An interaction i∈ I, where $r_{xy}(i_k)$ is the rating agent x has given to agent y for an interaction i_k. The rating scale or grade for interactions is defined as G = {-2, -1, 0, +1, +2}. It can be translated into linguistic form as very poor, poor, moderate, good, and very good, respectively. This is because the binary ratings are not considered enough to present various degrees of judgment; therefore G should give more than one choice for positive/negative ratings besides a neutral rating. As a consequence, ratings can be easily assigned and understood by human users and therefore a more accurate judgment can be obtained. The set of ratings agent x has given to agent y is S_{xy} = { $r_{xy}(i_k)$ | i_k∈ I } and the whole past history of agent x is H_x = {S_{xy} | ∀ y(≠ x) ∈ A}. The agent's rating for an interaction (argument) i_k is derived implicitly from the number of matches the argument parameters strike with the agent's preference list. The rating for an interaction (recommendation) i_k is given explicitly by the user on scale G as mentioned above.

Given that A is the set of agents.

We define an agent's trustworthiness as follows: $TRUST : A \times A \times F \rightarrow [0,1]$.

This function associates to each agent a fuzzy measure representing its trustworthiness according to another agent. To evaluate the trustworthiness of an agent y, an agent x uses the history of its interactions with y. Equation 1 shows how to calculate this fuzzy measure of trustworthiness.

$$TRUST_{xy} = \frac{\sum_{Arg \in I} FM\_agree(Arg_{xy}) - \sum_{Arg \in I} FM\_disagree(Arg_{xy}) + \sum_{R \in I} FM(R_{xy})}{T\_N\_Arg_{xy} + T\_N\_R_{xy}} \quad (1)$$

Where $TRUST_{xy}$ denotes the trustworthiness of y according to x's point of view.

$\sum_{Arg \in I} FM\_agree(Arg_{xy})$ is the summation of the fuzzy measure of the degree of x's agreement over y's arguments that are acceptable to x; obtained from eq. (12).

$\sum_{Arg \in I} FM\_disagree(Arg_{xy})$ is the summation of the fuzzy measure of the degree of x's disagreement over y's arguments that are unacceptable to x; obtained from eq. (13).

$\sum_{R \in I} FM(R_{xy})$ is the summation of the fuzzy measure of the degree of x's satisfaction (or dissatisfaction) over a recommendation R by y for x; obtained from eq. (7).

$T\_N\_Arg_{xy}$ is the total number of arguments made by y towards x; a count maintained by the agent's persistent belief base in the system.

$T\_N\_R_{xy}$ is the total number of recommendations made by y for x; a count maintained by the agent's persistent belief base in the system.

All these arguments and recommendations are related to a particular domain. The basic idea is that the trust degree of an agent can be estimated according to how much information acquired from him has been accepted as belief in the past.

Particularly, in the argumentation-based recommendation, the accepted arguments capture the agent's reputation level. If some argument conflicts exist between the two agents, this will affect the confidence they have about each other. Because all the factors of Equation 1 are related to the past, this information number is finite. Trustworthiness is a dynamic characteristic that changes according to the interactions taking place between agent x and agent y. This assumes that x knows y. If not, or if the number of interactions is not sufficient to determine this trustworthiness, the consultation of other agents becomes necessary. Therefore, each agent has two kinds of beliefs when evaluating the trustworthiness of another agent: local beliefs and global beliefs. Local beliefs are based on the direct interactions between agents. Global beliefs are based on the combination of the different experiences of other agents that we call neighbors or witnesses. In our model, local beliefs to determine direct trust are given by Equation 1. A global belief about an agent or the indirect trust (reputation) measure is formulated by combining trust measures given by witnesses using equation 2 as explained below.

Let us suppose that an agent x wants to evaluate the trustworthiness of an agent y with whom he never (or not enough) interacted before. This agent must ask agents he knows to be trustworthy (we call these agents *neighbor agents*). To determine whether a neighbor agent p is confident or not, a trustworthiness threshold w must be fixed. Thus, p will be considered trustworthy by x iff $\mathrm{TRUST}_{xp} \geq w$. Agent x attributes a trustworthiness measure to each neighbor agent p. When neighbors are consulted by x, each neighbor agent p provides a trustworthiness value for agent y if p knows y. Neighbor agents use their local beliefs to assess this value (Equation 1) of trust. Thus, the problem consists in evaluating y's trustworthiness using the trustworthiness values transmitted by neighbor agents. The value $\mathrm{REPUTATION}_y$ given by equation 2, represents a fuzzy estimation of indirect trust that agent x have in agent y. This value is based on the fuzzy trust that x's N number of neighbors p have in agent y, and x's trust over its neighbor p. Also, an agent y is said to be trustworthy if the $\mathrm{TRUST}_{xy} \geq w$ or $\mathrm{REPUTATION}_y \geq w$

$$REPUTATION_y = \frac{\sum_{p=1toN} TRUST_{xp} \times TRUST_{py}}{\sum_{p=1toN} TRUST_{xp}} \tag{2}$$

4.2 The Fuzzy Model for Determining Trust Based on Interactions between Agents

Trust is a complex concept that constitutes many properties. Therefore to model it, we need to find the agreement (both individuals are satisfied to what extent) and disagreement (only one of them is unsatisfied to what extent) between two agents. To do so, we can define two fuzzy subsets on each agent's ratings, say satisfied and unsatisfied. This is because the fuzzy sets can clearly capture the concept of finding the extent (membership value) to which an agent is satisfied or unsatisfied with an interaction, i.e. argument or recommendation. The membership values of satisfied and unsatisfied fuzzy subsets for a given encounter always sum up to one,

for example, 65% satisfaction indicates 35% dissatisfaction. From these two fuzzy sets we can find the agreement and disagreement between the two agents. The satisfied and unsatisfied fuzzy subsets for agent x are defined as below:

$$\text{satisfied}(x) = \{\text{sat}_x(i_k) \mid i_k \in H_x\} \tag{3}$$

$$\text{unsatisfied}(x) = \{\text{unsat}_x(i_k) \mid i_k \in H_x\} \tag{4}$$

where $\text{sat}_x(i_k)$ and $\text{unsat}_x(i_k)$ are membership values of x's ratings for an interaction i_k in the fuzzy subsets satisfied(x) and unsatisfied(x), respectively. Now we give a simple triangle membership functions for satisfied(x) and unsatisfied(x) fuzzy subsets that are defined by the following popular equations [5]:

$$sat_x(i_k) = \begin{cases} 0 & r_{xy}(i_k) = g_{\min} \\ \dfrac{r_{xy}(i_k) - g_{\min}}{g_{\max} - g_{\min}} & g_{\min} < r_{xy}(i_k) < g_{\max} \\ 1 & r_{xy}(i_k) = g_{\max} \end{cases} \tag{5}$$

$$\text{unsat}_x(i_k) = 1 - \text{sat}_x(i_k) \tag{6}$$

where g_{min} and g_{max} are the minimum and the maximum possible ratings for a given system with $G = \{ g_{min}, \ldots, 0, \ldots, g_{max}\}$.

Therefore, to evaluate the $\sum_{R \in I} \text{FM}(R_{xy})$ component of eq. (1), we consider only those instances of $i_k \in R \subset I$, i.e. when the interaction between agents means a recommendation is communicated. Hence,

$$\sum_{R \in I} \text{FM}(R_{xy}) = \sum_{i_k \in R \subset I} \text{sat}_x(i_k) \text{ for a particular recommender agent y.} \tag{7}$$

The possible combinations of satisfied and unsatisfied fuzzy subsets define four values for any two agents [5]. For argumentation between agents in a recommender system we need to define the following two combinations only, i.e. satisfied-satisfied SS(x, y), and unsatisfied-satisfied US(x, y), assuming that agent x represents a user whereas agent y represents a recommender.

$$SS(x, y) = \frac{|satisfied(x) \cap satisfied(y)|}{|satisfied(x) \cup satisfied(y)|} \tag{8}$$

$$US(x, y) = \frac{|unsatisfied(x) \cap satisfied(y)|}{|unsatisfied(x) \cup satisfied(y)|} \tag{9}$$

Fuzzy sets literature describes many alternatives for union and intersection of crisp sets. The popular one is minimum for intersection and maximum for union. The min–max alternative and the definition of fuzzy set's cardinality as given by Zadeh in [11] yield the following:

$$| \text{satisfied}(x) \cap \text{satisfied}(y) | = \sum_{i_k \in H_x \cap H_y} \min(\text{sat}_x(i_k), \text{sat}_y(i_k)) \tag{10}$$

$$| \text{satisfied}(x) \cup \text{satisfied}(y) | = \sum_{i_k \in H_x \cap H_y} \max(\text{sat}_x(i_k), \text{sat}_y(i_k)) \tag{11}$$

Therefore, for a rating fuzzy trust system, the agreement and disagreement values over an argument between any two agents x and y are given by:

$$\text{agree}(\text{Arg}_{xy}) = \text{SS}(x, y) \tag{12}$$

$$\text{disagree}(\text{Arg}_{xy}) = \text{US}(x, y) \tag{13}$$

Now we can also express an implicit agent feedback mathematically in terms of $\text{agree}(\text{Arg}_{xy})$ and $\text{disagree}(\text{Arg}_{xy})$ after an argument as follows:

$$\text{feedback}(x, y) = (1 - \text{disagree}(\text{Arg}_{xy})) \times \text{agree}(\text{Arg}_{xy}) \tag{14}$$

The mutual exchange of arguments is represented by $\text{agree}(\text{Arg}_{xy})$ value (both agents are satisfied). On the other hand, $\text{disagree}(\text{Arg}_{xy})$ value represents the conflict both agents have. The feedback becomes positively high if the disagreement is low and vice versa. Accordingly, formula gives disagreement more weightage, i.e. when the $\text{disagree}(\text{Arg}_{xy})$ becomes one, the feedback value is zero whatever be the $\text{agree}(\text{Arg}_{xy})$ value. Hence, equation (14) can be used for the agent feedback mechanism.

5 An Illustration to Estimate an Agent's Trustworthiness

We now briefly describe the concept with help of a walk-through example to show the calculation of trust values based on agent interactions. We consider a scenario, in which an agent x needs to determine the better recommender agent out of the two agents, say: agent y and agent z. The example shows how the agent x uses the trust and the reputation values based on interactions (of both types: arguments and recommendations) to estimate the trustworthiness of other agents.

Refer table 1 for the calculations. The agent uses equations (1) to (13) for its calculation of trust and reputation. To calculate trust it uses its own direct experiences or interactions with agent y and agent z. Since the number of interactions in the case of agent z are not enough, therefore agent x's trust in agent z also falls below the threshold w.

In such cases agent x consults its neighbors whom it trusts and their trust value is greater than or equal to the threshold w. Then it calculates the reputation of agent z. Though, the reputation of agent z is greater than threshold w but still it falls lesser than the direct trust of agent x in agent y. Direct trust score in an agent can override the reputation score of other agent in cases where direct trust is stronger than the indirect trust. Hence, agent x considers agent y to be more trustworthy than agent z.

Table 1. Computation of a Target Agent's Trustworthiness

TRUST SCORE FOR Interactions (I) >= 5 and THRESHOLD *w* >= 0.3						
Target agent	***Total no. of interactions (I)***	***I-type***	***Sum(FM_agree(Arg))***	***Sum(FM_disagree(Arg))***	***Sum(FM(R))***	***Trust score***
Y	4	argument	0.8439	0.4313	----	0.5543
Y	2	recommendation	----	----	0.8503	
Z	2	argument	0.4232	0.3222	----	0.1723; less value for I, w
Z	1	recommendation	----	----	0.2139	
REPUTATION SCORE FOR I < 5 and THRESHOLD w >= 0.3						
Target agent	***Neighbors***	***Trust in Z***	***Trust in neighbor***	***Reputation score***		
Z	neighbor1	0.3241	0.5493	0.3139		
Z	neighbor2	0.5451	0.3215			
Z	neighbor3	0.2119	0.7839			
Result: Agent x determines Agent y to be more trustworthy than Agent z because TRUSTxy > REPUTATIONz						

6 Conclusion and Future Work

A fuzzy computational model for agent-based recommender systems is proposed for reputation and trust in this paper. The proposed model can be implemented for an application which is argument-based and use ratings explicitly or implicitly. Our model enables a personal agent to identify agreements, conflicts with other agents and use the information to determine whether to trust them or not. Therefore, the resulting model allows us to produce an assessment of the agents' credibility in an argumentation-based recommendation setting.

As part of our future work, we intend to extend our present model for trust and reputation, in order to provide the rationale behind trusting decisions by reasoning about them using argumentation. This would further enhance the recommendation quality in RS.

References

[1] Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., Tennenholtz, M.: Trust-Based Recommendation Systems: an Axiomatic Approach, WWW 2008, Refereed Track: Internet Monetization - Recommendation & Security (April 21-25, 2008)

[2] Bedi, P., Vashisth, P.: Interest Based Recommendations with Argumentation. Journal of Artificial Intelligence, ANSI, 119–142 (2011)

[3] Bedi, P., Vashisth, P.: Social-cognitive trust integrated in agents for E-commerce. In: Proceedings of the 2nd IC4E, Mumbai, India, January 7-9, pp. 1–11 (2011)

[4] Bentahar, J., Meyer, J.J.C.: A New Quantitative Trust Model for Negotiating Agents using Argumentation. International Journal of Computer Science & Applications IV(II), 1–21 (2006)

[5] Bharadwaj, K.K., Al-Shamri, M.Y.H.: Fuzzy computational models for trust and reputation systems. Electron. Comm. Res. Appl. (2008), doi:10.1016/j.elerap.2008.08.001

[6] Chesnevar, C., Maguitman, A.G., Gonzalez, M.P.: Empowering Recommendation Technologies Through Argumentation. In: Argumentation in Artificial Intelligence, p. 504. Springer, Heidelberg (2009) ISBN-13: 978-0387981963

[7] Parsons, S., Tang, Y., Sklar, E., McBurney, P., Cai, K.: Argumentation-based reasoning in agents with varying degrees of trust. In: Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS, pp. 879–886 (2011)

[8] Stranders, R., de Weerdt, M., Witteveen, C.: Fuzzy Argumentation for Trust. In: Sadri, F., Satoh, K. (eds.) CLIMA VIII 2007. LNCS (LNAI), vol. 5056, pp. 214–230. Springer, Heidelberg (2008)

[9] Tang, Y., Cai, K., Sklar, E., McBurney, P., Parsons, S.: A system of argumentation for reasoning about trust. In: Proceedings of the 8th European Workshop on Multi-Agent Systems, Paris, France (2010)

[10] Wei, Z.: A Novel Trust Model Based on Recommendation for E-commerce. IEEE (2007) 1-4244-0885-7

[11] Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computer and Mathematics with Applications 9(1), 149–184 (1983)

Artificial Bee Colony Algorithm with Uniform Mutation

Amit Singh[1], Neetesh Gupta[2], and Amit Sinhal[3]

[1] (IT Dept.)
[2] (IT Dept.)
[3] (CSE Dept.)
{amitrajput3,gupta.neetesh81}@yahoo.co.in,
amit_sinhal@rediffmail.com

Abstract. Swarm intelligence systems are typically made up of a population of simple agents or bodies interacting locally with one another and with their environment. Particle swarm, Ant colony, Bee colony are examples of swarm intelligence. In the field of computer science and operations research, Artificial Bee Colony Algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm. In this work, we present an extended form of ABC algorithm, where we have added mutation operator of Genetic algorithm (GA) in the classical ABC algorithm. We have used mutation operator of real coded genetic algorithm. We have added mutation operator after the employed bee phase of ABC algorithm. This paper presents some experiments on ABC with real coded mutation operator on benchmark optimization problems.

Keywords: Artificial Bee Colony (ABC), Genetic Algorithm (GA), Mutation.

1 Introduction

The problem of optimization is the most crucial problem in today's era and a great many work have been done to solve it. Previously a lot of work has been done on GA, ABC and hybridization of various evolutionary algorithms. There are few literatures available which compares their performance evaluation and suggests the best technique to be opted for specific problem.

Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervis Karaboga in 2005 [1], motivated by the intelligent behavior of honey bees. It is as simple as Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms, and uses only common control parameters such as colony size and maximum cycle number. ABC as an optimization tool provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 503–511.
springerlink.com

In this Paper, we have extended the classical Artificial Bee Colony algorithm to the area of optimization problem. Proposed method basically adds an additional step of mutation operator in the Artificial Bee Colony for finding out the optimality. To validate the performance of proposed method four benchmark functions is used in our experiment.

The organization of the paper is as follows section 2 gives brief introduction on Artificial Bee Colony algorithm. Genetic mutation is explained in section 3, Section 4 explain Proposed Methodology. Section 5 describes the Experiment Setup and Benchmark functions, Experimental Results are given in section 6. Section 7 gives Conclusion.

2 Artificial Bee Colony

In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source is called an onlooker and a bee going to the food source visited by it previously is named an employed bee. A bee carrying out random search is called a scout. In the ABC algorithm, first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. The main steps of the algorithm are given below:

- Initialize.
- REPEAT.

(a) Place the employed bees on the food sources in the memory;

(b) Place the onlooker bees on the food sources in the memory;

(c) Send the scouts to the search area for discovering new food sources.

- UNTIL (requirements are met).

In the ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts; selecting of the food sources by the onlookers after sharing the information of employed bees and determining the nectar amount of the foods; determining the scout bees and then sending them onto possible food sources. At the initialization stage, a set of food source positions are randomly selected by the bees and their nectar amounts are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. At the second stage, after sharing the information, every employed bee goes to the food source area visited by her at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of

visual information in the neighborhood of the present one. At the third stage, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. As the nectar amount of a food source increases, the probability with which that food source is chosen by an onlooker increases, too. Hence, the dance of employed bees carrying higher nectar recruits the onlookers for the food source areas with higher nectar amount. After arriving at the selected area, she chooses a new food source in the neighborhood of the one in the memory depending on visual information. Visual information is based on the comparison of food source positions. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one.

In ABC algorithm each food source position represents a candidate solution of optimization problem. In optimization problem each solution is associated with the fitness value on the basis of fitness value it is decided that which solution is better. So the nectar amount of a food source corresponds to the fitness value of the associated solution in ABC algorithm. The number of employed bees or the onlooker bees is equal to the number of solutions in the population. The ABC algorithm generates a random solution or initial population of size NF, where NF denotes the size of population or total number of food source. Each solution is represents the position of food source and denoted as x_{ij}, where i represents a particular solution (i=1,2,…..,NF) and each solution is a D-dimensional vector so j represents a particular dimension of a particular solution (j=1,2,….,D). After initialization of random solution employed bees start their searching. Employed bees search the food source near the previous food source, if the generated new solution is better than the previous solution than new solution replaces the old one. The comparison of food sources or solutions is done on the basis of fitness value or nectar amount of food source.

After all employed bees complete the search process; they share the nectar information of food sources (solutions) and their position information with onlooker bees. Now onlooker bee chooses a food source depending on the probability value P_i associated with the food source. Probability value for each food source is calculated by following equation (1):

$$p_i = \frac{f_i}{\sum_{n=1}^{NF} \text{fn}} \tag{1}$$

Where is the fitness value of the solution *i* or the nectar amount of food source evaluated by employed bee and *NF* is the number of food source. So after the evaluation of the food source by the employed bees the probability value for each food source is determined which is used by onlooker bees.

To produce the candidate solution from the previous solution artificial bee uses the following equation (2):

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \tag{2}$$

Where j is a index for dimension ($j=1,2,....,D$), k is a index which represents particular individual or solution from the population ($k=1,2,3.....,NF$), and i is also a index represents a particular solution ($i=1,2,.....,NF$). The difference between i and k is that k is determined randomly and value of k has to be different from i. $\emptyset_{ij}$ is a random number between [-1,1]. It controls the production of the neighbor food positions around $x_{ij.}$. The difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is reduced. After the production of candidate solution v_{ij} , its fitness value is calculated and then it is compared with the fitness of x_{ij}. If the new candidate solution has equal or better nectar or fitness than the old, it is replaced with the old one in the memory. Otherwise, the old is retained. If a solution is not improved further through a predetermined number of cycles then that food source is assumed to be exhausted. Exhausted food source is replaced by new food source generated by scout bees.

3 Genetic Mutation

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

In genetic algorithms of computing, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is analogous to biological mutation. The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified.

Mutation is the unary operator which changes genes in a chromosome. Various mutation operators such as uniform mutation [12], real number creep mutation [6], dynamic mutation [13], and Miihlenbein mutation [11], etc., have been provided in RCGA. In this work we have used real coded uniform mutation In uniform mutation, we randomly selects one gene x_{ij} and sets its value equal to a uniform random number in $[lb_j, ub_j]$.

4 Artificial Bee Colony with Mutation

In this proposed method, one additional step is added to standard Artificial Bee Colony Optimization is of mutation operator. Mutation operator is added after the employed bee phase of Artificial bee colony algorithm. ABC algorithm has four

phases initialization phase, employed bees phase, onlooker bees phase and scout bees phase, we add mutation phase after the employed bee phase. Employed bee phase do the local search and mutation after the employed bee phase explore the search space and do the searching new area of solution space. Through mutation, on the one side, there is a chance of changing the local best position, and the algorithm may not be trapped into local optima. On the other side, individual can make use of the others' advantage by sharing information mechanism. In this method, the mutation step is carried out on the probabilistic way in each food searching operation for each iteration during the life cycle of ABC optimization technique. Selection of food source is done in a random manner. Food Source is selected arbitrarily from the food size and mutation is performed. In mutation, generated offspring's replaces the older offspring's. The mutation operator used in this paper is uniform mutation. When performing mutation, we randomly select one food source x_{ij} and replace its one of the dimension value by random number generated in between lower and upper bound value of the food source.

Below the proposed ABC with mutation after employed bee phase is shown. First phase is initialization phase where individuals are randomly selected from the search space.

Algorithm 1: ABC with Mutation after Employed Bee Phase
[Initialization Phase]
for i=0 to max number of Food source NF **do**
for d=0 to dimension size **do** Randomly initialize food source positions Xij
end for d Compute fitness of each food source
end for i
Repeat [Employed Bee Phase]
for i=0 to max no of employed bee **do**
for d= 0 to dimension **do** produce new candidate solution
end for d Compute fitness of individual
if fitness of new candidate solution is better **than** the existing solution replace the older solution.
end for i
for i = 0 to max number of food source NF **do** Calculate the probability for each food source.
end for i
[Mutation Phase] if mutation criteria is met **then**

Select random particle from current population for mutation operator. Apply mutation operation to generate new individuals new offspring generated from the result of mutation. New set of sequence is generated for offspring compute the cost for that offspring compute the fitness of updated individual

[Onlooker Bee Phase]
for i=0 to max no of onlooker bee **do** choose food source on the basis of probability Pi

for d= 0 to dimension **do** produce new candidate solution for food source position Xij
end for d compute fitness of individual food source
if fitness of new candidate solution is better **than** the existing solution replace the older solution.
end for i
[Scout Bee Phase] If any food source exhausted **than**
replace it by new random position generated by scout bee.
Memorize the best food source so far
Until (Stopping criteria met).

Next is employed bee phase where local search is performed by employed bees. Next we have added additional operator to the ABC algorithm to check whether it improves the performance of algorithm. We have added a mutation operator after employed bee. Mutation is performed on the individual food source or individual if mutation probability satisfied. Onlooker phase and scout bee phase performed on current population or food sources after the mutation phase. At last best food source or individual is considered as global best solution.

5 Parameter Setup for ABC with Mutation and Benchmark Functions

5.1 Benchmark Functions

From the standard set of benchmark problems available in the literature, four important functions are considered to test the efficacy of the proposed method. All the problems are composed of continuous variables and have different degree of complexity and multi-modality. The set of test functions include unimodal and multimodal functions which are scalable (the problem size can be varied as per the user's choice). In our experiments, problem size for all problems is set to 30. These are minimization problems having a minimum at 0. The problems are listed in Table 1.

5.2 Parameter Setup for ABC with Mutation

For every algorithm there are some control parameters which are used for its efficient working. Hence, there are some control parameters for Bee Colony with Mutation Algorithm also. We did an extensive literature survey and carried out our own experiments for determining the values of these control parameters. From this we found that the values which we have taken in this experiment are standard values and they are also suitable for this experiment.

Table 1. Benchmark problems

Function Name	Function	Search Space
Sphere	$\text{Min} f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \leq x_i \leq 5.12$
Griewank	$\text{Min} f(x) = 1 \mid \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \pi_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$600 \leq x_i \leq 600$
Rosenbrock	$\text{Min} f(x) = \sum_{i=1}^{n-1} [100 \quad (x_{i+1} \quad x_i^2)^2 \mid (x_i \quad 1)^2]$	$30 \leq x_i \leq 30$
Rastrigin	$\text{Min} f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	$-5.12 \leq x_i \leq 5.12$

The first control Parameter is Maximum cycle number and the value of this parameter we have taken in our experiment as 2500. The next parameter in our experiment is maximum number of food source and we have taken its value to be 30. Another control parameter is number of runs and we have taken its value in our experiment as 30. It must be noted that each run contains maximum cycle number, which is 2500 in our experiment. The fourth control parameter is Dimension and its value is taken as 30. In this experiment we are using the feature of uniform mutation operator in the ABC with mutation algorithm. The control parameter for mutation operator is Probability. Therefore we need to find the value of this parameter also. Its value can range from 0.1 to 0.9 we have taken as 0.7 for the mutation.

Table 2. Experiment Result for ABC and Proposed ABC with Mutation after Employed Bee Phase of ABC

Benchmark Function	**ABC**	**ABC with Mutation**								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Sphere	1.1402e-15	1.1739e-15	**1.1251e-15**	1.1493e-15	1.2187e-15	1.1573e-15	**1.0353e-15**	1.2568e-15	**1.0930e-15**	1.1975e-15
Rosenbrock	1.2651e	**1.2627e**	1.3612e	2.2461e	1.7602e	1.2801e	2.3009	1.9934	2.3042e	1.7346e
Griewank	1.1102e-15	**1.0880e-15**	**1.0695e-15**	1.1139e-15	1.1176e-15	**1.0510e-15**	1.2323e-15	**1.0325e-15**	**1.0140e-15**	**1.1028e-15**
Rastrigin	1.5889e-3	**6.0197e-5**	**1.346e-5**	3.320e-2	**7.497e-4**	2.2981e-3	6.6397e-2	**2.1956e-4**	3.3554e-2	3.3907e-2

6 Experimental Results

In this section we analyze the result obtained by our algorithm. To test the efficiency of our algorithm results of ABC with Mutation is compared with classical ABC algorithm results. Table II shows the experiment result for ABC and ABC with mutation operators after the employed bee phase of ABC algorithm. Results show the mean fitness values calculated by ABC and ABC with mutation operator for four benchmark function sphere, rosenbrock, griewank and rastrigin. From the result table II it is clear that our proposed algorithm outperforms the original ABC algorithm. Our proposed algorithm performs better than the ABC algorithm and there is no fixed value for the mutation probability for which algorithm performs good for all function. We have used mutation probability from 0.1 to 0.9 as shown in table II and it is clear from table that different there is no such probability value for which all function has optimum result.

7 Conclusion

In this paper, real coded mutation operator is applied to the ABC after the employed bee phase of ABC algorithm. In iteration, with some probabilistic criteria selected food source is altered by mutation operator. The experiments are performed on a set of four benchmark problems available in the literature. There is no specific value for mutation probability for which we can obtain best results for these benchmark problems. As future work we have the intention to apply other types of mutation operators in the ABC algorithm.

References

[1] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim., 459–471 (2007)

[2] Silvaa, C.A., Sousaa, J.M.C., Runkler, T.A.: Rescheduling and optimization of logistic processes using GA and ACO. Engineering Applications of Artificial Intelligence 21, 343–352 (2008)

[3] Wong, L.-P., Hean Low, M.Y., Chong, C.S.: A Bee Colony Optimization Algorithm for Traveling Salesman Problem. In: Second Asia International Conference on Modelling & Simulation, pp. 818–823 (2008)

[4] Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, Technical Report-TR06 (2005)

[5] Karaboga, D., Akay, B.: A Survey: Algorithms Simulating Bee Swarm Intelligence. Artificial Intelligence Review 31, 68–85 (2009)

[6] Jin, G.-G., Joo, S.-R.: A Study on a Real-Coded Genetic Algorithm. Journal of Control, Automation, and Systems Engineering 6(4), 268–274 (2000)

[7] Biesmeijer, J.C., Seeley, T.D.: The use of waggle dance information by honey bees throughout their foraging careers. Behav. Ecol. Sociobiol. 59, 133–142 (2005)
[8] Wright, A.: Genetic Algorithms for Real Parameter Optimization. In: Rswlins, G. (ed.) Foundations of Genetic Algorithms, pp. 205–218. Morgen Kaufmann publishers, CA (1991)
[9] Davis, L.: Handbook of Genetic Algorithms. Von Nostrand Reinhold, N.Y (1991)
[10] Janikow, C.Z., Michalewicz, Z.: An Experimental Comparisonof Binary and Floating Point Representations in Genetic Algorithms. In: Proc. 4th Int. Conf. on Genetic Algorithm. Morgan Kaufmann publishers, CA (1991)
[11] Miihlenbein, H., Schlierkamp-Voosen, D.: Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization. Evolutionary Computation 1, 25–49 (1993)
[12] Michalewicz, Z.: Genetic algorithm + data structure = evolution pmgram. Springer-Verlag, Inc., Heidelberg (1996)

An Efficient Framework Using Fuzzy Logic and Multi Agent System for Cellular Network

Megha Kamble and Roopam Gupta

Dept. of Information Technology, U.I.T., RGTU, Bhopal
meghakamble@gmail.com, roopamgupta@rgtu.net

Abstract. In this paper, we present a multi agent system for resource allocation problem in cellular network. A multi agent system based approach is proposed by taking into consideration the concurrent nature of services and involvement of number of participants in cellular network. This work emphasizes on enhancement of QoS using fundamental mechanism of call admission control as well as resource allocation as a additional measure. The Quality of Service (QoS) parameters are often measured in terms of call dropping probability, call blocking probability and failure rate. To meet this achievement, this work proposes soft computing approach for implementation of multi agent system to increase system capacity to accommodate the calls and to reduce failure rate. The results are produced using simulation model for microcellular network for two different traffic scenarios and fuzzy approach has shown the enhancement.

Keywords: Cellular Network, Fuzzy Logic Control, Multi Agent System, Defuzzification, Failure Rate, Uniform and Nonuniform Traffic.

1 Introduction

We consider in a cellular mobile network, the geographical area served by the system is divided into several hexagonal-shaped cells. The mobile hosts in each cell are served by a base station (BS) located in the cell and the BSs are interconnected via a wired network. A mobile host can communicate only with the base station in its cell directly. For this communication, the available wireless bandwidth is divided into channels, where each channel is capable of supporting a communication session. The call can be set up only if a channel is assigned to support the communication between the mobile host and the base station. Central pool of channels is available at Mobile Switching Center (MSC). No two cells in the system can use the same channel at the same time if they are within minimum channel reuse distance; otherwise, channel interference will occur. So channel allocation is a key problem in cellular network[9].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 513–523.
springerlink.com

1.1 Channel Allocation Approach

There are different strategies for channel allocation in Cellular Networks such as fixed channel allocation and dynamic channel allocation provided in literature[9-12]. Fixed and dynamic approach can be combined to form hybrid channel allocation strategy. This paper proposes soft computing approach to implement hybrid channel allocation over the traditional approach. Distributed nature of approach tries to manage the radio resources of individual cell(local scenario) and tries to provide free resources from heavily loaded regions to less loaded regions using fuzzy control. Thus helps in maintaining QoS for non uniform traffic scenario.

1.2 Agent Structure

Figure 1 in [4][5] shows the layered architecture adapted for Multi Agent system and also elaborates functional architecture. This paper has presented details of fuzzy logic implementation at reactive layer and related experimental results. Novelty of the work lies in redesigning the fuzzy logic parameters[11] and using defuzzification formulae for call buffering parameters.

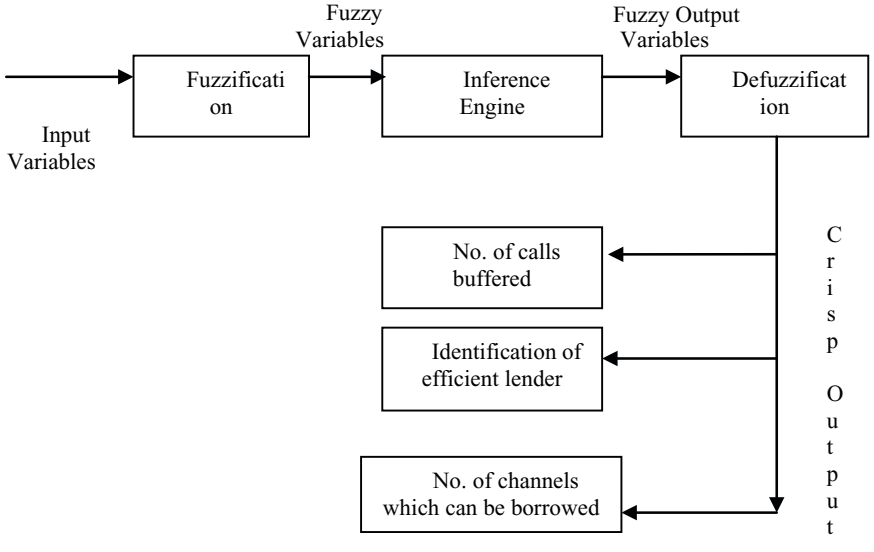


Fig. 1. Fuzzy CAC System in Reactive Layer (MAS FRAMEWORK)

1.3 Related Work

Parag Pendharkar[3] had implemented MAS using DCA and the results can be improved using HCA in place of DCA. E.L. Bodanase[2] had presented one of the early research about MAS in cellular network. Yao-Tien Wang [11] had implemented fuzzy logic for DCA. The framework presented in the literature[1]

was missing call admission decision at entry level so as to enhance QoS. Another is application of HCA in local planning layer which will improve the QoS.

2 Proposed Framework

2.1 Fuzzy CAC System in Reactive Layer

Fuzzy CAC system is comprising of three blocks as shown in Figure 1. Fuzzification is initial level. Inference Engine is decision making and defuzzification gives crisp output.

2.2 Fuzzification

Fuzzifier initiates the process of fuzzification that is representation of uncertainties using fuzzy sets, A fuzzy set A is represented through use of support of a fuzzy set. The support of a fuzzy set A is the crisp set of all x $\in$ U such that $\mu_x(x) > 0$, where U is the universe of discourse. The function is represented using notation $A=\int_{\mu} \mu_A(x)/x$ and is defined on the interval $[a_0,a_6]$.

This work implements fuzzification of following variables as shown in the sample figures Figure 2(a) and 2(b).

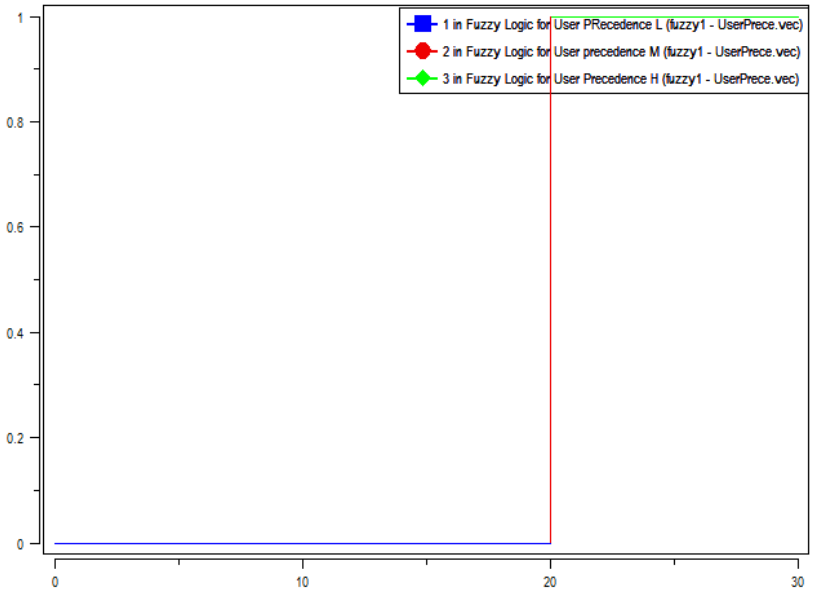


Fig. 2(a). Membership Function for User Precedence defined over class interval (0..30)

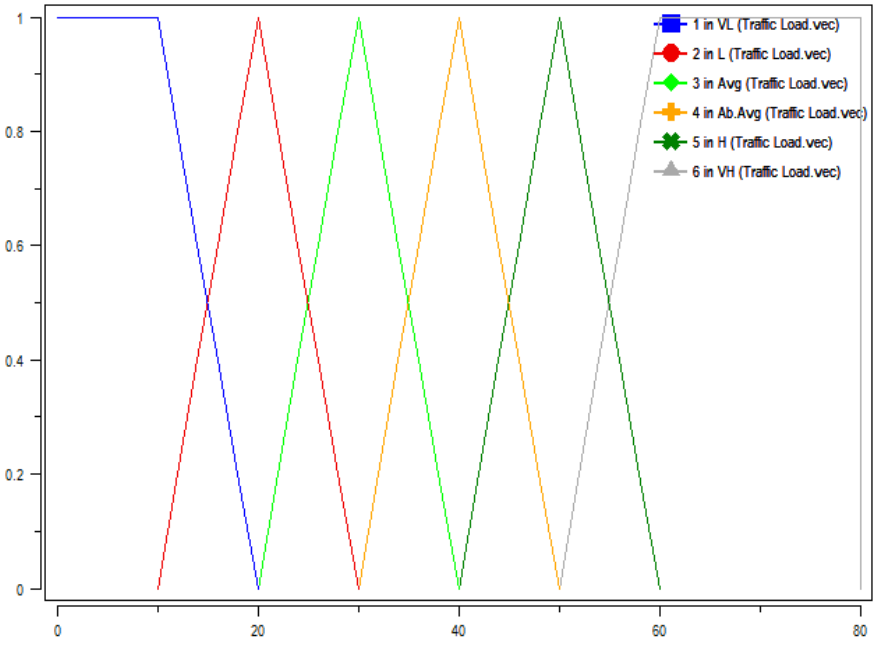


Fig. 2(b) Membership Function for Traffic Load defined over (0..80)

- Network load is defined as all the admitted users in a cell over the average number of channels available for data traffic.
- Traffic load is calls per hour in a cell, can also be represented in Erlangs.
- Dynamic channel requirement can be determined using Erlang B formula, for specific traffic load.
- User Precedence is priority of the call, highest for call in progress, medium for the originating call and low for call about to terminate.

General form of fuzzy control rules for multi-input-single-output system[7] is followed.

Input: R1: If x is A1 and y is B1 and z is C1 and w is D1, then decision is Output O.

x,y,z,w are linguistic variables represented above, which are involved in decision making. O is conclusion, mapped to linguistic values Strongly Accept(SA), Weakly Accept and buffer(WAB), Weak Rejection and buffer(WRB), Strongly Reject(SR).

2.3 Defuzzification

The purpose of defuzzification is to convert each result obtained from the inference engine, which is expressed in terms of fuzzy sets, to a single real number. This process is necessary because in many practical application crisp

controls action is required for the actual control. Let c represents class of fuzzy control for Fuzzy Output Variable, i represents count for number of rules n, μi represents degree of membership evaluated for rule i. The objective of the proposed work is determining crisp output using Mean of Maxima (MOM) method, MOM c,I is determined for every class using the formula: MOM= ½(a_0+a_6) for a class c over the interval [a_0,a_6].

This value of Crisp output is calculated using the equation:

$$f(x) = \sum_{i=1,c=1}^{n,CMAX} \mu i * MOM\ c,i \ / \sum_{i}^{n} \mu i \tag{1}$$

The proposed work evaluates f(x) for two outputs : i) Call admission decision – Call Accepted or Buffered (if buffered, number of calls to be buffered) ii) Call buffered delay in time units(seconds).

Local planning layer of the framework is working with the help of hybrid channel allocation Existing literature works for borrowing using either search or update method. This work proposes identification of promising neighbor that is effective lender with the help of soft computing approach where heuristic will be distance between the cells[10].

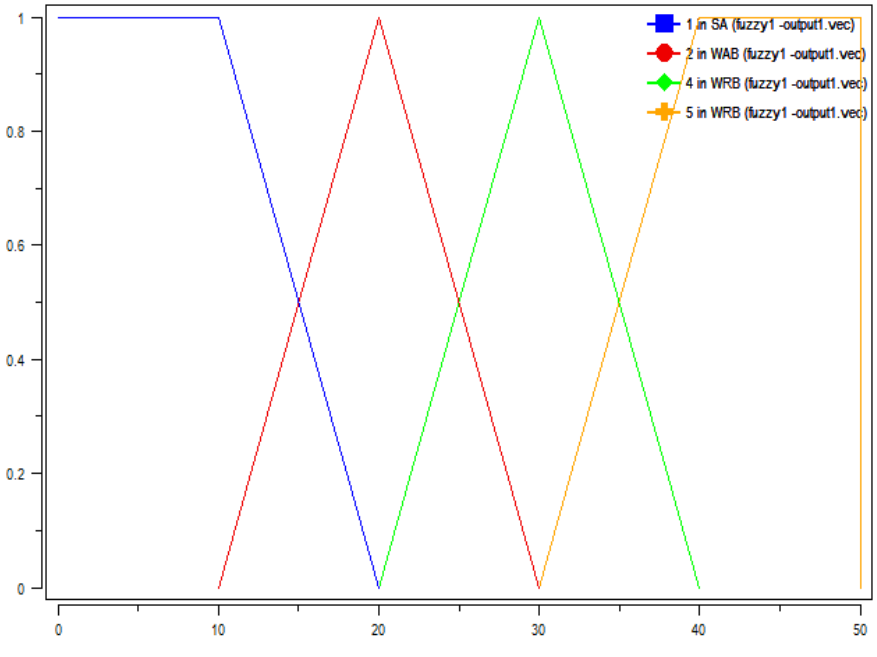


Fig. 3(a). Membership Function for Fuzzy Output for Number of Calls that can be buffered

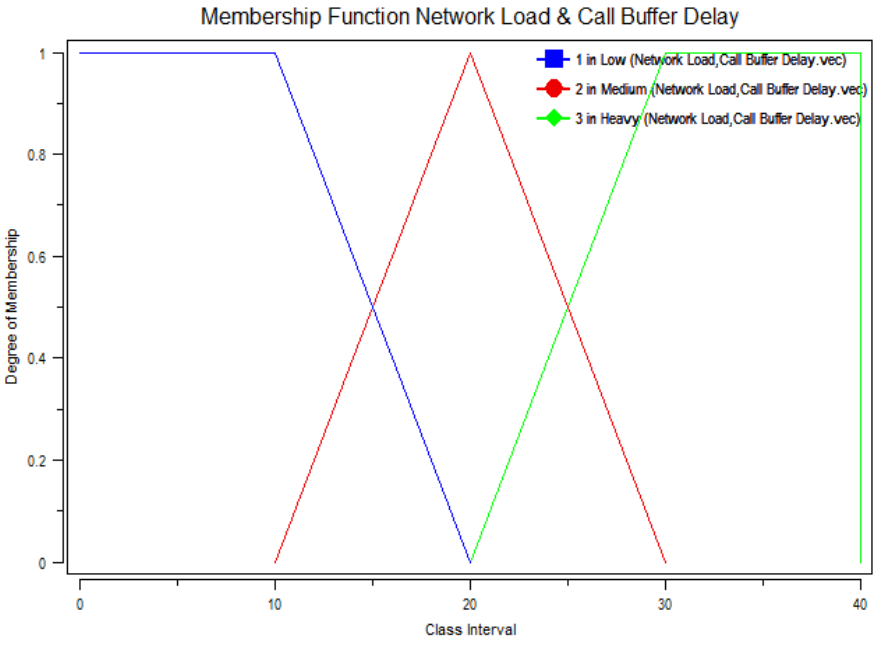


Fig. 3(b). Membership Function for Fuzzy Output for Call Buffer Delay as well as Input Variable Network Load

3 Simulation Experiment

The simulation parameters used in this paper are quite similar to the ones used in [4]. The simulated cellular network consists of a 2D structure of 7 x 7 hexagonal cells - a wrapped-around layout with cells, with each cell having six neighbors. Model is representing microcellular network[12] which is like a city model spread over area of 1 km^2. This model represents some areas with heavy traffic during peak hours. Two types of traffic distribution are considered, Uniform traffic scenario where all calls are arriving uniformly and second is nonuniform traffic scenario where hot-spots and cold-spots are created randomly. There are 70 channels in total in the system. A frequency reuse factor of 3 is assumed (i.e., $N = 3$).

The arrival of calls at any cell is assumed to be a Poisson process and that the call duration is exponentially distributed with a mean of 3 minutes.

The arrival of new calls is a Poisson process. The lifetime (length) of a new call is decided from an exponential distribution with an average of 180 seconds; its initial service cell is randomly chosen.

Nonuniform traffic scenario assumes 3 times mean arrival rate as compared to uniform traffic scenario. Cell is always in one of the two states. Every 3 min.s hot state changes to cold state and cell remains in cold state for about 30 mins. Average calls per hr are considered to be 200-500.

3.1 Numerical Results

Tabulated Statistics is given after application of Fuzzy Logic in Reactive Layer. The data is provided for values- Traffic Load 380 calls/hr, Network Load 8.03, Dynamic Channel Requirement 12.2 and User Precedence 0 (originating call).

Sample Fuzzy Rules and execution

R1:IF NW is Medium and TL is very low and DYN is low and UP is high THEN admission decision is SA
R2:IF NW is Medium and TL is low and DYN is low and UP is high THEN admission decision is SA
R3:IF NW is Medium and TL is avg and DYN is low and UP is high THEN admission decision is SA
R4:IF NW is Medium and TL is ab. Avg and DYN is low and UP is high THEN admission decision is WAB
R5:IF NW is Medium and TL is High and DYN is low and UP is high THEN admission decision is WRB
R6:IF NW is Medium and TL is Very High and DYN is low and UP is high THEN admission decision is WRB

R1 is executed as min(Ai(xi) ^ Bi(yi) ^ Ci(zi) ^ Di(wi))

Equation (1) gives crisp output for number of calls to be buffered instead of rejected using membership function shown in Fig.3(b). Equation(1) gives crisp output for average buffering delay for the calls using membership function shown in Fig.3(a).

The real number values are tabulated and compared with without fuzzy CAC approach. The results are showing enhancement in failure rate.

Call blocking probability[8] holds by the equation

$$Rb = \frac{no.of\ calls\ blocked}{total\ no.of\ calls} \tag{2}$$

Call dropping probability[12] is given by the following equation

$$Rd = \frac{no.of\ calls\ dropped}{total\ no.of\ calls - no.of\ blocked\ calls} \tag{3}$$

Based on the above two points, a performance index that is failure rate[9] can be derived which indicates the QoS of the network. The equation holds as

$$failure\ rate = Rb + (1 - Rd) * Rb \tag{4}$$

Average Call Buffer Delay which is calculated is 25-26 secs for both uniform and nonuniform traffic distribution.

Table 1. Membership Grades Contributing to the fuzzy output Admission Decision, which resulted from membership functions Fig.2(a)-2(d) and Fig. 5

Rules	SA C1,μ1	WAB C2,μ2	WRB C3,μ3	WA C4,μ4
R1:min(0.0,0.0,0.0,1)	0.0			
R2: min(0.1,0.1,0.7,1)	0.1			
R3: min(0.0,0.0,0.2,1)	0.0			
R4: min(0.13,0.4,0.6,1)		0.13		
R5: min(0.0,0.4,0.0,1)		0.0		
R6: min(0.7,0.8,0.9,1)			0.7	
R7: min(0.7,0.8,0.1,1)				0.1

Simulation model is implemented using Omnet++ Simulator. Uniform traffic scenario is considering equal distribution of calls in all the cells where as nonuniform traffic scenario generates heavy traffic areas(hot spots, generally when call arrival is greater that 45 calls per min.). Failure rate is determined by changing the ratio of channel allocation for hybrid channel allocation. 21:49 and 49:21 are the two ratios for static and dynamic channels available in the scenario.

Graphical analysis is showing that marginal enhancement is seen in uniform traffic and 21:49 ratio. 49:21 ratio produces better results in reduction of failure rate in uniform as well as nonuniform scenario.

Table 2 shows statistical analysis for traffic 200-500 calls/ hr , number of calls which can be buffered for average delay by application of Fuzzy CAC system in reactive layer. This data results into reduction of failure rate.

Table 2. Failure Rate without and with Fuzzy Approach

Traffic Load	Call rejected with FCA and DCA 21:49	% Failure rate without fuzzy	%Failure Rate with Fuzzy	Calls buffered
200	125	2.46	2.10	18.2
220	145	2.85	2.46	20
240	165	3.24	2.82	20
260	185	3.63	3.28	20
280	205	4.02	4.01	20
300	223	4.41	4.28	20
320	243	4.79	4.78	19.7
340	261	5.17	4.88	19.4
360	287	5.56	5.17	19.1
380	305	5.94	5.27	18.8
400	317	6.32	5.90	18.6
420	347	6.70	6.33	17.0
440	366	7.08	6.34	18.1
460	405	7.46	7.09	18.3
480	425	7.84	7.47	17.0
500	445	8.21	8.06	17.5

3.2 Graphical Analysis

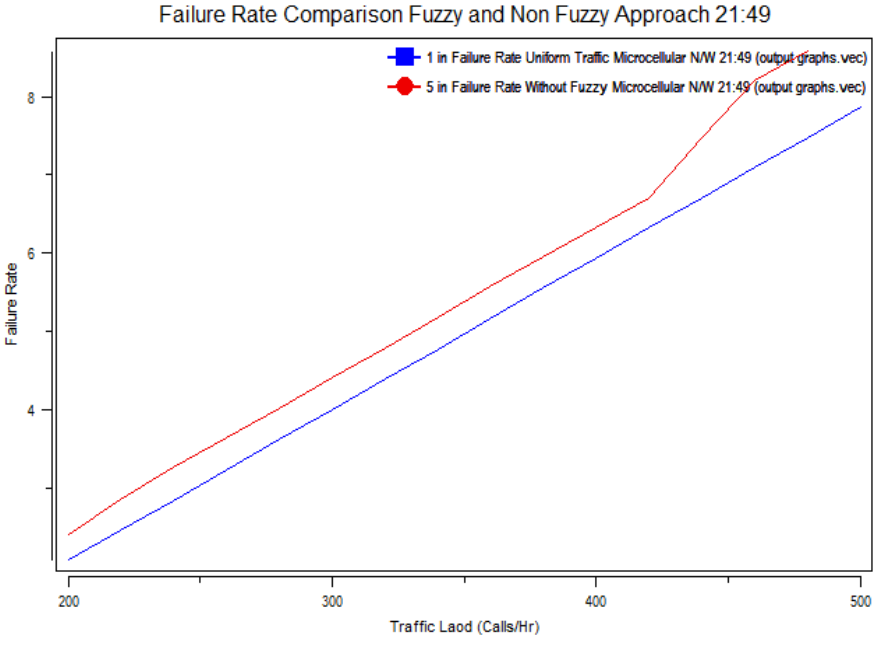


Fig. 4. Failure Rate Comparison Uniform Traffic Scenario 21:49 With and without Fuzzy Logic Approach

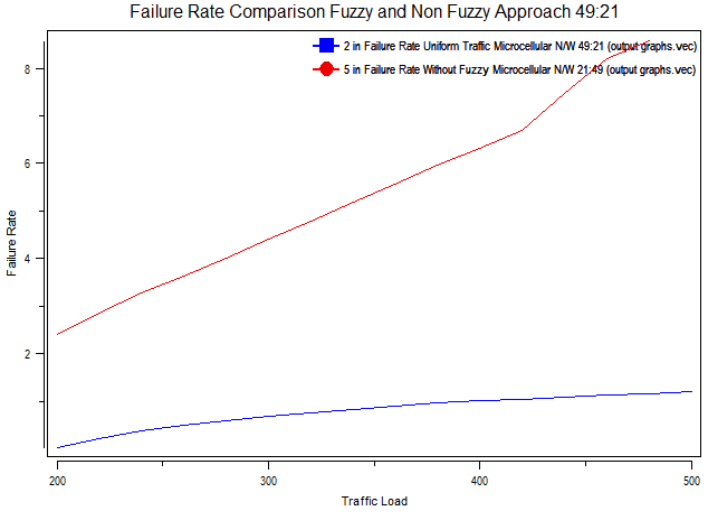


Fig. 5. Failure Rate Comparison Uniform Traffic Scenario 49:21 With and without Fuzzy Logic Approach

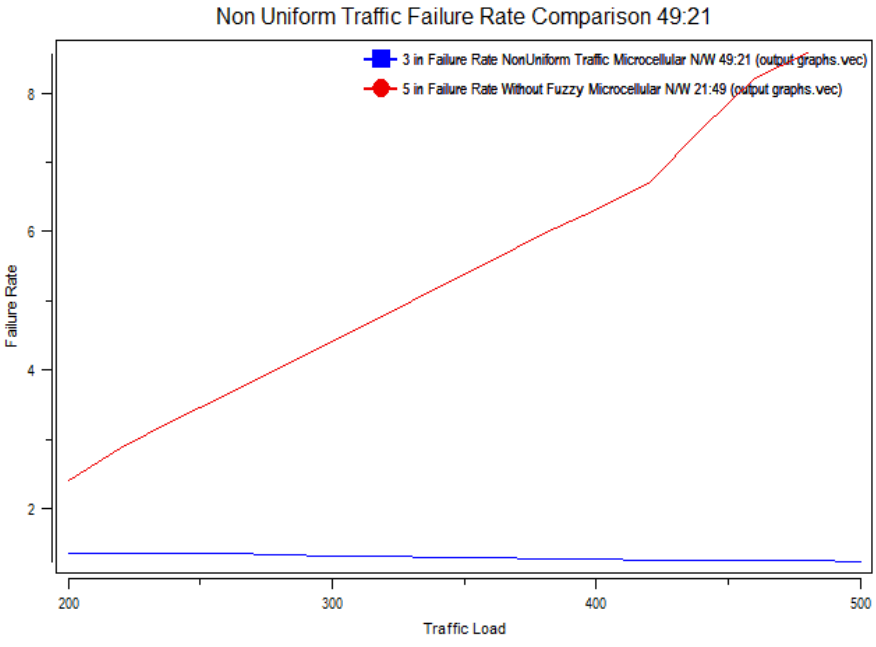


Fig. 6. Failure Rate Comparison Nonuniform Traffic Scenario 49:21 With and without Fuzzy Logic Approach

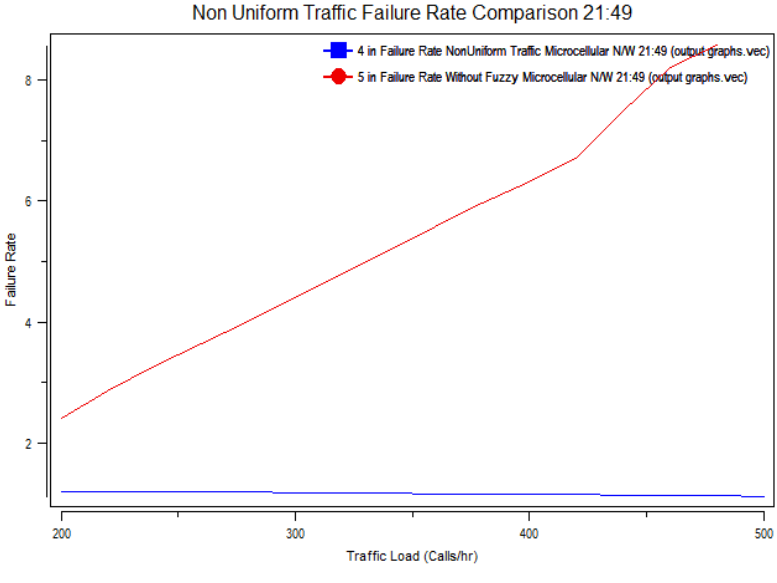


Fig. 7. Failure Rate Comparison Nonuniform Traffic Scenario 21:49 With and without Fuzzy Logic Approach

4 Conclusion

Fuzzy inference mechanism at call admission level has been proposed along with Multi agent system. It has tried to reduce failure rate when call arrival rate is high. Presented framework has stated the results for reduction of failure rate when fuzzy logic approach is applied. Results stated for different traffic scenarios are showing enhancement. The complexity of the proposed fuzzy call admission controller may be a concern, especially when the size of the system is large in terms of the cell number, the traffic class number, and the admission region size.

References

1. Papazoglou, P.M., Karras, D.A.: An Improved Multi-agent Simulation Methodology For Modelling And Evaluating Wireless Communication Systems Resource Allocation Algorithms. Journal Of Universal Computer Science 14(7), 1061–1079 (2008)
2. Bodanese, E.L., Cuthbert, L.G.: Application Of Intelligent Agents In Channel Allocation Strategies For Mobile Networks. In: IEEE International Conference On Communications, vol. 1, pp. 18–22 (June 2000)
3. Pendharkar, P.C., Harrisburg, P.S.: A Multi-agent Distributed Channel Allocation Approach For Wireless Networks. International Transactions In Operation Research 15(3), 325–337 (2006)
4. Kamble, M., Gupta, R.: A new framework for call admission control in wireless cellular network. In: IEEE Int. Conf. ETNCC (April 2011) 978-1-4577-0239-6/178-181
5. Kamble, M., Gupta, R.: DDCA in Cellular Network using Fuzzy Rule Based Multiagent System. International Journal of Computer Applications (0975–8887) 1(17) (February 2010)
6. Naik, Wei: Call-on-Hold for Improving the Performance of Dynamic Channel-Assignment Strategies in Cellular Networks. IEEE Transactions on Vehicular Technology 53(6), 1780–1793 (2004)
7. Ravichandran, M., Sengottuvelan, P., Shanmugam, A.: An Approach For Admission Control And Bandwidth Allocation In Mobile Multimedia Network Using Fuzzy Logic. International Journal Of Recent Trends In Engineering 1(1) (May 2009)
8. Raiyn, J.: Development And Comparison Between Resource Allocation Strategies Based On Intelligent Scheme And Traditional Distributed Dynamic Resource Allocation Strategies In Mobile Communication Networks. Wireless Personal Communications 40, 495–509 (2007)
9. Cao, G., Singhal, M.: Distributed Fault-tolerant Channel Allocation For Cellular Networks. IEEE Journal on Selected Areas in Communications 18(7), 1326–1337 (2000)
10. Jiang, Lal, Soundrajan: On Distributed Dynamic Channel Allocation in Mobile Cellular Networks. IEEE Transactions on Parallel and Distributed Systems 13(10) (October 2002)
11. Wang, Y.-T.: A Fuzzy-Based Dynamic Channel Borrowing Scheme for Wireless Cellular Networks. IEEE, 0-7803-7757-5/03/$17.K
12. Mahantasanapong, T., Phongphanphanee, C.: Automatic Channel Allocation in Cellular Network. In: Applied Telecommunications Symposium, 1-56555-276-8

Fuzzy Programming Approach to Solve Multi-objective Transportation Problem

Sandeep Kumar and Diwakar Pandey

Department of Mathematics, C.C.S.University, Meerut-250004 India
drsandeepmath@gmail.com, pandey_diwakar2k1@rediffmail.com

Abstract. Present paper deals with a multi-objective transportation problem. This problem has equality type constraints and some non-commensurable and conflicting objectives. The above objectives are fuzzy in nature. The problem has been solved by three methods using fuzzy programming approach. A numerical example is also given to compare the results obtained by different methods.

Keywords: Multi-objective transportation problem, Fuzzy programming, Goal programming.

1 Introduction

One of the important optimization problems is transportation problem. The classical transportation Problem (TP) is developed by Hithcock[7]. It is widely applied in real life problems such as distribution system, job assignment, and transportation. All the constraints (source and destination constraints) in TP have sign of equality and the the aim of its objective function is to minimize the total transportation cost. In practical situations, the coefficients of objective function do not have only transportation costs, these may also include the delivery time, number of goods transported, unfulfilled supply and demand, and others. In these situations, TP become multi-objective. Diaz [4], Diaz [5], and Ringuest and Rinks [9], provided various efficient algorithms to solve multi-objective transportation problem.

Zimmermann [13] introduced fuzzy linear programming with several objectives in which decision maker sets the limits of fuzziness with respect to the objective function and/or constraints. This approach is based on Bellmen and Zadeh's [1] decision making principle. Using fuzzy programming technique, Bit et al.[2] solved the multi-objective transportation problem with linear membership function. Thereafter, many researchers such as Dutta and Murthy [6], Mahapatra et al. [8], Ritha and Vinotha [10], Verma et al.[15] studied TP in fuzzy environment. To solve fuzzy multi-objective transportation problem, Verma et al.[15] used non-linear membership functions (exponential and hyperbolic) in fuzzy programming

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 525–533.
springerlink.com

and they concluded that nonlinear membership functions are not useful in comparison to the linear membership function.

Often the goals in multi-objective problem are conflicting. Present paper considers linear multi-objective transportation problem in which only objectives are fuzzy and are conflicting using different fuzzy programming techniques.

2 Crisp Multi-objective Transportation Problem

In a typical TP, a homogeneous product is transported from m sources to n destinations and they are characterized by available supply $a_1, a_2, ...a_m$ and demand levels $b_1, b_2, ...b_n$ respectively. A penalty C_{ij} is associated with transporting unit of the product from source i to destination j. A variable X_{ij} represents the unknown quantity to the shipped from source i to destination j.

Most of practical problems are multi-objective. So, a multi-objective TP is more efficient and realistic than to a single objective TP.

The general format of a multi-objective TP is stated as:

$$\text{Miniimize } z_k = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}^k X_{ij}, \quad k = 1, 2, ..., K$$

subject to,

$$\sum_{j=1}^{n} X_{ij} = a_i, \quad i = 1, 2, ..., m,$$

$$\sum_{i=1}^{m} X_{ij} = b_j, \quad j = 1, 2, ..., n,$$

$$X_{ij} \geq 0, \; X_{ij} \geq 0, \quad \text{for all } i \text{ and } j.$$

We assume that $a_i \geq 0$ for all i, $b_j \geq 0$ for all j and $C_{ij} \geq 0$ for all i and j, and

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \quad \text{(equilibrium condition)}$$

The equilibrium condition is necessary and sufficient condition ffor the existence of a flexible slution to the TP.

3 Fuzzy Multi-objective Transportation Problem (FMOTP)

Let U_k and L_k be the upper bound (worst value) and lower bound (best value) for the k^{th} objective Z_k, $k = 1, 2, \ldots, K$ respectively and $d_k = U_k - L_k$ is the degradation allowance for the k^{th} objective.

To find out the values of U_k and L_k, solve the multi-objective TP as a single objective TP k times by taking one of the objectives at a time and ignore others. Using these results, we prepare a pay-off matrix as follows:

$$\begin{array}{c} \\ X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(K)} \end{array} \begin{array}{c} \begin{array}{cccc} Z_1 & Z_1 \cdots & Z_K \end{array} \\ \begin{bmatrix} Z_{11} & Z_{12} \cdots & Z_{1K} \\ Z_{21} & Z_{22} \cdots & Z_{2K} \\ \cdots & \cdots & \cdots \\ Z_{K1} & Z_{K2} \cdots & Z_{KK} \end{bmatrix} \end{array}$$

where $X^{(1)}$,…, $X^{(k)}$ are optimal solutions for first, second,…, $k-th$ objective and $Z_{ij} = Z_j(X^i)$, $(i = 1, 2, ..., K, j = 1, 2..., K)$ be iow and $j-th$ column element of pay-off matrix.

Here, $U_k = \max\{Z_{1k}, Z_{2k}, ..., Z_{Kk}\}$ and $L_k = Z_{kk}$, $k = 1, 2, ..., K$.

The model for FMOTP is stated as:

Find $X_{ij}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n,$ so as to satisfy

$$Z_k \lesssim L_k, \quad k = 1, 2, \ldots, K$$

subject to,

$$\sum_{j=1}^{n} X_{ij} = a_i, \quad i = 1, 2, \ldots, m,$$

$$\sum_{i=1}^{m} X_{ij} = b_j, \quad j = 1, 2, \ldots, n,$$

$$X_{ij} \geq 0, \quad \text{for all } i \text{ and } j.$$

4 Solution Procedures

Firstly, we define a triangular membership function for the k^{th} objective function as follows:

$$\mu_k(Z_k) = \begin{cases} 1 & , Z_k \leq L_k, \\ 1 - \dfrac{Z_k - L_k}{U_k - L_k} & , L_k \leq Z_k < U_k, \\ 0 & , Z_k \geq U_k. \end{cases}$$

In this paper, to solve the FMOTP, we use the following three weighted fuzzy programming techniques:

(i) Weighted fuzzy linear programming [11]
(ii) Additive fuzzy goal programming with differential achievement degrees [3, 13]
(iii) Minmax technique [15]

4.1 Weighted Fuzzy Linear Programming Technique

In this technique, the crisp equivalent of FMOTP as follows:

$$\text{Max } \lambda$$

subject to,

$$W_k\left(1-\frac{Z_k-L_k}{U_k-L_k}\right)\geq\lambda,\ k=1,2,\ldots,K,$$

and

$$\sum_{j=1}^{n}X_{ij}=a_i,\quad i=1,2,\ldots,m,$$

$$\sum_{i=1}^{m}X_{ij}=b_j,\quad j=1,2,\ldots,n,$$

$$\lambda\leq 1,$$

$$X_{ij}\geq 0,\ \text{for all } i \text{ and } j.$$

where W_k is positive weight for the k[th] objective.

4.2 Additive Fuzzy Goal Programming with Differential Achievement Degrees

In this technique, decision maker chooses the desirable achievement degree for each membership function of the objective function instead of weights. The crisp model for FMOTP can be formulated as

$$\text{Max } \sum_{k=1}^{K}\mu_k$$

subject to

$$\mu_k=1-\frac{Z_k-L_k}{U_k-L_k},\ k=1,2,\ldots,K,$$

and

$$\sum_{j=1}^{n}X_{ij}=a_i,\quad i=1,2,\ldots,m,$$

$$\sum_{i=1}^{m}X_{ij}=b_j,\quad j=1,2,\ldots,n,$$

$$\mu_k\geq\alpha_k,$$

$$X_{ij}\geq 0,\quad \text{for all } i \text{ and } j.$$

where α_k is the desirable achievement degree for k[th] fuzzy goal.

4.3 Minmax Technique

This technique is proposed by [12] In this technique, the crisp model for FMOTP is described as

$$\text{Max } \lambda$$
$$\text{subject to,}$$
$$Z_k - p_k \le L_k,\ k = 1,2,\ldots,K,$$
$$\lambda + p_k \le 1,\ k = 1,2,\ldots,K,$$

and

$$\sum_{j=1}^{n} X_{ij} = a_i, \quad i = 1,2,\ldots,m,$$
$$\sum_{i=1}^{m} X_{ij} = b_j, \quad j = 1,2,\ldots,n,$$
$$\lambda \le 1,$$
$$X_{ij} \ge 0, \text{for all } i \text{ and } j,\ \lambda, p_k \ge 0,\ k = 1,2,\ldots,K,.$$

where p_k is positive deviational variable for k[th] objective Z

5 Numerical Example

$$\text{Min } Z_1 = 16X_{11} + 19X_{12} + 12X_{13} + 22X_{21} + 13X_{22} + 19X_{23} + 14X_{31} + 28X_{32} + 8X_{33}$$
$$\text{Min } Z_2 = 9X_{11} + 14X_{12} + 12X_{13} + 16X_{21} + 10X_{22} + 14X_{23} + 8X_{31} + 20X_{32} + 6X_{33}$$
$$\text{Min } Z_3 = 14X_{11} + 15X_{12} + 9X_{13} + 17X_{21} + 12X_{22} + 16X_{23} + 10X_{31} + 20X_{32} + 7X_{33}$$

subject to,

$$\sum_{j=1}^{3} X_{1j} = 14, \quad \sum_{j=1}^{3} X_{2j} = 16, \quad \sum_{j=1}^{3} X_{3j} = 12,$$
$$\sum_{i=1}^{3} X_{i1} = 10, \quad \sum_{i=1}^{3} X_{i2} = 15, \quad \sum_{i=1}^{3} X_{i3} = 17,$$
$$X_{ij} \ge 0,\ i = 1,2,\ldots,m;\ j = 1,2,\ldots,n.$$

Using TORA package[12], the optimal solution for objective Z_1 as follows:

$$X_{11} = 9, X_{13} = 5, X_{21} = 1, X_{22} = 15, X_{33} = 12$$

and remaining all X_{ij} are zeros. $Z_1 = 517.$

The optimal solution for objective Z_2 as follows:

$$X_{11} = 10, X_{13} = 4,\ X_{22} = 15, X_{23} = 1, X_{33} = 12$$

and remaining all X_{ij} are zeros. $Z_2 = 374.$

The optimal solution for objective Z_3 is obtained as:

$$X_{13}=14,\ X_{21}=1, X_{22}=15, X_{31}=9, X_{33}=3,$$

and remaining all X_{ij} are zeros. $Z_3=434$.

From the pay-off matrix, we find

$$U_1=535, U_2=424, U_3=472; L_1=517, L_2=374, L_3=434\ .$$

Solution based on technique 4.1

The equivalent crisp model can be formulated as

Max λ

subject to,

$$W_1\left[\frac{535-(16X_{11}+19X_{12}+12X_{13}+22X_{21}+13X_{22}+19X_{23}+14X_{31}+28X_{32}+8X_{33})}{18}\right]\geq\lambda,$$

$$W_2\left[\frac{424-(9X_{11}+14X_{12}+12X_{13}+16X_{21}+10X_{22}+14X_{23}+8X_{31}+20X_{32}+6X_{33})}{50}\right]\geq\lambda,$$

$$W_3\left[\frac{472-(14X_{11}+15X_{12}+9X_{13}+17X_{21}+12X_{22}+16X_{23}+10X_{31}+20X_{32}+7X_{33})}{38}\right]\geq\lambda,$$

$$\sum_{j=1}^{3}X_{1j}=14,\quad \sum_{j=1}^{3}X_{2j}=16,\quad \sum_{j=1}^{3}X_{3j}=12,$$

$$\sum_{i=1}^{3}X_{i1}=10,\quad \sum_{i=1}^{3}X_{i2}=15,\quad \sum_{i=1}^{3}X_{i3}=17,$$

$$X_{ij}\geq 0,\ i=1,2,\ldots,m;\ j=1,2,\ldots,n.$$

Table 1. Optimal results for different set of weights

Value of weights	Optimal solution	Value of λ
$W_1=0.5, W_2=0.3,$ $W_3=0.2$	$X_{11}=4\cdot9351,\ X_{13}=9\cdot0649,\ X_{21}=1,$ $X_{31}=4\cdot0649,\ X_{22}=15,\ X_{33}=7\cdot9351$	0.1481
$W_1=0.4, W_2=0.2,$ $W_3=0.4$	$X_{11}=9\cdot4915,\ X_{13}=4\cdot5085,\ X_{21}=0.5085,$ $X_{22}=15,\ X_{23}=0.4915,\ X_{33}=12$	0.1898
$W_1=0.3, W_2=0.2,$ $W_3=0.5$	$X_{11}=10,\ X_{13}=4\cdot0,\ X_{22}=15,\ X_{23}=1,\ X_{33}=12$	0.20
$W_1=0.2, W_2=0.5,$ $W_3=0.3$	$X_{11}=9,\ X_{13}=5,\ X_{21}=1,\ X_{22}=15,\ X_{22}=15,$ $X_{33}=12$	0.20

Solution based on technique 4.2

In his technique, firstly the crisp additive model without differential achievement of degree can be formulated as

$$\text{Max} \quad \mu_1 + \mu_2 + \mu_3$$

subject to,

$$18\mu_1 + 16X_{11} + 19X_{12} + 12X_{13} + 22X_{21} + 13X_{22} + 19X_{23} + 14X_{31} + 28X_{32} + 8X_{33} = 535,$$

$$50\mu_2 + 9X_{11} + 14X_{12} + 12X_{13} + 16X_{21} + 10X_{22} + 14X_{23} + 8X_{31} + 20X_{32} + 6X_{33} = 424,$$

$$38\mu_3 + 14X_{11} + 15X_{12} + 9X_{13} + 17X_{21} + 12X_{22} + 16X_{23} + 10X_{31} + 20X_{32} + 7X_{33} = 472,$$

$$\sum_{j=1}^{3} X_{1j} = 14, \quad \sum_{j=1}^{3} X_{2j} = 16, \quad \sum_{j=1}^{3} X_{3j} = 12,$$

$$\sum_{i=1}^{3} X_{i1} = 10, \quad \sum_{i=1}^{3} X_{i2} = 15, \quad \sum_{i=1}^{3} X_{i3} = 17,$$

$$\mu_i \le 1,\ i = 1,2,3.$$

$$X_{ij} \ge 0,\ i = 1,2,3;\ j = 1,2,3.$$

The optimal is obtained as

$$\mu_1 = 1,\ \mu_2 = 0.9,\ \mu_3 = 0.2105,$$

$$X_{11} = 9,\ X_{13} = 5, X_{21} = 1, X_{22} = 15,\ X_{33} = 12$$

Now, we choose different sets of desirable achievement of degrees and obtain the results as follows:

Table 2 Optimal results for different set of achievement degrees

Value of desirable achievement degrees	Optimal solution	$\sum_{i=1}^{3} \mu_i$
$\alpha_1 = 0.70, \alpha_2 = 0.70,$ $\alpha_3 = 0.60$	$\mu_1 = 0 \cdot 8444,\ \mu_2 = 0 \cdot 7600,\ \mu_3 = 0 \cdot 60, \backslash\backslash X_{11} = 7 \cdot 6,$ $X_{13} = 6 \cdot 4,\ X_{21} = 1,\ X_{22} = 15,\ X_{31} = 1 \cdot 4,\ X_{33} = 10 \cdot 6$	2.2044
$\alpha_1 = 0.90, \alpha_2 = 0.85,$ $\alpha_3 = 0.55$	$\mu_1 = 0.95,\ \mu_2 = 0 \cdot 85,\ \mu_3 = 0 \cdot 55,\ X_{11} = 8.55,$ $X_{13} = 5.45,\ X_{21} = 1, X_{22} = 15, X_{31} = 0.45, X_{33} = 11.55$	2.350
$\alpha_1 = 0.95, \alpha_2 = 0.80,$ $\alpha_3 = 0.54$	$\mu_1 = 0.9711,\ \mu_2 = 0 \cdot 8740,\ \mu_3 = 0 \cdot 5400, X_{11} = 8.74,$ $X_{13} = 5.26,\ X_{21} = 1, X_{22} = 15, X_{31} = 0.26, X_{33} = 11.74$	2.3851

Solution based on technique 4.3

Apply the technique 4.3, the crisp equivalent comes out to be

$$\text{Max} \quad \lambda$$

subject to,

$$16X_{11}+19X_{12}+12X_{13}+22X_{21}+13X_{22}+19X_{23}+14X_{31}+28X_{32}+8X_{33}-p_1\le 517$$
$$9X_{11}+14X_{12}+12X_{13}+16X_{21}+10X_{22}+14X_{23}+8X_{31}+20X_{32}+6X_{33}-p_2\le 374$$
$$14X_{11}+15X_{12}+9X_{13}+17X_{21}+12X_{22}+16X_{23}+10X_{31}+20X_{32}+7X_{33}-p_3\le 434$$

$$\lambda+\frac{1}{18}p_1\le 1,$$
$$\lambda+\frac{1}{50}p_2\le 1,$$
$$\lambda+\frac{1}{38}p_3\le 1,$$
$$\sum_{j=1}^{3}X_{1j}=14,\quad \sum_{j=1}^{3}X_{2j}=16,\quad \sum_{j=1}^{3}X_{3j}=12,$$
$$\sum_{i=1}^{3}X_{i1}=10,\quad \sum_{i=1}^{3}X_{i2}=15,\quad \sum_{i=1}^{3}X_{i3}=17,$$
$$\lambda\le 1,$$
$$\lambda\ge 0,\ p_i\ge 0, X_{ij}\ge 0,\ i=1,2,3;\ j=1,2,3.$$

The optimal solution is obtained as :

$$\lambda=0\cdot 6552, X_{11}=6\cdot 5517,\ X_{13}=7\cdot 4483,$$
$$X_{21}=1, X_{22}=15,\ X_{31}=2\cdot 4483, X_{33}=9\cdot 5517.$$

6 Conclusions

In this paper, three fuzzy programming techniques have been applied to solve the multi-objective transportation problem. In first technique, the importance of objectives is measured through weights. In the second technique, we used a set of desirable achievement degrees instead of weights, and then improved the achievement degrees. In the third technique, minimax approach has been used to solve the problem. The second approach has been observed to be better other than two.

Acknowledgements. The authors are grateful to the anonymous referee for his valuable suggestions. The first author is also thankful to Council of Scientific and Industrial Research, India for providing him financial assistance to pursue this work.

References

[1] Bellman, R.E., Zadeh, L.A.: Decision-making in a Fuzzy Environment. Management Science 17(4), 141–164 (1970)

[2] Bit, A.K., Biswal, M.P., Alam, S.S.: Fuzzy Programming Problem. Fuzzy Sets and Systems 50, 135–141 (1992)

[3] Chen, L.-H., Tsai, F.-C.: Fuzzy Goal Programming with Different Importance and Priorities. European Journal of Operational Research 133, 548–556 (2001)

[4] Diaz, J.A.: Solving Multi-objective Transportation Problem. Ekonom. Mat. Obzor 14, 267–274 (1978)
[5] Diaz, L.A.: Finding a Complete Description of all Efficient Solutions to a Multi-objective Transportation Problem. Ekonom. Mat. Obzor 15, 62–73 (1979)
[6] Dutta, D., Murthy, A.S.: Fuzzy Transportation Problem with Additional Restrictions. ARPN Journal of Engineering and Applied Sciences 5(2), 36–40 (2010)
[7] Hitcbcock, F.L.: The Distribution of a Product from Several Sources to Numerous Localities. J. Math. Phys. 20, 224–230 (1941)
[8] Mahapatra, D.R., Roy, S.K., Biswal, M.P.: Multi-objective Stochastic Transportation Problem involving Log-normal. Journal of Physical Sciences 14, 63–76 (2010)
[9] Ringuest, J.L., Rinks, D.B.: Interaction Solutions for the Linear Multi-objective Transportation Problem. European J. Oper. Res. 32, 96–106 (1987)
[10] Ritha, W., Vinotha, J.M.: Multi-objective Two Stage Fuzzy Transportation Problem. Journal of Physical Sciences 13, 107–120 (2009)
[11] Roy, T.K., Maiti, M.: Multi-objective Inventory Models of Deteriorating with Some Constraints in a Fuzzy Environment. Computers Ops. Res. 25(12), 1085–1095 (1998)
[12] Taha, H.A.: Operation Research, An Introduction, 5th edn. Macmillan, New York (1992)
[13] Tiwari, R.N., Dharmar, S., Rao, J.R.: Fuzzy Goal Programming-an Additive Model. Fuzzy Sets and Systems 24, 27–34 (1987)
[14] Rakesh, V., Biswal, M.P., Biswas, A.: Fuzzy Programming Technique to Solve Mult-objective Transportation Problems with Some Non-linear Membership Functions. Fuzzy Sets and Systems 91, 37–43 (1997)
[15] Yaghoobi, M.A., Tamiz, M.: A Method for Solving Fuzzy Goal Programming Problems Based on Minmax Approach. European Journal of Operational Research 177, 1580–1590 (2007)
[16] Zimmermann, H.-J.: Fuzzy Programming and Linear Programming with Several Objective Functions. Fuzzy Sets and Systems 1, 45–55 (1978)

Review on Sinkhole Detection Techniques in Mobile Adhoc Network

Nisarg Gandhewar and Rahila Patel

Department of CSE RCERT, Chandrapur, India
{nisarg.gandhewar,rahila.patel}@gmail.com

Abstract. MANET is multi-hop wireless network of autonomous mobile nodes with no preset infrastructure where each node can move in any direction as well play a role of router. Emergence of cheaper and more powerful wireless devices make MANET a fastest growing network, which is increasingly being used in many applications at the same time cooperative nature of nodes to communicate beyond their transmission range & vulnerability of MANET expose them to a wide variety of active & passive attacks, Out of which sinkhole is one of severe representative attack in MANET, where malicious node tries to draws all network traffic towards it & drop packets, which leads to performance degradation of network as well it can cause other attacks possible. So sinkhole node should be detected as well as detached as early as possible thus few techniques have been proposed for sinkhole detection in MANET.

So this paper presents overview of various sinkhole detection techniques, brief description of sinkhole attack, along with other possible routing attacks, their causes & metrics affected by these attacks.

Keywords: MANET, Vulnerability, Active, Passive, Sinkhole.

1 Introduction

MANET is one of self configuring fastest emerging wireless technology whose structure is shown in fig 1, due to commencement of economical, small & more powerful wireless devices. It is being used in most of applications, ranging from military to civilian, where each node acts as router. To facilitate communication in adhoc network, a routing protocol is vital whose primary goal is to establish accurate & efficient route between pair of nodes, due to this lot of reactive, proactive & hybrid routing protocols have been proposed for MANET & its success depends on people's confidence in its security.

Active research work for MANET is carrying on mainly in the fields of routing, resource management, power control, and security, among this security issue has

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 535–548.
springerlink.com

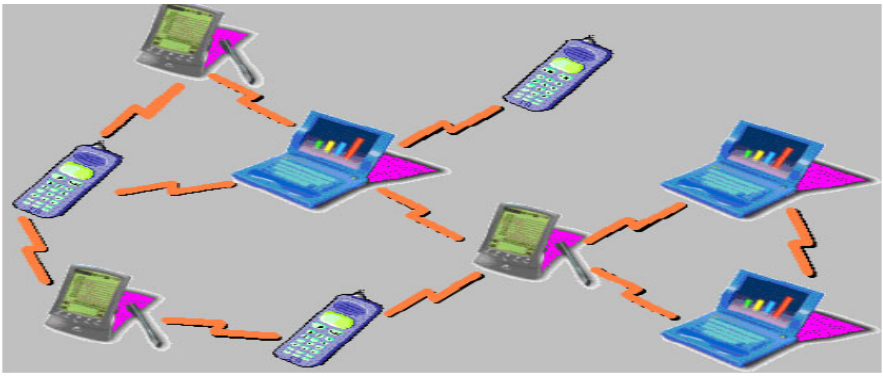


Fig. 1. Structure of MANET

become one of the major concerns & challenge in MANET due to its various vulnerability, especially for those selecting sensitive applications. In most of the routing protocols, cooperative nodes expose MANET to a wide range of security attacks, as they take cooperation of other nodes to communicate beyond their transmission range. A unified security solution is in very much need for such networks to protect both route and data forwarding operations in the network layer. This will solely disturb the network operation from correct delivering of the packets, like the malicious nodes can give stale routing updates or drop all the packets passing through them.

Attacks in MANET mainly classified into two categories as passive where attacker does not disrupt proper operation of network, with no alteration of any data & Active where attacker modifies some data stream or creates false stream, some of attacks under this category are flooding, wormhole, black hole, grey hole, rushing, selfish, fabrication, route falsification, sleep deprivation, byzantine & sinkhole attack [1].

Among all of these attacks, sinkhole is one of dangerous representative attack in MANET, where malicious node try to attract all network traffic towards it by convincing its neighbors through broadcasting fake routing information & modify or drops packets sent for forwarding, which results in degradation of network performance. So few sinkhole detection techniques have been proposed such as collaborative, adaptive, cooperative, cluster analysis & intrusion indicator technique [5]. The consequences of various attacks on routing protocols in MANET can be realized quantitatively by using various performance metrics such as PDR packet delivery ratio, end to end delay, network overhead, packet loss & throughput.

The rest of the paper is organized as follows: Section 2 presents analysis of attacks in MANET, while section 3 describes realization of sinkhole attack on the context of DSR protocol along with different sinkhole detection techniques. Whereas section 4 represents various metrics affected due to different attacks, Section 5 concludes this paper & finally section 6 focuses on our proposed work.

2 Analysis of Attacks

Security has becomes one of great challenge & concern for MANET due to variety of reasons, on which active research work has been carrying out continuously. Attack in MANET can be realized in fig 2 where adversaries can easily disturb and absorb network traffic, inject themselves into the selected data transmission path between the source and destination, thus control network traffic flow, as shown in Fig 2, where a malicious node M can interfere itself in between any of intermediate nodes participating in the communication in the chosen path (in figure 1 to N represents the number of Intermediate nodes) between source S and destination D.

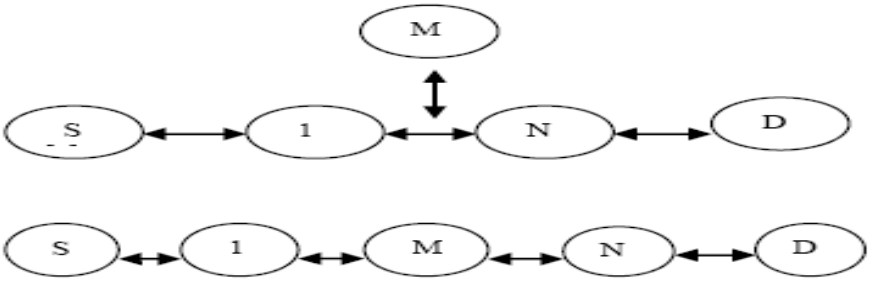


Fig. 2. Interference of malicious node in between source& destination communication

2.1 Attack Characteristics

Many characteristics can be used to classify attacks in the ad hoc network, which would include looking at the behavior of the attacks as (passive vs. active), source of the attacks as (external vs. internal) and the number of the attackers as (single vs. multiple).

2.1.1 Passive vs. Active Attacks

Passive attacks do not intend to disrupt the network operations, they are launched to steal valuable information in the targeted networks. Examples of passive attacks are eavesdropping attacks and traffic analysis attacks. Detecting this kind of attack is difficult because neither the system resources nor the critical network functions are physically affected to prove the intrusions. Active attacks on the other hand actively alter the data with the intention to obstruct the operation of the targeted networks. Examples of active attacks comprise actions such as message modifications, message replays, message fabrications and the denial of service attacks. Active attacks have following special characteristics

- *Route Disruption*: A malicious node either destroys an existing route or prevents a new route from being established.
- *Route Incursion*: A malicious node adds itself into a route between source and destination nodes.

- *Node Segregation:* A given node is prevented from communicating with any other nodes. It differs from route disruption in that route disruption is targeting at a route with two given nodes, while node isolation is targeting at all possible routes to or from a given node.
- *Resource Consumption:* The communication bandwidth in the network or storage space at individual Node is consumed [2].

2.1.2 External vs. Internal Attacks

External attacks are launched by adversary who is not initially authorized to participate in the network operations. These attacks usually aim to cause network congestion, denying access to specific network function or to disrupt the whole network operations. Bogus packets injection, denial of service, and impersonation are some of the attacks that are usually initiated by the external attackers.

More severe attacks in the ad hoc networks might come from the second source of attacks, which is the internal attack. Internal attacks are initiated by the authorized nodes in the networks, and might come from both compromised and misbehaving nodes. Internal nodes are identified as compromised nodes if the external attackers hijacked the authorized internal nodes and are then using them to launch attacks against the ad hoc networks. Attacks that are caused by the misbehaving internal nodes are difficult to detect because to distinguish between normal network failures and misbehaviour activities in the ad hoc networks is not an easy task.

2.1.3 Single vs. Multiple Attackers

Attackers might choose to launch attacks against the ad hoc networks independently or by colluding with the other attackers. One man action or single attackers usually generate a moderate traffic load as long as they are not capable to reach any wired facilities. Since they also have similar abilities to the other nodes in the networks, their limited resources become the weak points to them.

However, if several attackers are colluding to launch attacks, defending the ad hoc networks against them will be much harder. Colluding attackers could easily shut down any single node in the network and be capable to degrading the effectiveness of network's distributed operations including the security mechanisms. Adding to the severity, colluding attackers could be widely distributed or reside at the certain area where they presumed high communication rate in the networks exist. If no suitable security measures employed, nodes in that targeted area are susceptible to any kind of denial of service (DoS) attacks that could be launched by the colluding attackers [22].

2.2 Causes of Attacks in MANET

Vulnerability is a weakness in security system while MANET is more vulnerable than wired network due to various reasons. So some of the causes of attack in MANET are listed below.

- **Lack of centralized management**

MANET doesn't have a centralized monitor server. The absence of management makes the detection of attacks difficult because it is not easy to monitor the traffic in a highly dynamic and large scale adhoc network. Lack of centralized management will impede trust management for nodes.

- **Resource availability**

Resource availability is a major issue in MANET. Providing secure communication in such changing environment as well as protection against specific threats and attacks, leads to development of various security schemes and architectures. Collaborative adhoc environments also allow implementation of self organized security mechanism.

- **Scalability**

Due to mobility of nodes, extent of adhoc network changing all the time. So scalability is a major issue concerning security. Security mechanism should be capable of handling a large network as well as small ones.

- **Cooperativeness**

Routing algorithm for MANET usually assume that nodes are cooperative and non-malicious. As a result a malicious attacker can easily become an important routing agent and disrupt network operation by disobeying the protocol specifications.

- **Dynamic topology**

Dynamic topology and changeable nodes membership may disturb the trust relationship among nodes. The trust may also be disturbed if some nodes are detected as compromised. This dynamic behavior could be better protected with distributed and adaptive security mechanisms.

- **Limited power supply**

The nodes in mobile adhoc network need to consider restricted power supply, which will cause several problems. A node in mobile adhoc network may behave in a selfish manner when it is find that there is only limited power supply [12].

2.3 Classical Attacks

- **RREQ Flooding Attack**

A malicious node sends a huge number of RREQ packets in an attempt to consume the network resources. The source IP address is forged to a randomly selected node and the broadcast ID is intentionally increased. It makes possible for an adversary to carry out DoS by saturating the support with a quantity of broadcasting messages, by reducing the output of nodes, and in the worst case, to prevent them from communicating [12].

- **Location disclosure attack**

An attacker discover the Location of a node or structure of entire networks and disclose the privacy requirement of network through the use of traffic analysis techniques, or with simpler probing and monitoring approaches. Adversaries try to figure out the identities of communication parties and analyze traffic to learn the network traffic pattern and track changes in the traffic pattern. The leakage of such information is devastating in security [10].

- **Replay Attack**

An attacker that performs a replay attack retransmite the valid data repeatedly to inject the network routing traffic that has been captured previously. This attack usually targets the freshness of routes, but can also be used to undermine poorly designed security solutions.

- **Jamming Attack**

In this kind of attack objective of a jammer is to interfere with legitimate wireless communications & to degrade the overall QoS of the network, Jammer can achieve this goal by either preventing a real traffic source from sending out a packet, or by preventing the reception of legitimate packets to disturb communications.

- **Byzantine Attack**

In this attack, a compromised intermediate node works alone, or a set of compromised intermediate nodes works in collusion and carry out attacks such as creating routing loops, forwarding packets through non-optimal paths, or selectively dropping packets, which results in interruption or degradation of the routing services.

- **Spoofing Attack**

In spoofing attack, the attacker steals the identity of another node in the network, thus it receives the messages that are meant for that node. Usually, this type of attack is launched in order to gain access to the network so that further attacks can be launched, which could seriously cripple the network.

- **Wormhole Attack**

In this attack two attackers, connected by a high-speed off-channel link, are strategically placed at different ends of a network. These attackers then records data they overhear, forward it to each other, and replay the packets at other end of the network. Replaying valid network messages at improper places, wormhole attackers can make far apart nodes believe they are immediate neighbors, and force all communications between affected nodes to go through them.

- **Blackhole Attack**

In this type of the attack, a malicious node waits for its neighbors to initiate a route discovery process. Once the malicious node receives a broadcasted RREQ packet, it immediately sends a false RREP packet with a greater sequence number.

So, the source node assumes that the malicious node is having a fresh route towards the destination node and ignores RREP packets received from other nodes. The malicious node takes all the routes towards it and does not allow forwarding any packet anywhere[15].

2.4 Advanced Attacks

- **Neighbor Attack**

When an intermediate node receives a RREQ/RREP packet, it adds its own ID in the packet before forwarding it to the next node. A malicious node simply forwards the packet without adding its ID in the packet. This causes two nodes that are not within the communication range of each other believe that they are neighbors, resulting in a disrupted route. In the Neighbor attack, the malicious node does not catch and capture the data packets from the source node [2].

- **Jelly Fish Attack**

A jellyfish attacker first needs to intrude into the multicast forwarding group. It then delays data packets unnecessarily for some amount of time before forwarding them. This results in significantly high end-to-end delay and thus degrades the performance of real applications.

- **Rushing Attack**

In on demand routing protocol each intermediate node must forward only the first received route request from each route discovery & all further received route requests are ignored So, a malicious node simply exploits this property of the operation of route discovery by quickly forwarding RREQ packets in order to gain access to the forwarding group. As a result, any discovered route includes attacker & source node will not be able to discover any valid routes that do not include the malicious node [2].

- **Selfish attack**

It mainly involves no collaboration for the good performance of the network. We can identify two types of nodes which do not wish to take part in the network. Defective nodes which do not work flawlessly & malicious, it is those which intentionally, try to tackle the system attack on the integrity of the data, availability of the services, authenticity of the entities. Selfish nodes are those entities whose objective is to maximize their benefit [1].

- **Grey Hole Attack**

This attack is also known as routing misbehavior attack which works in two phases. In first phase node advertises itself as having valid route to destination while in second phase nodes drops intercepted packets with certain probability [19].

- **Sleep deprivation**:

In a routing protocol, sleep deprivation attacks might be launched by flooding the targeted node with unnecessary routing packets. For instance, attackers could flood any node in the networks by sending a huge number of route request (RREQ), route replies (RREP) or route error (RERR) packets to the targeted node. As a result, that particular node is unable to participate in the routing mechanisms and is unreachable to the other nodes in the networks [1].

- **Route falsification attack**

In this type of attack, malicious node may works in both direction, source to destination during route request and destination to source during Route reply. When source sends request to destination node or when destination or other node gives reply for request. In this attack, malicious node falsify the route request or route reply packets to indicates a better path to the source for making large portion of the traffic go through them. When the source selects the falsified path, the malicious nodes can drop data packets they receive silently [18].

- **Fabrication Attack**

This kind of an active attack breaks authenticity by exposing itself to become the source entity. After become a part of the network it sends error message to other legal nodes to say the route is not available any more. Thus, other node will then update their table with this false information. By this way it drops the routing packets [18].

- **Sinkhole Attack**

This is one of severe attack in MANET which is described briefly in next section.

3 Analysis of Sinkhole Problem

3.1 Sinkhole Problem

In this type of attack sinkhole node tries to attract data to itself by convincing neighbors through broadcasting fake routing information & let them know itself on the way to specific nodes. Through this procedure, sinkhole node attempts to draw all network traffic to itself. Thereafter it alters the data packet or drops the packet silently. It increases network overhead, decreases network's life time by boosting energy consumption, finally destroy the network.

In DSR protocol, sinkhole attack is set up by modifying sequence number in RREQ, higher the sequence number, then route will be more recent the packet contains. Sinkhole node selects the source, destination node. It observes the source node's sequence number carefully, and generates bogus RREQ with selected source, destination and higher sequence number than observed source sequence

number. It then adds itself on the source route and broadcasts the bogus RREQ. Nodes that take this bogus RREQ recognize that the reversed route could be a better route to the source than incumbent route.

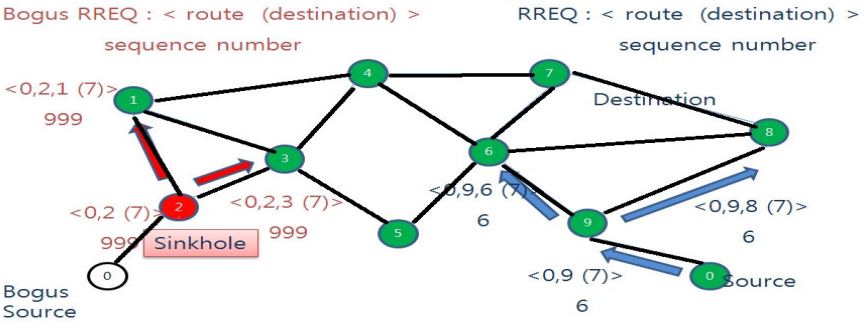


Fig. 3. Bogus RREQ Creation

Fig. 3 [5] shows the creation of the bogus RREQ packet. Sinkhole node 2 makes the bogus RREQ with bogus source creation which looks as if it is originated by node 0. Sequence number of bogus packet is 999, much higher than original source's, 6.

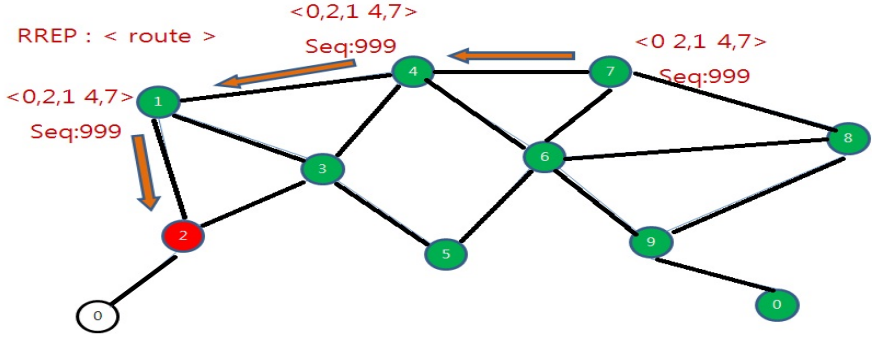


Fig. 4. Bogus RREP Circulation

Bogus RREP is shown in Fig 4 [5] where intermediate nodes on route learn that node 2 is on one hop distance to node 0 and to send packet to node 0, the data packet may go through the node 2. Sinkhole node 2 can easily repeat this procedure & can draw all local network traffic to itself. Thereafter node 2 can do malicious acts including dropping, or modifying the traffic.

3.2 Sinkhole Detection Techniques

There are lot of generalized IDS is available which can detect different attacks but we require specialized IDS which can detect specific attack efficiently such as sinkhole. An excellent intrusion detection technique should take care of security in the network. This section explores different sinkhole detection techniques in MANET as follows.

3.2.1 SIIT - Sinkhole Intrusion Indicators Technique

This technique makes use of two sinkhole detection indicators for MANET after systematic study of sinkhole problem on the context of DSR protocol in wireless ad-hoc network. The two indicators proposed here are Sequence number discontinuity and Route add ratio. The sequence number discontinuity is measured by the overall average difference between the current and the last sequence number from each node, when source node initiates route discovery it publishes sequence number & increase it by one, as sinkhole node advertises fake route information by generating high sequence number, sequence number difference between normal & sinkhole node can be an clue of sinkhole existence. The limitation of this indicator is that the attacker may use sequence numbers that are not unusually high, but only just high enough to cause the route overriding effect that the attacker desires.

The route-add ratio is the proportion of routes that traverse a particular node to the total number of routes added to this node's routing table. The sinkhole attack causes nodes in the network to add routes that pass through the sinkhole. The problem here is that, for each time a source route is found, the route-add ratio has to be calculated. All the two variables must have values greater than the corresponding before the system issues an intrusion alert [5] [6].

3.2.2 Adaptive Technique

The adaptive technique makes use of trust based mechanism. This approach is fully distributed and allows the ad hoc node to detect sinkholes. Initially, every ad hoc node assigns a trust weight to its neighbors. During the transmission if a neighbor fails to transmit its message to a designated receiver, the ad hoc node then reduces the trust weight it has given to the neighbor. The neighbor then subsequently deducts the assigned weight of the next ad hoc node in the transmission path. Once the trust weight of any neighbor falls below a set threshold, then the neighbor is treated as a suspicious node. The adhoc node will then consult with its other neighbors in order to find out whether or not suspicious node should be identified as malicious node [7].

3.2.3 Cooperative Technique

This technique makes use of three different kinds of packets namely Sinkhole Alarm Packet (SAP), Sinkhole Detection Packet (SDP) and Sinkhole Node Packet (SNP). Sinkhole Alarm Packet will contains sinkhole route, sequence number of

the bogus RREQ, current sequence number of the node itself, while Sinkhole Detection Packet contains the common path, sequence number of bogus RREQ, network id of itself. The nodes in the sinkhole path are not allowed to generate or forward an SDP. If any node receives an SDP from the nodes in the sinkhole path, it simply discards the packet and detaches the sender of the SDP from the network & in Sinkhole Node Packet to inform the network of sinkhole node, the node broadcasts a SNP. The SNP packet will contain the sinkhole node to the whole network unless it received an SNP for this sinkhole route from another node [9].

3.2.4 Collaborative Technique

This technique mainly involves selection of single node as monitor node out of many where monitor node initiates detection process, based on the message that it receive during this process, each node determines node it suspect to be malicious & send votes to monitor node which upon carefully inspecting votes determine malicious node from the suspected nodes.

The problem with this collaborative technique is that it puts some extra burden on the node which will be performing role of monitor node. As the mobile nodes have limitation of battery power, this approach can cause monitor node to fail by consuming some fair bit of power. Even though we select a node which has high capabilities as the monitor node, the problem will still persist in the form of mobility [8].

3.2.5 Cluster Analysis Technique

Cluster analysis is a data mining technique which works by grouping patterns which are similar to each other in single group and different from patterns in other groups. In this approach, cluster analysis is used to separate false RREQs from normal RREQs and to verify indicators for detection. It also makes use of hierarchical approach, which does not require predetermined numbers of groups. This Approach is used because there may be more than two groups such as false RREQs or normal RREQs. Cluster analysis requires distance measures to examine the differences among clusters. But the problem with this approach can be the mobility of the nodes and the necessary of some controlling point [13].

4 Performance Metrics

As routing protocols in MANET may be victim of various active attacks, the consequences of that attack can be realized by quantitative study of values of different metrics used to measure performance of routing protocols which are as follows.

- **Throughput:** This is the percentage of sent data packets to the actually received by the intended destination.
- **Average end-to-end delay:** It is defined as the average time taken by the data packets to propagate from source to destination across a MANET.

This includes all possible delays caused by buffering during routing discovery latency, queuing at the interface queue, and retransmission delays at the MAC, propagation and transfer times.

- **Packet Delivery Ratio:** It is the ratio of the number of packets received by the destination to the number of data packets generated by the source.
- **Network Overhead:** This is the ratio of routing-related transmissions (ROUTE REQUEST, ROUTE REPLY, and ROUTE ERROR) to data transmissions in a simulation. Some routing packets are more expensive to the network than other packets.
- **Packet Loss:** It is the measure of number of packets dropped by nodes due to various reasons [4].

5 Conclusion

Thus we have studied various routing attacks, their causes & sinkhole detection techniques available in MANET & found that sinkhole is one of severe attack which needs to be detected as early as possible, which is studied & simulated on the context of DSR protocol only, thus it needs to be studied on the context of other protocol also & change in the performance of protocol needs to be determined after this attack, whereas various sinkhole detection techniques are available but they are having some limitations. Therefore we require strong mechanism which can efficiently detect & helps to prevent adhoc network from sinkhole attack by overcoming limitations of available detection techniques. Hence we are trying to develop efficient mechanism, which will overcome the limitations of existing sinkhole detection methods.

6 Proposed Work

Our work mainly focuses on analyzing sinkhole problem on the context of very popular Adhoc on Demand Distance Vector (AODV) protocol by systematically evaluating its performance before & after sinkhole attack by using various performance metrics. AODV is considered to be the best alternative to DSR protocol on which sinkhole problem is already analyzed. Also we are planning to develop a new mechanism which will detect sinkhole problem & mitigate its effect on AODV protocol by overcoming limitations of existing mechanism. We are planning to use most widely used network simulation tool that is NS2 in order to carry out simulation of our proposed work. Basically our work has following objectives.

- To evaluate original performance of AODV by using various Performance metrics.
- To evaluate change in performance of AODV under sinkhole attack.
- To develop mechanism for detection of sinkhole attack in MANET.
- To develop a mechanism to prevent network from sinkhole attack.
- To perform overall analysis of results obtained.

References

[1] Konate, K., Abdourahime, G.: Attack Analysis in mobile Adhoc network Modeling and Simulation. In: Second International Conference on Intelligent Systems. IEEE (2011)

[2] Alikhany, M., Abadi, M.: A Dynamic Clustering Based Approch For Anomaly Detection In AODV Based MANET. In: International Symposium on Computer Network & Distributed System. IEEE (2011)

[3] Kannan, S., Kalaikumaran, T., Kartik, S.: Review of Attack prevention Methods in MANET. Journal of Modern Mathematics & Statistics 5(1), 37–42 (2011)

[4] Shrivastava, L., Bhadauria, S.S., Tomar, G.S.: Performance Evaluation of Routing Protocols in MANET with different traffic loads. In: International Conference on Communication Systems and Network Technologies. IEEE (2011)

[5] Kim, K., Kim, S.: A Sinkhole Detection Method based on Incremental Learning in Wireless Ad Hoc Networks. Korea Advanced Institute of Science & Technology, Korea

[6] Culpepper, B.J., Chris Tseng, H.: Sinkhole Intrusion Indicators in DSR MANET. In: First International Conference on Broadband Networks. IEEE (2004)

[7] Thumthawatworn, T., Yeophantong, T., Sirikriengkrai, P.: Adaptive Sinkhole Detection on Wireless Ad Hoc Networks. Assumption University, IEEE, Thailand (2006)

[8] Marchang, N., Datta, R.: Collaborative techniques for intrusion detection in mobile ad-hoc networks. Ad Hoc Networks 6, 508–523 (2008)

[9] Kim, G., Han, Y., Kim, S.: A cooperative-sinkhole detection method for mobile ad hoc networks. International Journal of Electronics and Communication 64, 390–397 (2010)

[10] Goyal, P., Parmar, V., Rishi, R.: MANET: Vulnerabilities,Challenges, Attacks, Application. IJCEM International Journal of Computational Engineering & Management 11 (January 2011)

[11] Mamatha, G.S., Sharma, S.C.: A Highly Secured Approach against Attacks in MANETS. International Journal of Computer Theory and Engineering 2(5) (October 2010)

[12] Manikandan, K.P., Satyaprasad, R.: A Survey on Attacks and Defense Metrics of Routing Mechanism in Mobile Ad hoc Networks. International Journal of Advanced Computer Science and Application (IJACSA) 2(3) (March 2011)

[13] Shim, W., Kim, G., Kim, S.: A distributed sinkhole detection method using cluster analysis. Expert Systems with Applications (2010)

[14] Shah, S., Khandre, A., Shirole, M., Bhole, G.: Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation. In: Mobile and Pervasive Computing, CoMPC 2008 (2008)

[15] Saini, A., Kumar, H.: Effect Of Black Hole Attack On AODV Routing Protocol In MANET. IJCST 1(2) (December 2010)

[16] Abidin, A.F.A., Usop, N.S.M., Yusof, M.K.: Performance Comparison of Wireless Ad Hoc Routing Protocols. International Journal on Computer Science and Engineering (IJCSE) 3(1) (January 2011)

[17] Jhaveri, R.H., Patel, A.D.: MANET Routing Protocols and Wormhole Attack against AODV. IJCSNS International Journal of Computer Science and Network Security 10(4) (April 2010)

[18] Singh, J., Singh, A., Shree, R.: An Assessment of Frequently Adopted Unsecure Patterns in Mobile Ad hoc Network: Requirement and Security Management Perspective. International Journal of Computer Applications (0975–8887) 24(9) (June 2011)

[19] Manikandan, K.P., Satyaprasad, R., Rajasekhararao: Analysis and Diminution of Security Attacks on Mobile Ad hoc Network. IJCA Special Issue on "Mobile Ad-hoc Networks" MANETs (2010)

[20] Banerjee, S.: Detection/Removal of Cooperative Black and Gray Hole Attack in Mobile Ad-Hoc Networks. In: Proceedings of the World Congress on Engineering and Computer Science, WCECS 2008 (October 2008)

[21] Sharma, S.C., Mamatha, G.S.: Network Layer Attacks and Defense Mechanisms in MANETS- A Survey. International Journal of Computer Applications (0975–8887) 9(9) (November 2010)

[22] Razak, S.A., Furnell, S.M., Brooke, P.J.: Attacks against Mobile Ad Hoc Networks Routing Protocols, Network Research Group, University of Plymouth, Devon

Performance of VANET Routing Protocols Using Realistic Mobility Model

Ravi S. Shukla[1] and Neeraj Tyagi[2]

[1] Invertis University, Bareilly
ravipraful@gmail.com
[2] MNNIT, Allahabad
neeraj@mnnit.ac.in

Abstract. Vehicular Ad hoc networks (VANETs) allow vehicles to form a self-organized network without the need for a permanent infrastructure. Successful implementations of VANET require standard network protocols, secure communication, and coverage of wireless network among vehicles as well as drivers themselves. This paper presents a simulation study of Realistic Vehicular Mobility Model on the performance study of VANET that uses Ad-hoc On-Demand Distance Vector (AODV), Dynamic MANET On demand (DYMO) and Optimized Link State Routing (OLSR). It gives an overview of the developments of the mobility model and routing protocols for VANET.

Keywords: VANET, Routing Protocol and Mobility models.

1 Introduction

A VANET is a wireless network that is formed between vehicles on an as needed basis. To participate in a VANET, vehicles must be equipped with wireless transceivers and computerized control modules that allow them to act as network nodes. Each vehicle's wireless network range may be limited to a few hundred meters, so providing end-to-end communication across a larger distance requires messages to hop through several nodes. Network infrastructure is not required for a VANET to be created, although permanent network nodes may be used in the form of roadside units. These roadside units open up a wide variety of services for vehicular networks, such as acting as a drop point for messages on sparsely populated roads. Most nodes in a VANET are mobile, but because vehicles are generally constrained to roadways, they have a distinct controlled mobility pattern that is subject to vehicular traffic regulations. In urban areas, gaps between roads are often occupied by buildings and other obstacles to radio communication, so communication along roads is sometimes necessary. Vehicles generally move at higher rates of speed than nodes in other kinds of MANETs. Speeds of vehicles moving in the same direction are generally similar, and they can therefore remain in radio contact with one another for much longer periods

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 549–555.
springerlink.com

of time than with vehicles moving in the opposite direction. Even though several unicast routing protocols have been developed for MANETs because of the unique characteristics of VANETs, these protocols cannot be directly used in VANETs efficiently. Hence, because of the expected potential impact of VANETs, several researchers have developed routing protocols that are suitable for VANETs. Simulation and testing of routing protocols for VANETs presents its own challenges in reproducing realistic vehicular motion on a simulated environment.

2 Considered Routing Protocol for VANETs

VANETs consist of mobile nodes having dynamic topology; the mechanism of finding, maintaining and using routes for communication is not trivial for fast moving vehicles. Short lifetime of communication link, less path redundancy, unpredictable node density and strict application requirements make routing in VANETs quite challenging. In the related and similar domain of MANETs, there has been extensive research about the routing protocols during the past decade. Because MANETs and VANETs have many similar characteristics, early prototypes and studies about VANETs made use of the routing protocols developed for MANETs, however there is a lack of a systematic comparison and performance evaluation study that presents conclusive results about the performance of both reactive and proactive routing protocols in VANET environment. Ad hoc On Demand Distance Vector (AODV), Dynamic MANET On Demand (DYMO) and Optimized Link State Routing (OLSR) protocols are considered for the performance evaluation over proposed mobility model.

AODV is a well know distance vector routing protocol and works as follows. Whenever a node wants to start communication with another node, it looks for an available path to the destination node in its local routing table. If there is no path available, then it broadcasts route request (RREQ) message to its neighborhood. Any nodes that receives this message looks for a path leading to the destination node. If there is no path then, it re-broadcasts the RREQ message and sets up a path leading to RREQ originating node. This helps in establishing the end to end path when the some node receives route reply (RREP) message. Every node follows this process until this RREQ message reaches to the destination node itself. Either way the RREQ receiving node will send a RREP to the sender of RREQ message. In this way, the RREP message arrives at the source node, which originally issued RREQ message.

Dynamic MANET On demand (DYMO) Routing protocol is another reactive routing protocol that works in multi hop wireless network. It is currently being developed in the scope of IETF's MANET working group and is expected to reach RFC status in the near future. DYMO is considered as a successor to the AODV routing protocol. DYMO has simple design and is easy to implement. The basic operations of DYMO protocol are route discovery and route maintenance. When a node wants to discover a path to a destination node. It initiates the route discovery operation. A RREQ message is broadcast to the network. Every intermediate node participates in hop-by-hop dissemination of

this message and records a route to the originator. When a destination node receives this RREQ message, it responds with a RREP message unicast towards the originating node. Every node receiving this message creates a route to the destination node and eventually this RREP message arrives at the originator of the RREQ message.

Optimized Link State Route (OLSR) Protocol is an optimization of the classical link state algorithm adapted for the use in wireless ad hoc networks. In OLSR three levels of optimization are achieved. First, few nodes are selected as multipoint relays (MPRs) to broadcast the message during the flooding process. This is in contrast to what is done in classical flooding mechanism, where every node broadcast the message and generates too much overhead traffic.

3 Mobility Models

As far as Vehicular Ad-hoc Networks (VANETs) are concerned, the research started to use more realistic models. The simple Freeway model and Manhattan (or Grid) model were the initial steps, then more complex projects were started involving the generation of mobility patterns based on real road maps or monitoring of real vehicular movements in cities. However, in most of these models, only the macro-mobility of nodes was considered. Although car-to-car interactions are a fundamental factor to take into account when dealing with vehicular mobility, little or no attention was paid to micro-mobility.

The studies of mobility patterns in different scenario's are depending on the movement of node with respect to their speed and velocity. The linked parameters remained part of study in separation for any working ambiguity. Most of these working patterns and models are highly regarded in the following literature.

According to Chia-Chen (2008), the architectural models for carrying reliable vehicle-to-vehicle services in an unreliable VANET environment has a range of factors. These variable factors of VANET are due to its multi-hop delivery mechanism with different network involvements. Therefore, emphasizing on single network, instead, concept of heterogeneous vehicular network (HVN) is proposed. The clusters of VANET with WMAN (wireless metropolitan area network) are incorporated with each other for observing different mobility patterns. The induction of Mobility Patterns Aware Routing Protocol (MPARP) and HVN gave progressive outcomes. The simulation results also show the credibility of enhancement in terms of packet delivery ratio (PDR), number of links break, and instant throughput and delay performance of the communication mediums. Similarly in Choffnes (2005), another mobility model is proposed for dynamic mobile nodes movement with concentration on metropolitan areas. It provide some achievable trials for C3 (car-to-car cooperation) project. To include the level of realistic features, actual city maps were used for real-time consequence in the metropolitan areas. The proposed model named STRAW (Street Random Waypoint) also evaluated the routing performance in the ad-hoc networks. In comparison of two main routing protocols DSR and AODV with respect to packet delivery ratio gives the clear picture of vehicular diversification of on the road networks. There is also a comparative study available in Djenouri

(2008) discussing different VANET mobility models like; Freeway, Manhattan, city Section Models (CSM), Stop Sign Model (SSM), and STRAW for some positive mobility consideration with different tools.

After studying above mobility model, we are considering city traffic scenario to generate realistic mobility model. The Scenario shows the common urban settings found in the city traffic. The two lanes road is created. This model is considering the main road pattern, vehicle speed, lane change and fast vehicle can overtake the slow running vehicle and is simulated using NS-2 & SUMO taking the city of Bareilly, U.P., India as a region instance to generate the movement patterns of vehicles. The Google earth map of any region is available on the internet for public use. SUMO easily converted these maps to generate movement patterns of vehicles at every timestamp. Using SUMO, movement patterns of variable number of vehicle were generated randomly. The specified regions within the movement files and their topology outlines from the Google maps are shown in Figure 1. This is the simple method of obtaining movement patterns of vehicle in the real world

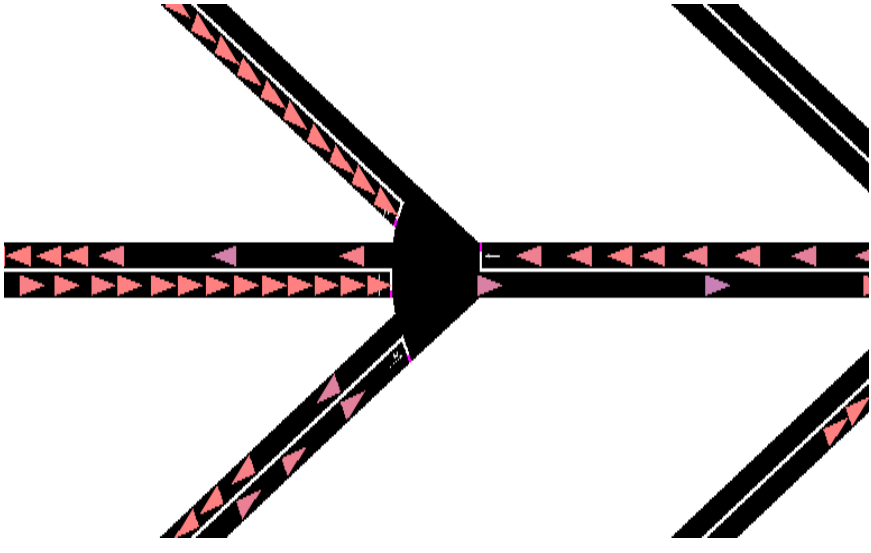


Fig. 1. Mobility Model

4 Simulation Parameter

The simulation model was based on the Network Simulation (NS2) and VANET. The network simulator (NS2) is a simulation tool for replicating real life networking environment and their working and adjoining standards respectively. It works with the combination of different development tools and language because of its environment of open source possessiveness. Mainly the backend object oriented and scripting language used by this simulator are the,"C++" and "TCL". SUMO is an open source, portable microscopic road traffic simulator. It allows the user to build a road topology and can import different readymade map formats of many cities.

An unslotted carrier sense multiple access with collision avoidance (CSMA/CA) is used for data transmission in MAC layer. The radio model uses characteristics similar to a commercial radio interface. In the simulation study, the Ad-hoc On-demand distance Vector (AODV), Dynamic MANET On demand (DYMO) and Optimized Link State Routing (OLSR) was used as the routing protocol. Table 1 provides all the simulation parameters of Realistic Mobility Models.

Table 1. Simulation parameter values

Time of Simulation	120 seconds
Routing Protocol	AODV, DYMO and OLSR
Number of Nodes	100
Network Interface	Wireless
Bandwidth	2 MB
Traffic Type	CBR
Maximum Packet in Queue	50
MAC Protocol Type	IEEE802.11
Packet Size	1500 Bytes
Area Size	1000*1000
Speed	60 kph -100kph

5 Simulation Results

The simulation results shows the ratio of the data packets successfully received at the destination and total number of data packets generated at source. Figure 2 shows the effects of node density on the packet delivery ratio of AODV, DYMO and OLSR in scenario. Here we observe that AODV performs better as compared to OLSR and DYMO routing protocol.

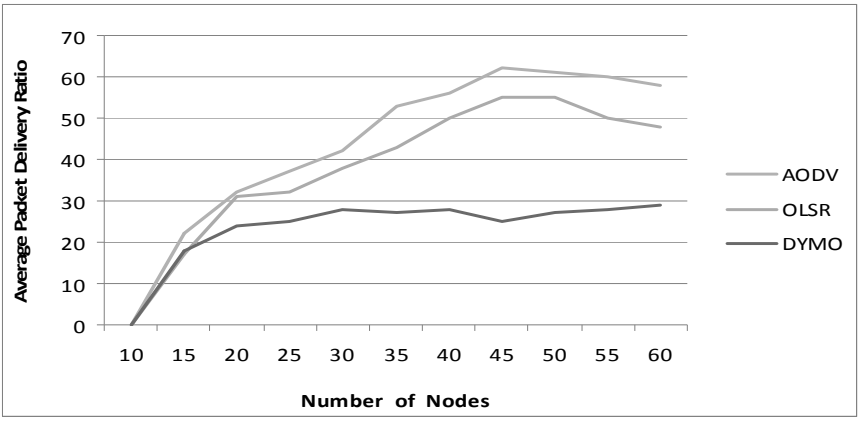


Fig. 2. Average Packet Delivery Ratio

Figure 3 shows the end-to-end delay of AODV, DYMO and OLSR in VANET using realistic mobility model. We observe that reactive routing protocol have higher end-to-end delay than proactive routing protocols. AODV outperforms DYMO when the number of node increases in this scenario. It is suggested to use realistic mobility model to evaluate the performance of routing protocols.

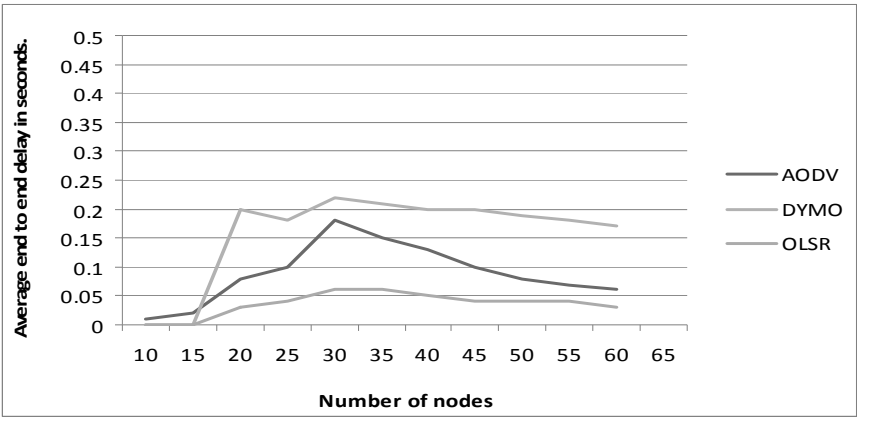


Fig. 3. Average End-to-End delay

6 Conclusion

This paper presents a simulation study of realistic mobility model on the performance study of VANET that uses Ad-hoc On- Demand Distance Vector (AODV), Dynamic MANET On demand (DYMO) and Optimized Link State Routing (OLSR) was used as the routing protocol, which is well suited for dynamic vehicular network and provide the reliable path for V2V communication. The performance result using realistic mobility model is suitable. At different levels of randomness setting in this model have no effect on the accuracy of packet delivery ratio and end-to-end delay.

References

[1] Bergh, A.E., Venture, N.: Movement Prediction Assisted Fast Handovers for IP Mobility. IEEE (2007)

[2] Pazand, B., Mc Donald, C.: A Critique of Mobility Models for Wireless Network Simulation. In: 6th IEEE/ACS International Conference on Computer and Information Science (ICIS 2007). University of Western Australia (2007)

[3] Checkers, Perkins, C.: Dynamic MANET On-demand Routing Protocol. Internet Draft (March 2006)\

[4] Chan, W., Sim, M., Lee, S.: Performance Analysis of Vehicular Ad hoc network with Realistic Mobility Pattern. In: Proceeding of 2007 IEEE International Conference on Telecommunications and Malaysia, International Conference on Communication, Penang, Malaysia, pp. 14–17 (May 2007)
[5] Fall, K., Varadhana, K.: The ns-Manual (formerly ns Notes and Documentation) (January 25, 2005)
[6] Zimmermann, H.M., Gruber, I.: A Voronoi- based Mobility Model of Urban Environments. In: Proc. of the European Wireless 2005, EW 2005 (April 2005)
[7] Mahajan, A.: Evaluation of Mobility Models for Vehicular Ad-hoc Network Simulation, Technical Report N.051220, Florida State university (2005)
[8] Uchiyama, A.: Mobile Ad-hoc Network Simulator based on Realistic Behavior Del. In: Demosessionin MobiHoc 2005 (2005)
[9] Zhou, B., Xu, K., Gerla, M.: Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks. In: Proc. of IEEE MILCOM 2004 (2004)
[10] The Obstacle Mobility Model, http://moment.cs.ucsb.edu/mobility/
[11] Saha, A.K., Johnson, D.: Modeling Mobility for Vehicular Ad-hoc Networks. In: Poster Session, 1st ACM Workshop on Vehicular Ad hoc Networks (VANET 2004) (October 2004)
[12] Clausen, T., Jacquet, P. (eds.): Optimized Link State Routing Protocol (OLSR), RFC 3626. Network Working Group (October 2003)
[13] Haerri, J., Filali, F., Bonnet, C.: Performance Comparison of AODV and OLSR in VANETs Urban Environment under Realistic Mobility Patterns. In: Proceedings of the 5th IFIP Mediterranean Ad hoc Networking Workshop, Lipari, Italy, pp. 14–17 (June 2006)
[14] Jaap, S., Bechler, M., Wolf, L.: Evaluation of Routing Protocols for Vehicular Ad hoc Networks in City Traffic Scenarios. In: Proceedings of the 5th International Conference on Intelligent Transportation Systems Telecommunications (ITS), Brest, France (June 2005)
[15] James, B., Manivanna, D.: Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification. Science Direct Passiver and Mobile Computing (2009)
[16] Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand Distance Vector (AODV) routing. RFC3561. Network Working Group (2003)
[17] Abedi, O., Fathy, M., et al.: Enhancing AODV routing protocol using mobility parameters in VANET. In: IEEEIACS International Conference on Computer System and Applications, AICCSA (2008)
[18] Djenouri, D., Soualhi, W., et al.: VANETs Mobility Models and Overtaking: An Overview. In: 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA 2008 (2008)
[19] VANET Projects List, http://www.vanet.info/projects
[20] Fall, K., Varadhan, K.: The ns- Manual (formerly ns Notes and Documentation) (January 25, 2005)

SEVO: Bio-inspired Analytical Tool for Uni-modal and Multimodal Optimization

Hema Banati[1] and Shikha Mehta[2]

[1] Department of Computer Science, Dyal Singh College, University of Delhi, Delhi
banatihema@hotmail.com
[2] Department of Computer Science, University of Delhi, Delhi
mehtshikha@gmail.com

Abstract. With the rising success of bio-inspired algorithms to solve combinatorial optimization problems, research is focused towards design of new bio-inspired algorithms or new variants of existing algorithms. To validate the reliability of new algorithms with respect to the existing algorithms, they are tested using benchmark test functions. However existing automated tools with benchmark test problems are limited to genetic and evolutionary algorithms only. Therefore large group of researchers have to repeatedly write the same code for the existing algorithms along with their own proposed version. To address this need, the paper presents a unified swarm and evolutionary optimization (SEVO) tool that automates established algorithms like genetic algorithm (GA), memetic algorithm (MA), particle swarm optimization (PSO) and shuffled frog leaping algorithm (SFLA) over sixteen benchmark test functions with diverse properties. SEVO tool provides a user friendly interface to the users for input parameters, options to simultaneously execute any combinations of algorithms and generate graphs for comparison. To substantiate the effectiveness of SEVO tool, experiments were performed to compare the abilities of GA, MA, PSO and SFLA to attain global minima and speed of convergence. Results establish that convergence rate of SFLA is significantly better than PSO, MA and GA. SFLA also outperformed PSO, MA and GA in attaining global minima. Thus SEVO toolbox may serve as an imperative aid for the bio-inspired research community to perform simulations of the embedded algorithms over varied classes of optimizations test problems with minimum time and effort.

Keywords: Evolutionary and swarm optimization toolbox, Benchmark test functions tool, Genetic algorithm tool, Memetic Algorithm tool, Particle swarm optimization tool, Shuffled frog leaping algorithm tool.

1 Introduction

Biologically inspired computation is a multi-disciplinary field that mimics the metaphor of natural biological evolution to provide solutions to a variety of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 557–566.
springerlink.com

intricate problems. The intent of this field is to devise mathematical and engineering tools to generate solutions to computationally hard problems. The remarkable success of bio-inspired techniques has motivated the researchers to explore new algorithms in this field. Researchers also attempt to improve the performance of conventional algorithms with new variants. To validate the reliability of newly proposed optimization algorithms, they are evaluated over the benchmark test functions. Test functions serve as important tools to corroborate the reliability and robustness of new optimization algorithms with respect to various existing optimization algorithms. Ali, Pant and Abraham [1] proposed ant colony differential evolution algorithm and validated it on a test bed of seven benchmark problems. Chauhan, Deep and Pant [2] evaluated their proposed variant of PSO with conventional PSO over ten benchmark test functions. Pant, Ali and Singh [11] used ten test problems to evaluate the impact of parent-centric approach on the performance of classical differential evolution. Shen, et. al. [12] evaluated the performance of BFOA and BF-PSO on 23 numerical benchmark functions. Lu, Ge and Gao, [8] assessed self-adaptive genetic algorithms (SaGA) over seven benchmark problems. Elbeltagi, Hegazy and Grierson [3] compared the performance of genetic algorithm, memetic algorithm, ant colony algorithm and shuffled frog leaping algorithm over two continuous benchmark problems.

To implement these benchmark test problems using evolutionary and swarm techniques, users require certain programming expertise with considerable time and effort to write a computer program. This task could be tedious as users not only have to implement their own proposed version of the evolutionary algorithm but also the existing algorithms for comparison. The rapid development of computer technology, reduced cost and the increased computing power available nowadays has encouraged the development of computer-aided optimization tools to provide virtual problem-solving environment for sophisticated optimization problems. Table1 depicts the various tools that automate the execution of swarm and evolutionary algorithms to reduce user's time and efforts.

Table 1. Comparison of optimization toolboxes

Toolbox	**Platform**	**Algorithms Automated in the respective toolbox**			
		GA	**MA**	**PSO**	**SFLA**
GEATbx[18]	Matlab[19]	Yes	No	No	No
GOAT[5]	Matlab	Yes	No	No	No
FlexToolGA[16]	Matlab	Yes	No	No	No
PSO toolbox [17]	Matlab	No	No	Yes	Yes
MOEA[13]	Matlab	Yes	No	No	No
SEVO(Proposed)	**Java**	**Yes**	**Yes**	**Yes**	**Yes**

Majority of the optimization toolboxes discussed in table1 provide automated support either for genetic algorithms or particle swarm optimization only. For rest of the algorithms like memetic algorithm and shuffled frog leaping algorithm etc users have to write their own set of code. Thus there is no unified toolbox that supports automated execution of all the algorithms. In addition, these tools require Matlab toolkit as pre-requisite which is not available freely. To address this need, this paper presents a unified swarm and evolutionary optimization (SEVO) toolbox that automates genetic algorithm, memetic algorithm, particle swarm optimization and shuffled frog leaping algorithm. Tool presents a single platform for users to compare, examine or analyze varied swarm and evolutionary algorithms for unimodal and multimodal optimization. In the present work, SEVO toolbox has been used to compare the performance of genetic algorithm, memetic algorithms, particle swarm optimization and shuffled frog leaping algorithm to achieve global minima and evaluate convergence speed over the sixteen benchmark test functions embedded in the toolbox. Rest of the paper is organized as follows: Section 2 presents design of evolutionary and swarm intelligence optimization (SEVO) toolbox. Section 3 presents experiments and results performed using the toolbox followed by conclusion and future work in section 4.

2 Design of Swarm and Evolutionary Optimization (SEVO) Toolbox

The design of SEVO toolbox as shown in fig1 presents an easy to use interface where users need to make certain selections for automatic execution of a particular algorithm for the chosen benchmark problem. It has been developed in java using JDK1.6 toolkit which is available freely. It carries all the advantages of java like platform independence, robustness etc. It eliminates the need of writing extra code for evaluating the performance of embedded algorithms over the benchmark test problems.

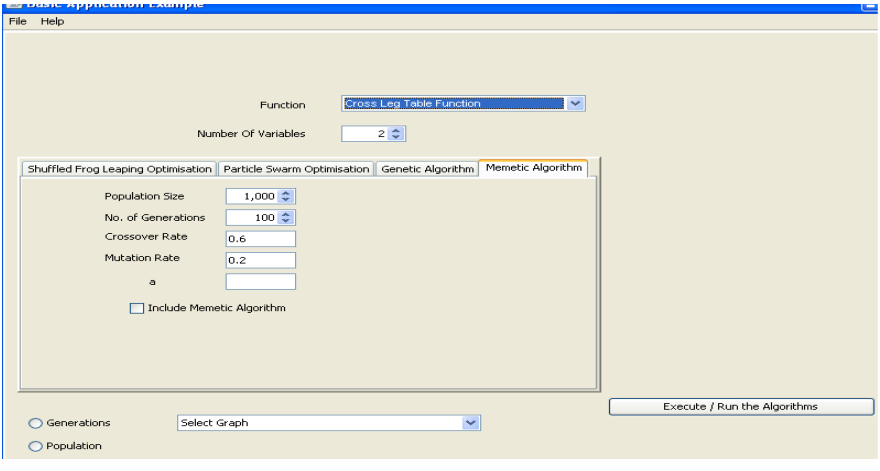


Fig. 1. Design of SEVO toolbox

SEVO toolbox automates the execution of four most prevalent bio-inspired algorithms like genetic algorithm (GA) [9], memetic algorithm (MA) [10], particle swarm optimization (PSO) [6] and shuffled frog leaping algorithm (SFLA) [4]. All these algorithms follow a similar stochastic iterative procedure for generating solution to a particular problem. Accordingly the various common parameters used in the SEVO toolbox are:

- **Initialization:** All bio-inspired techniques are population based which require an initial population of potential solutions to be generated randomly. In SEVO toolbox initial population is generated based on the upper and lower limit range of the benchmark problem selected by the user.
- **Fitness function:** It is used to evaluate the quality of a solution in the population. It reflects the objective of the test function selected by the user. In SEVO toolbox, all functions are minimization problems.
- **Convergence criteria/Number of generations:** It defines the stopping criteria for an algorithm. The algorithm may either be executed for fixed number of generations or till the fitness of the population does not change for next few generations. Presently in SEVO toolbox, convergence criteria are fixed to number of generations.

Table2 displays the updating rules coded in the SEVO toolbox for evolving the population for next generation for the respective algorithms. It also illustrates the various initial inputs parameters required in the SEVO toolbox for automatic execution of a particular algorithm.

Table 2. Input parameters and updating rules for bio-inspired algorithms

Algorithm	**Initial Input Parameters required in SEVO toolbox**	**Population Updating Rules used in SEVO toolbox**
Genetic Algorithm	Crossover rate and Mutation rate.	Selection, Recombination and Mutation
Memetic Algorithm	Crossover rate, mutation rate, Local search step constant value.	Selection, Recombination and Mutation, Local Search mechanism
Particle Swarm Optimization	Learning factors C1 and C2, inertia weight w, maximum change of a particle velocity Vmax	New Velocity V_i = w * current Vi + C1* rand() * (Pi – Xi) + C2 * Rand() * (Pg-Xi) New position Xi = current position Xi + New Vi; Vmax >=Vi >=Vmax
Shuffled frog Leaping Algorithm	Number of memeplexes, number of memeplex iterations and maximum change in position D_{max}	Change in frog position (D_i) = rand () * (X_b- X_w) New Position $X_w = X_w + D_i$ (D_{max} >= D_i >=-D_{max})

The standard set of test functions[7][14][15] included in the SEVO toolbox belong to varied classes of problems like unconstrained/constrained optimization functions, discrete/continuous functions, separable/non-separable functions and unimodal/multi-modal test functions as listed in fig2. As shown in fig 2, functions F1-F5 are unimodal, F6 is a noisy quartic function where random [0, 1) is a uniformly distributed random variable in [0, 1) and F7-F16 are multimodal test functions. From the list of various test functions, algorithms may be evaluated using any subset of functions. The test functions with diverse properties may be used to substantiate the ability of an algorithm to solve certain type of optimization problems efficiently.

Function	Function	Range	Minima
F_1	**De Jong's function**: $F(x) = \sum x^2$	[5.12,-5.12]	0
F_2	**Axis Parallel Hyper-ellipsoid function:** $F(x) = \sum i * x_i^2$	[5.12,-5.12]	0
F_3	**Rotated hyper-ellipsoid function**: $F(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	[65.536,-65.536]	0
F_4	**Sum of different powers function:** $F(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	[1,-1]	0
F_5	**Rosenbrock's Valley:** $F(x) = \sum_{i=1}^{n-1} \left[100 * (x_{i+1} - x_i^2) + (x_i - 1)^2 \right]$	[2.048,-2.048]	0
F_6	**Quartic function:** $F(x) = \sum_{i=1}^{n} i * x_i^4 + random[0,1)$	[1.28,-1.28]	0
F_7	**Schwefel's function:** $F(x) = \sum_{i=1}^{n} -x_i * \sin\left(-\sqrt{\lvert x_i \rvert}\right)$	[500,-500]	-418.9829
F_8	**Rastrigin's function:** $F(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right)$	[5.12,-5.12]	0
F_9	**Greiwangk's function:** $F(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[600,-600]	0
F_{10}	**Ackley function:** $F(x) = -20\exp\left(0.02 \sum_{i=1}^{n} \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	[32.768, -32.768]	0
F_{11}	**Cross-leg table function:** $f(x) = -\left[\lvert \sin(x_1)\sin(x_2) e^{\lvert 100 - (x_1^2 + x_2^2)^{0.5}/\pi \rvert} \rvert + 1 \right]^{-0.1}$	[-10 10]	-1
F_{12}	**Modified Schaffer function #1:** $f(x) = 0.5 + \frac{\sin^2(x_1^2 + x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$.	[-100, 100]	0
F_{13}	**Sine envelope sine wave function:** $f(x) = \sum_{i=1}^{n-1} \left(\frac{\sin^2\left[\sqrt{x_{i+1}^2 + x_i^2}\right] - 0.5}{(0.001(x_{i+1}^2 + x_i^2) + 1)^2} + 0.5 \right)$	[-100, 100]	Difficult to optimize
F_{14}	**Levy function (#13):** $f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$.	[-10 10]	0
F_{15}	**Cross in tray function:** $f(x) = -0.0001\left[\lvert \sin(x_1)\sin(x_2) e^{\lvert 100 - (x_1^2 + x_2^2)^{0.5}/\pi \rvert} \rvert + 1 \right]^{0.1}$	[-10 10]	-2.06261218
F_{16}	**Giunta function:** $f(x) = 0.6 + \sum_{i=1}^{2} [\sin(\frac{16}{15} x_i - 1) + \sin^2(\frac{16}{15} x_i - 1) + \frac{1}{50} \sin(4(\frac{16}{15} x_i - 1))]$.	[-1,1]	0.0602472184

Fig. 2. List of benchmark functions implemented in tool

The detailed working of the SEVO toolbox is as follows:

To evaluate the performance of a particular algorithm like shuffled frog leaping algorithm over a benchmark test problem like Ackley function, users need to

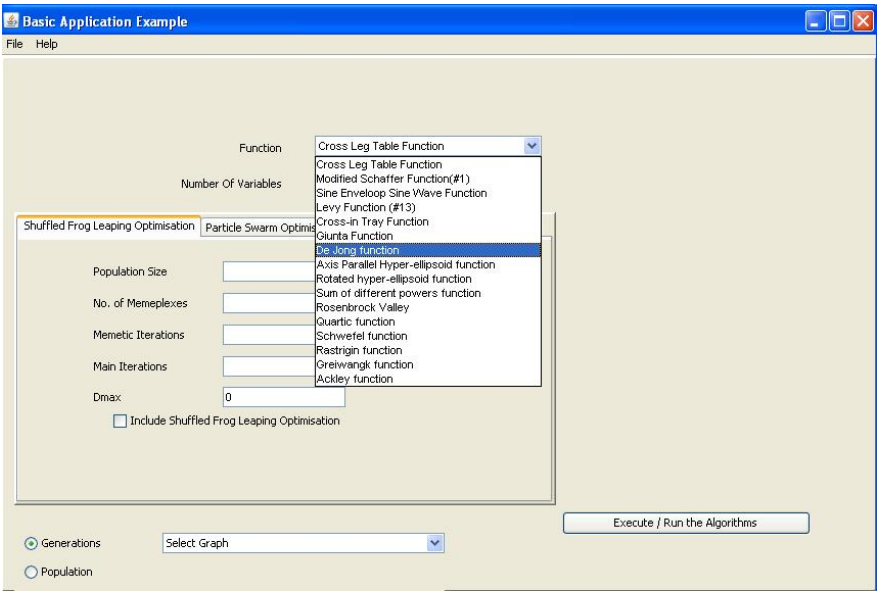


Fig. 3. SEVO toolbox interface for SFLA input parameters

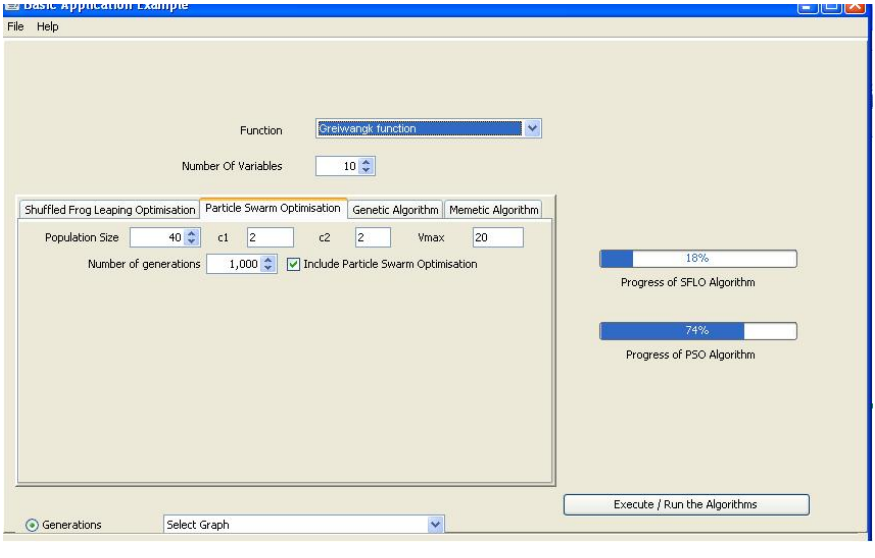


Fig. 4. Execution Progress of SFLA and PSO in SEVO toolbox

select 'Ackley function' from the function dropdown list, enter the number of dimensions, provide population size, number of memeplexes, number of memetic iterations, number of generations and the value of dmax. On submitting all the input parameters for the selected algorithm, users have to click on the execute/run algorithm button to execute the algorithm. Users also have the option to generate graphs by varying number of generations or population as shown in Fig3. For comparative analysis, users have the option to simulate any combination from the given set of algorithms and save results to be used later. Fig4 illustrates the progress of simulating shuffled frog leaping algorithm and particle swarm optimization algorithm simultaneously. As the progress reaches to 100%, graphs are generated automatically. Graphs reflect the performance of the chosen algorithm in terms of mean best fitness over the generations/population according to the option selected by the user.

3 Experiments and Discussion of Results

To illustrate the efficiency of the SEVO toolbox, experiments were performed to compare the performance of genetic algorithm (GA), memetic algorithm (MA), particle swarm optimization (PSO) algorithm and shuffled frog leaping (SFLA) algorithm over the benchmark test functions embedded in SEVO toolbox. Experiments were performed to evaluate rate of convergence and ability to attain global minima for all the above mentioned algorithms.

Experiment 1: It was done to evaluate the relative abilities of GA, MA, PSO and SFLA to attain global minimum for functions F1 to F16 included in the SEVO toolbox. Results as shown in Table 3 depict the mean best and standard deviation obtained for 20 dimensions over 20000 generations and 25 independent runs. It illustrates that for all functions, mean best of memetic algorithm is better than genetic algorithms except for functions F3, F4, F7, F8, F10 and F12 where genetic algorithm performs slightly better than memetic algorithm. For all unimodal and multimodal functions F1 to F16, particle swarm algorithm outperforms both genetic and memetic algorithm in attaining the global minima. Shuffled frog leaping algorithm which integrates the benefits of both memetic and particle swarm optimization algorithm, significantly enhances the global minima further as compared to genetic, memetic and particle swarm optimization algorithm.

Experiment 2: It was performed to analyze the convergence speed of GA, MA, PSO and SFLA over most commonly used Ackley function (F_{10}). Experiment was carried out by plotting the mean best value after every 100^{th} generation for GA, MA, PSO and SFLA for 20 dimensions with number of function evaluations fixed to 20000. Mean best value obtained for 25 independent runs over every 100^{th} function evaluation is shown in Fig5.

Table 3. GA,MA,PSO and SFLA mean best and standard deviation using SEVO tool

Function	GA		MA		PSO		SFLA	
	Mean Best	Std	Mean Best	Std	Mean Best	Std	Mean Best	Std
F1	77.02455	1.42375	**76.5133795**	1.515909	**58.90391942**	3.590444	**13.0775109**	1.790724
F2	791.8911	20.03245	**785.645751**	22.00939	**560.5527486**	6.59E-05	**423.486124**	40.75674
F3	**67169.29**	4876.593	67474.7847	5548.939	**26187.74629**	4054.389	**6560.36209**	547.2211
F4	**0.38743**	0.030433	0.39292239	0.020783	**0.150828085**	0.031577	**0.00085841**	0.000381
F5	-1495.7	39.6346	**-1498.85457**	44.29488	**-2026.35005**	116.5491	**-5548.252**	257.4728
F6	38.05731	1.258916	**37.7470581**	1.280532	**24.57710921**	2.258082	**17.5836857**	0.856378
F7	**-281.703**	52.39501	-278.877692	40.31424	**-875.063331**	146.364	**-2114.7019**	20.58836
F8	**269.6093**	2.196916	270.552469	2.420015	**231.5861595**	11.75014	**218.432454**	9.722658
F9	265.3374	6.275616	**264.89992**	766.275	**201.1629075**	17.83263	**104.178035**	11.63619
F10	**6.105897**	0.045723	6.10750605	0.045784	**5.534116993**	0.16953	**4.39904569**	0.016469
F11	-0.00041	1.52E-05	**-0.00040714**	1.51E-05	**-0.00063531**	0.000111	**-0.0020613**	3.82E-05
F12	**0.499797**	3.99E-05	0.49981984	2.57E-05	**0.499357119**	0.000148	**0.49673954**	0.000105
F13	9.321048	0.033543	**9.30925545**	0.033301	**8.873276342**	0.146664	**7.78418523**	0.022681
F14	459.2544	12.05468	**456.625987**	11.05298	**347.1873426**	26.5526	**175.007858**	3.664865
F15	-0.38361	0.009616	**-0.38411912**	0.010506	**-0.53122621**	0.043771	**-0.854325**	0.005999
F16	-2.02774	0.033532	**-2.03725185**	0.031028	**-2.48233745**	0.111541	**-3.2015356**	7.327362

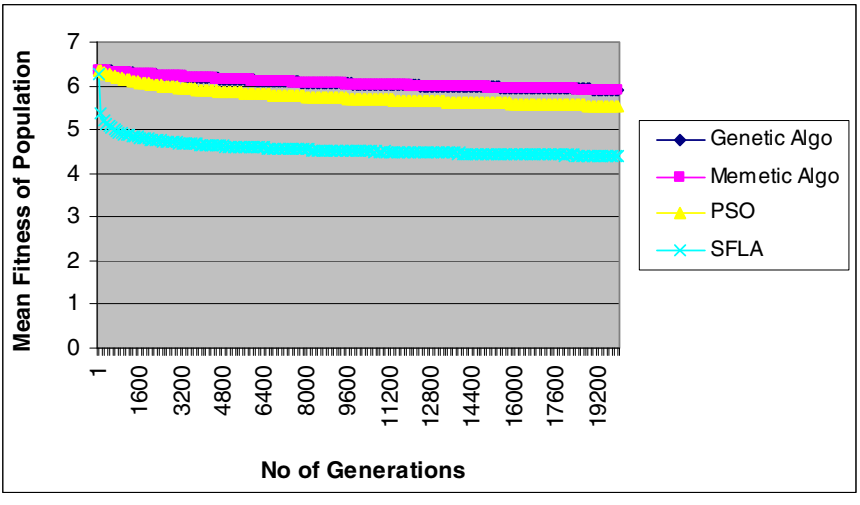


Fig. 5. Convergence speed of GA, MA, PSO and SFLA and over Ackley function

Fig5 illustrates that convergence speed of memetic algorithm is slightly better than genetic algorithm whereas particle swarm optimization algorithm convereges faster than both memetic and genetic algorithm. As depicted in fig5, rate of convergence of shuffled frog leaping algorithm is significantly better than GA,MA

and PSO. Therefore SFLA may be a good choice for solving unimodal and multimodal problems in the real world

4 Conclusion

The paper presented unified bio-inspired toolbox -SEVO to automate the execution of most widespread swarm and evolutionary algorithms such as genetic algorithm, memetic algorithm, particle swarm optimization and shuffled frog leaping algorithm over diverse categories of benchmark test functions. It presents a user friendly interface where naïve users need to select the test function and algorithm and enter only the algorithm specific input parameters to generate the results. It relieves the bio-inspired community from the hassles of writing code for the existing algorithms. To establish the reliability of the SEVO tool, experiments were performed to analyze the convergence speed and ability to attain global optima for all included algorithms and benchmark test problems. Results illustrate that shuffled frog leaping algorithm performs noticeably better than particle swarm optimization, memetic and genetic algorithm. Future work includes further enhancement of SEVO tool by integrating other evolutionary and swarm techniques like ant colony optimizations, bacteria foraging, and artificial bee colony algorithm along with other classes of bench mark test functions.

References

[1] Ali, M., Pant, M., Abraham, A.: A Hybrid Ant Colony Differential Evolution and its Application to Water Resources Problems. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE (2009)

[2] Chauhan, P., Deep, K., Pant, M.: Power Mutation Embedded Modified PSO for Global Optimization Problems. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 139–146. Springer, Heidelberg (2010)

[3] Elbeltagi, E., Hegazy, T., Grierson, D.: Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics 19(1), 43–53 (2005)

[4] Eusuff, M.M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management 129(3), 210–225 (2003)

[5] Houck, C., Joines, J., Kay, M.: The Genetic Algorithm Optimization Toolbox (GAOT). North Carolina State University, Department of Industrial Engineering, NCSU-IE TR 95-09 (1995), http://www.ie.ncsu.edu/gaot/

[6] Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)

[7] Liang, J.J., Suganthan, P.N., Deb, K.: Novel Composition Test Functions for Numerical Global Optimization. In: Proceedings Swarm Intelligence Symposium. IEEE (2005)

[8] Lu, F., Ge, Y., Gao, L.: A Novel Genetic Algorithm with Multiple Sub-population Parallel Search Mechanism. In: 2010 Sixth International Conference on Natural Computation, ICNC. IEEE (2010)
[9] Mitchell, M.: An Introduction to Genetic Algorithm. The MIT Press (1999)
[10] Moscato, P., Cotta, C.: A Gentle Introduction to Memetic Algorithms. In: Handbook of Meta-heuristics, pp. 1–56. Kluwer, Dordrecht (1999)
[11] Pant, M., Ali, M., Singh, V.P.: Parent-centric differential evolution algorithm for global optimization problems. Opsearch 46(2), 153–168, doi:10.1007/s12597-009-0010-5
[12] Wu, C.-G., Zhang, N., Jiang, J.-Q., Yang, J., Liang, Y.: Improved Bacterial Foraging Algorithms and Their Applications to Job Shop Scheduling Problems. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 562–569. Springer, Heidelberg (2007)
[13] Tan, K.C., Lee, T.H., Khoo, D., Khor, E.F.: A Multiobjective Evolutionary Algorithm Toolbox for Computer-Aided Multiobjective Optimization. IEEE Transactions on Systems, Man and Cybernetics. PartB: Cybernetics 31(4) (2001)
[14] Yang, X.S.: Test problems in optimization. In: Yang, X.-S. (ed.) Engineering Optimization: An Introduction with Metaheuristic Applications. John Wiley & Sons (2010)
[15] Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation 3(2), 82–102 (1999)
[16] Flex Tool (GA) (1999), http://www.flextool.com/
[17] http://www.mathworks.com/matlabcentral/fileexchange/7506
[18] Pohlheim, H.: Genetic and Evolutionary Algorithm Toolbox (GEATbx) for Use with Matlab (1998), http://www.geatbx.com
[19] Using MATLAB, ver.5. The MathWorks, Inc., Natick (1998)

Quantitative and Qualitative Analysis of Unmanned Aerial Vehicle's Path Planning Using Master-Slave Parallel Vector-Evaluated Genetic Algorithm

Djamalladine Mahamat Pierre, Nordin Zakaria, and Anindya Jyoti Pal

High-Performance Computing Center, Universiti Teknologi PETRONAS
djamal2810@gmail.com, nordinzakaria@petronas.com.my,
anind.jyotipal@petronas.com

Abstract. The demand of Unmanned Aerial Vehicle (UAV) to monitor natural disasters extends its use to multiple civil missions. While the use of remotely control UAV reduces the human casualties' rates in hazardous environments, it is reported that most of UAV accidents are caused by human factor errors. In order to automate UAVs, several approaches to path planning have been proposed. However, none of the proposed paradigms optimally solve the path planning problem with contrasting objectives. We are proposing a Master-Slave Parallel Vector-Evaluated Genetic Algorithm (MSPVEGA) to solve the path planning problem. MSPVEGA takes advantage of the advanced computational capabilities to process multiple GAs concurrently. In our present experimental set-up, the MSPVEGA gives optimal results for UAV.

Keywords: Path-planning, UAV, Genetic Algorithm, Contrasting Objectives, Multi-objective (Times; 10pt).

1 Introduction

Unmanned Aerial Vehicles (UAVs), also referred to as drones, are pilotless aircraft that are either remotely controlled or travel autonomously as self-directing entities. Primarily used in military for intelligence gathering through surveillance and reconnaissance, the use of UAVs can be expanded to monitor natural disasters or explore dangerous areas that present hazards to human beings. Recently, during the humanitarian crisis caused by the earthquake in Haiti, drones have been successfully used to provide ground-based first responders with highly details imaginary of the most affected zones.

Although the use of drone for information gathering reduces human casualties in dangerous zones, it is reported that a high percentage of drone-related accidents are caused by human factor errors [1]. Our research is motivated by the idea to fully automate the drones. The research is focused on exploring the possibilities of replacing the human operators by a scalable computer program capable of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 567–577.
springerlink.com

navigating the drones in response to the environments' geometry (obstacles avoidance) and the mission objectives (information gathering). In other words, we are designing a path-planning algorithm for the drone.

Path planning is the process of determining a collision free pathway between a drone's current position and its destination. Several approaches to path planning have been introduced over the past years. Those approaches include cell decomposition [11], potential field [4], and road map [10]. However, those approaches fail to efficiently solve multi-objective problems. Because of its robustness and flexibility to support multi-objectivity, Genetic Algorithm (GA) is used as an optimization tool for path planning algorithms [8]. However, conventional genetic algorithm fails to provide satisfactory results in the case of contrasting objectives. We use a modified master-slave parallel vector-evaluated genetic algorithm to solve our path planning problem. Our algorithm takes advantage of the "independent local evolution" described by Parsopoulos et. al. [9], and the parallelism described in [13].

2 Related Works

Over the past decades, path planning has become an important component of robotics [7] and virtual reality (for camera navigation [14]). Several paradigms to path planning have been proposed.

Potential field is a technique that assigns attraction values to the destination and repulsive values to obstacles. While potential field allows the robots to navigate safely towards its destination, it fails to prevent the robot from reaching local minima [2]. Local Potential field method described in [3] generates a collision free pathway by connecting a local minimal to a potential field configured in the robot's search space. A similar approach is used to determine a collision free pathway for an Autonomous Underwater vehicle [4]. Although the above cited methods [3] [4] are free of local minima, the resulting pathway is exclusively applicable to navigation; thus, multi-objectivity is not taken into account.

Ross Graham [6] exploited path finding algorithm using neural network. Leigh et. al [5] describes an alternative to A*-Like path finding using genetic algorithm. Their algorithm appears more efficient with multiple agents. Unfortunately, it relies on 2D testing ground. Expanding the use of the algorithm to 3D environment does not necessarily produce the same success level. Furthermore, Ross Graham favors genetic algorithm over neural network, for multi-objective applications [6].

Hagen et al.[7] provides a path planning approach for mobile robots using genetic algorithm. Sensors are used to locate the obstacles while the obstacle response is computed locally. However, this algorithm has some shortcomings. While the robots operate in a 3D real world space, the path planning is computed for a 2D world assuming that the robots operate on a flat floor. Furthermore, Hagen's algorithm is only concerned with the path's computation. The robot's operation is, therefore, limited to travelling between a start point and a destination point.

Shashi et. al [8] expand the use of simple evolutionary algorithm to path planning by using a multi-objective GA approach to solve the path planning problem for Unmanned Aerial Vehicles. Using a simple Objective Weighting P(γ) (variation of Multi-objective genetic algorithm), the algorithm does not necessarily provide a satisfactory result in case of contrasting objectives. The algorithm provides a single solution. That solution can either be a middle-ground between all the potential solutions (if the weights of the objectives are equal), or converges towards a single objective (the objective with the highest weight).

There are variations of multi-objective genetic algorithm that can solve the contrasting objectives problems. Multi-objective Optimization using Parallel Vector-Evaluated Particle Swarm Optimization (PVEPSO) [9] is a method inspired by Vector-Evaluated Genetic Algorithm. In PVEPSO, each swarm operates independently, with its own population evaluated according to a single objective. After a certain number of iterations, particles from each swarm migrate to the neighboring swarm, providing variety to local swarm. A series of tests conducted on four well known problems showed that PVEPSO outperforms Vector- Evaluated Genetic Algorithm (VEGA) at each case. We introduce a modified Master-Slave Parallel-Vector Evaluated Genetic Algorithm (MSPVEGA) that inherits some features of PVEPSO and the parallelism of Master-Slave Parallel Genetic Algorithm (MSGA) described in [13] to solve our path planning problem.

3 Parallel Vector-Evaluated GA

Genetic algorithm (GA) is a search heuristic that imitates evolution in nature. A potential solution, or individual, is represented by a chromosome. Each chromosome may contain several genes representing characteristics of the individual. Starting from a couple of potential solutions, the individuals go through some transformations to generate new and fitter individuals. The transformations include reproduction, crossover and mutation. Reproduction involves making a copy of a chromosome. Crossover changes the content of individual by swapping the values of genes between two chromosomes. Mutation, on the other hand, alters the value of a gene to generate a new individual. Mutation adds variety to the population and reduces the probability of occurrence of premature convergence. If an individual is judged unfit in the process, it is simply discarded. The judgment on fitness of the individuals is based on a value (fitness value) computed using a fitness function. The fitness function represents the objectives of the problem in hand.

Master-Slave Parallel Vector-Evaluated Genetic Algorithm is based on master-slave strategy. In MSPVEGA, the master coordinates the migration while the slaves run the parallel GAs. All GAs execute concurrently. Each GA operates with its own population which is evaluated with a single objective. Fig.1 depicts the building blocks of MSPVEGA.

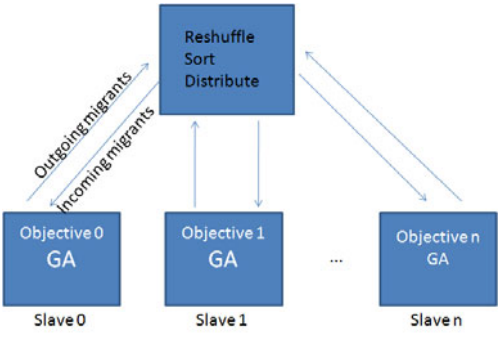


Fig. 1. MSPVEGA Building Blocks.

After a predefined number of iterations, a selected subset of the local population migrates from the local GA to the master node. The master node reshuffles all the received individuals, segregates them according to the objective with which they perform better. Finally, the segregated groups are sent back to the appropriate slaves as shown in Fig. 2.

Within each GA, the fitness function with which each individual is evaluated as a mathematical representation of a particular objective. Given a number of objectives, α, MSPVEGA dynamically configures *b+1* slaves indexed from *0* to *b*; the slave indexed by *b* runs a particular GA that combines all the objectives in Objective Weighting Genetic Algorithm. The fitness function corresponding to such objective is given in (1).

$$f = \sum_{i=0}^{b-1} fi \tag{1}$$

where f_i represents the fitness function of the GA running at the slave *i*.

4 Representation

Our testing ground is a virtual environment. In the virtual world, there are several sound emitters placed at various locations. Each emitter dispatches a sound at a particular frequency. The frequency of emission of each emitter is known to the UAV's configuration. From its current position, the mission of the UAV is to reach a preset destination position as soon as possible. Along the way, the UAV should capture as much distinctive information as possible and avoid all the obstacles.

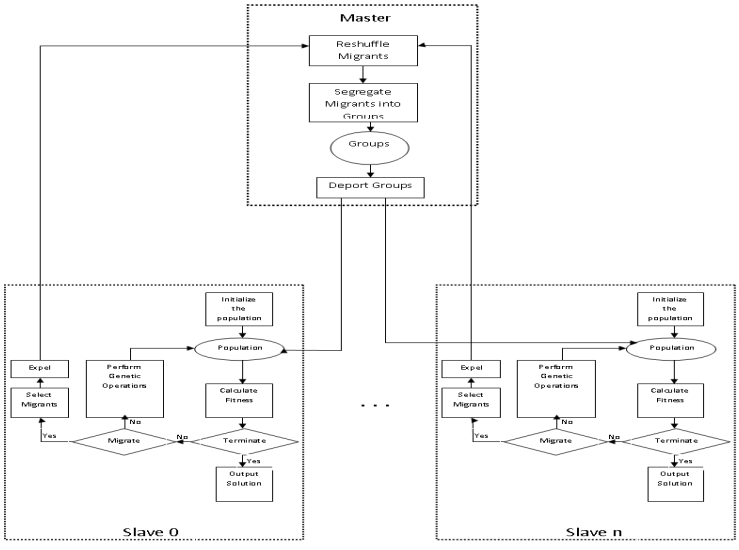


Fig. 2. MSPVEGA flowchart.

In this problem, chromosomes represent both the pathway, and the speed of the UAVs at each discrete point of the pathway. In order to reduce the length of the chromosome to speed up the iterations, the section of the chromosome that represents the pathway contains the information of control points of B-Spline curves. The resulting curve is represented as a set of discrete points. At each discrete point, the drone adjusts its speed to fulfill its mission.

We are presented with a multi-objective time-constrained optimization (maximization) problem. The fitness function corresponding to each objective is described in the following lines:

- f_1 is the ratio of the information captured to the maximum amount of information. It ensures that the drone captures as much information as possible as it flies from its current position to its destination. The longer the drone flies over an emitter, the more information it captures.
- f_2 is the ratio of the current speed to the maximum speed. This objective regulates the speed of the drone in accordance to the information being captured. The drone slows down near an emitter, and speeds up as the sounds transmitted by the emitter fades away.
- f_3 is the ratio of the distance of current pathway by the distance of shortest possible pathway. This function ensures that the resulting pathway is short.
- f_4 validates feasible points of the pathway. This function ensures that the resulting pathway is collision-free.
- f_5 is the ratio of the projected time of the resulting pathway by a predefined threshold time. This function ensures that the drone reaches its destination as soon as possible. The objective represented by this function clearly contradicts the objective represented by f_1.

- f_6 combines all the objectives into one objective using objective weighting method (2).

$$f_6 = 1/(f_1 + f_2 + f_3 + f_4 + f_5) \quad (2)$$

5 Results and Discussion

The algorithm is implemented using MPI (for parallelization) and galib (for genetic algorithm library). Each slave runs a GA. The local GA determines the fittest individual of a local population evaluated with a single objective. Objective1 ensure that the UAV captures as much information as possible, whereas Objective 2 ensures that the drone travels at a speed responsive to the objective 1 (at slave 1). Objective3 optimizes the length of the path taken by the drone; meanwhile Objective 4 ensures that the path is collision free. Objective 5 ensures that the drone reaches its destination at a time smaller than a predefined threshold time.

At each slave, the local GA's maximum number of generation is set to 200 while its population size is set to 1000. The crossover rate is 0.60 and the mutation rate is 0.20. The replacement rate is 0.25. Fig. 3 shows the change in fitness of the fitter individual at each generation. The result shows a constant fitness value obtained from the GA optimizing Objective 4. This can be explained by the big population size (1000) used to optimize a pathway in a scene with a low complexity. Because there are very few obstacles in the scene, it is possible to determine a collision free pathway from the early stage of the evolution (generation 1).

The results obtained by the GA optimizing Objective 1 (Fig. 4) show sudden, yet important, rises in fitness at generations 50, 74 and 82. Those changes can be justified by sound intensity factor that is inversely proportional to the square value of the distance between the drone and the source. The factor imposes a huge difference in the intensity even if distance varies by a few units; a slight change in the distance might have a great influence in the result.

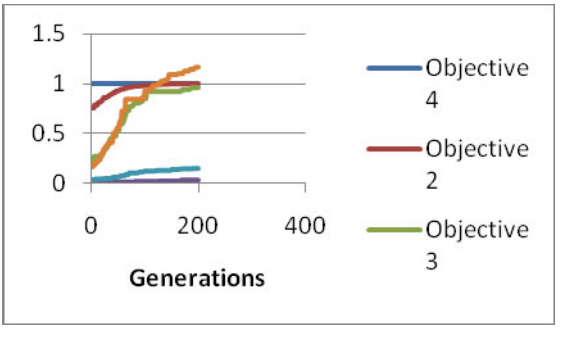


Fig. 3. Fitness Change in MSPVEGA.

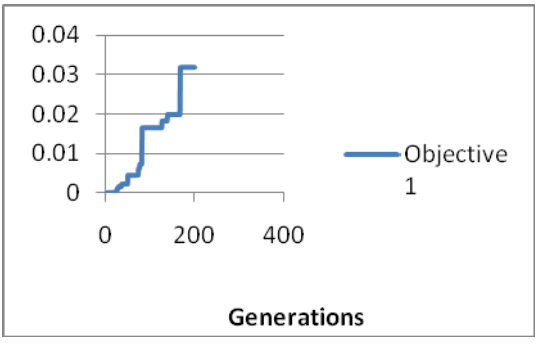


Fig. 4. Fitness change with Objective 1.

A progressive improvement in fitness is observed in the GA optimizing the pathway with Objective 2 (Fig.5). The change, however, becomes less important at the later stages of the evolution. After the 150th generation, the change is almost inexistent. This can be justified by the small range of speed (from 0 to 5) provided to the drone at each point of the pathway. From the results, it is safe to predict that feeding the local GA with a big number of generations (more than 200) would be a waste of computational power.

The GA responsible for finding the shortest path shows a progressive change in fitness in the early stages of the evolution (Fig. 6). After the 100th generation, the fitness improvement becomes less important. From the 100th generation to the 150th generation, there is practically no improvement in fitness. At the later stages of the evolution (after the 150th generation), however, a slight improvement in fitness is observed at some generations. The latter changes, compared to those observed in the early stages, are less important.

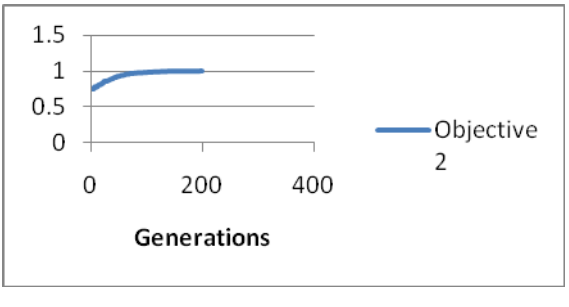


Fig. 5. Fitness change with Objective 2.

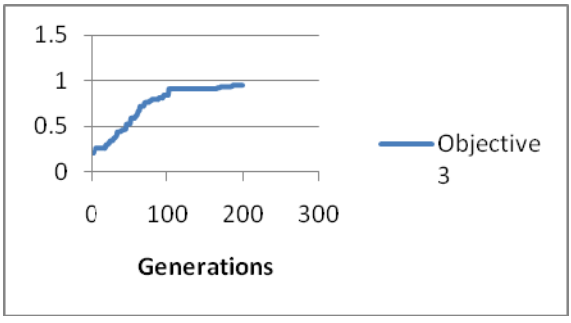


Fig. 6. Fitness change with Objective 3.

There is a constant growth of fitness in the GAs optimizing Objective 5 and Objective 6 (Fig. 7). Those changes are noticeable throughout the whole evolution. The results show the need for exhaustive operations to optimize a multi-objective problem (represented by objective 6) and a time-constrained problem (represented by objective 5), with genetic algorithm.

In addition to the above quantitative analysis, a qualitative analysis is conducted. Qualitative analysis is based on the observation of the phenotype of the optimal results obtained from the local GAs. Fig. 8 shows the difference of two phenotypes issued from different GAs. The longest path is the pathway defined by Objective 4. Although this path represents an individual of perfect fitness (1.0), its length makes it unfit for other objectives. With this GA, any individual whose phenotype is collision free has a perfect fitness. The shortest pathway (red) is the representation of the genotype obtained from the GA optimizing the Objective 3. There is a clear difference in length observed on both pathways even they are issued from the GAs with the same number of generations and population size.

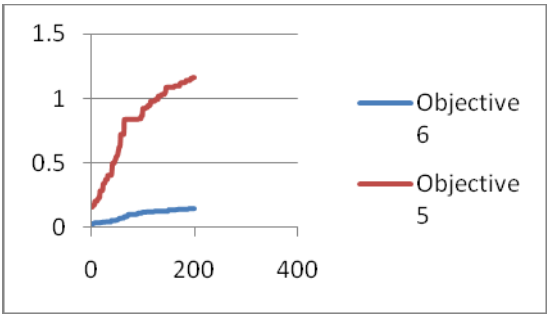


Fig. 7. Fitness change with Objectives 5 and 6

The pathway obtained from the solution of the GA optimizing the combined objective is shown in Fig. 8 while the pathway which represents the time constrained-objective is shown in Fig. 10. The pathway in Fig. 9 is straightened while the pathway in Fig. 8 shows a lot of curves along the line. In a time-constrained optimization problem, the result shown in Fig. 9 is preferable. This result shows the superiority of MSPVEGA over MOGA in multi-objective optimization problems with contrasting objectives.

6 Conclusion and Future Work

MSPVEGA is introduced to solve the multi-objective problems with contrasting objectives. MSPVEGA combines the features of MSGA and PVEPSO. The migrations that occur during the optimization process add variety to the local population. Thus, the algorithm is less likely to be affected by premature convergence. Through the crossover process, the offspring's obtained after the migrations may share features of individuals residing in other populations. The importance of migration is shown in the phenotype pictured Fig. 9. The pathway resulting from the GA optimizing Objective 5 is collision free, short and responds the time constraint requirement. The collision free and short distance are features inherited from the other GAs. The quantitative analysis shows that the performances of local GAs are more influenced by specific genetic parameters. For instance, the GA optimizing Objective 4 requires less number of generations if the population size is big. The required population size depends on the complexity of the scene. There are a number of generations after which the results of GAs optimizing Objective 2 and Objective 3 show fewer improvements. The GAs optimizing Objective 5 and Objective 6 require a big number of generations.

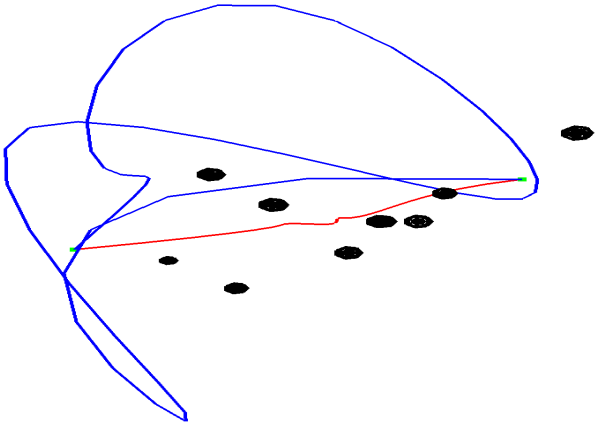


Fig. 8. Pathways defined by Objectives 2 and 3.

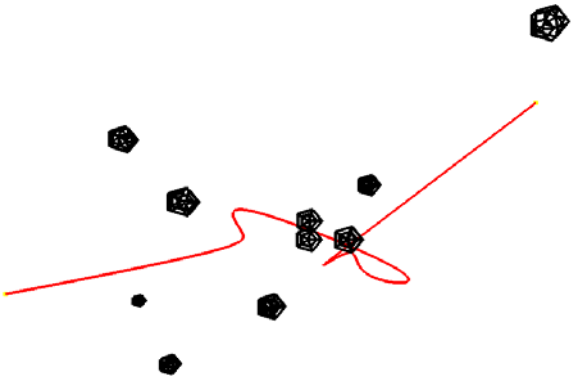


Fig. 9. Phenotype of the result obtained from MOGA.

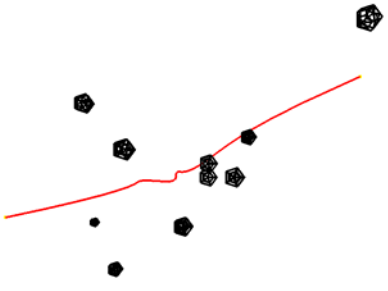


Fig. 10. Phenotype of the result obtained from the GA optimizing Objective 5.

Our future works include implementing multi-objective problem path planning algorithm using different heuristics (neural networks and potential field). A comparative analysis based on the above heuristics will allow us to conclude that MSPVEGA outperforms MOGA, neural networks and potential field in solving the multi-objective path planning problem. Finally, the system will be deployed in real environment to test its performance.

References

[1] Manning, S.D., Rash, C.E., Leduc, P.A., Noback, R.K., McKeon, J.: The Role of Human Causal Factor in U.S. Army Unmanned Aerial Vehicle Accidents. USAARL Report No. 2004-11 (March 2004)

[2] Fahimi, F.: Autonomous Robots - Modeling, Path Planning, and Control (2008)

[3] Barraquand, J., Langlois, B., Latombe, J.-C.: Numerical Potential Field Techniques for Robot Path Planning. IEEE Transactions on Systems, Man, and Cybernetics 22(2) (March/April 1992)

[4] Kding, F.-G., Jiao, P.: AUV Local Path Planning Based on Virtual Potential Field. In: Proceedings of the IEEE International Conference on Mechatronics & Automation (July 2005)
[5] Leigh, R., Louis, S.J., Miles, C.: Using a Genetic Algorithm to Explore A*-like Pathfinding Algorithms. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games (CIG 2007) (2007)
[6] Graham, R.: Real-time Agent Navigation with Neural Networks for Computer Games. M.Sc Thesis (2006)
[7] Burchardt, H., Salomon, R.: Implementation of Path Planning using Genetic Algorithms on Mobile Robots. In: 2006 IEEE Congress on Evolutionary Computation, July 16-21 (2006)
[8] Mittal, S., Deb, K.: Three-Dimensional Offline Path Planning for UAVs Using Multiobjective Evolutionary Algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2007) (2007)
[9] Parsopoulos, K.E., Tasoulis, D.K., Vrahatis, M.N.: Multiobjective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization
[10] van der Berg, J.P., Overmars, M.H.: Roadmap-based motion planning in dynamic environments. IEEE Transaction on Robotics 21(5), 885–897 (2005)
[11] Lingebach, F.: Path Planning using Probabilistic Cell Decomposition. Licentiate Thesis Stockholm, Sweden (2005)
[12] Haddal, C.C., Gertler, J.: Homeland Security: Unmanned Aerial Vehicles and Border Surveillance (July 8, 2010)
[13] Ismail, M.A.: Parallel Genetic Algorithms (PGAs)-Master Slave Paradigm Approach using MPI. IEEE (2004) 0-7803-8655-8/04
[14] Beckhaus, S., Ritter, F.: Cubicalpath - Dynamic Potential Fields for Guided Exploration in Virtual Environments (2001)

Modified Mutation Operators for Differential Evolution

Pravesh Kumar[1], Millie Pant[1], and V.P. Singh[2]

[1] Indian Institute of Technology, Roorkee, India
{praveshtomariitr,millidma}@gmail.com
[2] Millennium Institute of Technology, Saharanpur, India
singhvp3@gmail.com

Abstract. Differential Evolution (DE) has emerged as a popular tool for solving global optimization problems occurring in various fields. However, like most of the population based random search techniques in their basic form, DE has some inherent drawbacks like slow and/or premature convergence, stagnation etc. which sometimes hinders its performance. In the present study we propose two new mutation schemes for DE to enhance its performance in terms of solution quality as well as convergence rate.

Keywords: Optimization, Differential evolution, Mutation.

1 Introduction

Differential evolution algorithm is a simple and powerful evolutionary algorithm which is used to solve global optimization problems over continuous space. It was first proposed by Storn and Price in 1995 [1]. According to frequently reported experimental studies, DE has shown better performance than many other evolutionary algorithms (EAs) in terms of convergence speed and robustness over several benchmark functions and real-world problems [2] It has many attractive characteristics, such as compact structure, ease to use, speedness and robustness [3]. DE is capable of handling non-differentiable, nonlinear and multi-modal objective functions and has been successfully demonstrated to a wide range of real life problems of science and engineering field such that engineering design, chemical engineering, mechanical engineering pattern recognition, and so on [3].

In order to improve the performance of DE, several versions of DE variants have been proposed by many researchers over the last few decades. Some of the modified variants are; Learning enhance DE (LeDE) [3], DE with Trigonometric Mutation (TDE) [4], DE with simplex crossover local search (DEahcSPX) [5], Cauchy mutation DE (CDE) [6], Mixed mutation strategy based DE [7] Fuzzy adaptive DE (FADE) [8], DE with self adaptive control parameter (jDE) [9], Opposition based DE(ODE) [10], Self adaptive DE (SaDE) [11], adaptive DE with optional external archive (JADE) [12], Modified DE (MDE) [13], DE with

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 579–588.
springerlink.com

random localization (DERL)[14], DE with global and local neighborhood (DEGL) [15] and so on.

A recently literature survey of DE variants is given in from; [15]-[17].

In basic DE, the base vector is either randomly selected (DE/rand/bin) or is selected 'greedily'. In this paper we have proposed two new mutation schemes for DE. The first scheme is Inverse Quadratic Interpolation (IQI) scheme and the second one is named Sequential Parabolic Interpolation (SPI). The corresponding DE variants are named IQI-DE and SPI-DE. Both schemes aim at efficiently generating the base vector in the mutation phase of DE. The only difference to DE and both proposed algorithms at base vector in mutation operation. The significance of selecting efficient base vector is discussed later in the paper.

The rest of the paper is structured as follows; In section 2 we give the introduction of basic DE. The description of proposed modified DE variants named IQI-DE and SPI-DE are given in section 3. Experimental settings and numerical results are discussed in section 4 and finally paper is concluded in section 5.

2 Basic Differential Evolution (DE)

DE is a stochastic, robust and direct search based global optimization algorithm. Like other evolutionary algorithms, DE is also start with a set of solutions which is termed as 'population'. Let $P=\{X_i^G,\ i=1,2,...NP\}$ be the population at any generation G which contain NP individuals and an individual can be defined as a D dimensional vector such as $X_i^G=(x_{1,i}^G, x_{2,i}^G..., x_{D,i}^G)$. For basic DE (DE/rand/1/bin) [1] mutation, crossover and selection operations are defined as below;

i. **Mutation:** For each target vector X_i^G select three different vector say X_{r1}^G, X_{r2}^G and X_{r3}^G from the population P such that $r1 \neq r2 \neq r3 \neq i$ then the mutant vector $V_i^G=(v_{1,i}^G, v_{2,i}^G..., v_{D,i}^G)$is defined as:

$$V_i^G = X_{r1}^G + F(X_{r2}^G - X_{r3}^G) \tag{1}$$

Here, X_{r1}^G is base vector, F is a real and constant factor having value between [0, 2] and controls the amplification of differential variation $(X_{r2}^G - X_{r3}^G)$.

ii. **Crossover:** Perform crossover operation to create a trial vector $U_i^G=(u_{1,i}^G, u_{2,i}^G..., u_{D,i}^G)$ as

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, if\ Cr < rand(01) \forall j = jrand \\ x_{j,i}^G \quad otherwise \end{cases} \tag{2}$$

rand (0, 1) is uniform random number between 0 and 1; *Cr* is the crossover constant takes values in the range [0, 1] and *jrand* $\in$ 1, 2,..., *D;* is the randomly chosen index.

iii. **Selection:** During the selection operation we generate a new population $Q = \{X_i^{G+1}, i=1,2,...NP\}$ for next generation $G+1$ by choosing the best vector between trial vector and target vector.

$$X_i^{G+1} = \begin{cases} U_i^G, if\ f(U_i^G) < f(X_i^G) \\ X_i^G \quad otherwise \end{cases} \tag{3}$$

3 Proposed Algorithms

Before discussing the proposed algorithms, let's have a look at the common mutation strategies of DE:

- DE/rand/1
- DE/rand/2
- DE/best/1
- DE/best/2
- DE/rand-to-best/1

In all these strategies we see that the base vector is either selected randomly or is the best solution vector of the population. This base vector plays an important role in generating the mutant vector. A random base vector provides diversity to the population but may slow down the convergence rate. On the other hand when base vector is selected as the best solution vector of the population, the nature of the search process becomes greedy in nature which makes it faster but may also lead to a premature convergence. In the present study we suggest two novel methods of generating the base vector; Inverse quadratic interpolation and sequential parabolic interpolation. These are defined below;

3.1 Inverse Quadratic Interpolation (IQI) Based Base Vector

Inverse quadratic interpolation uses three prior points to fit an inverse quadratic function (x as a quadratic function of y) whose value at y=0 is taken as the next estimate of the root x. If the three point pairs are $(a, f(a))$, $(b, f(b))$, $(c, f(c))$ then the interpolation formula is;

$$x = \frac{(y - f(a))(y - f(b))c}{(f(c) - f(a))(f(c) - f(b))} + \frac{(y - f(b))(y - f(c))a}{(f(a) - f(b))(f(a) - f(c))} + \frac{(y - f(c))(y - f(a))b}{(f(b) - f(c))(f(b) - f(a))} \tag{4}$$

Setting $y=0$ gives a result for the next root estimate.

The working of IQI-DE is as follows;

- First select 3 mutually different random vectors X_{r1}, X_{r2}, and X_{r3} from the population and then find the best among these (say X_{tb}).
- Now take a=X_{r1} b=X_{r2} and c=X_{r2}. and find a new vector (say X_Q) using by Equation-4.
- If any of ($f(a)$~$f(b)$), ($f(b)$~$f(c)$) and ($f(c)$~$f(a)$) are zero then we take these values as ($f(a)$+$f(b)$), ($f(b)$+$f(c)$) and ($f(c)$+$f(a)$).
- Now this new vector X_Q will be the best vector from all X_{r1}, X_{r2}, and X_{r3} because it is next root estimation of function using of X_{r1}, X_{r2}, and X_{r3} as the initial solutions.

3.2 Sequential Parabolic Interpolation (SPI) based Base Vector

Like inverse quadratic interpolation SPI uses the three points (a, $f(a)$), (b, $f(b)$), (c, $f(c)$) to find the next estimation solution. The formulation of SPI is given as below;

$$x = a + \frac{1}{2}\frac{(a-b)^2(f(a)-f(c)) - (a-c)^2(f(a)-f(b))}{(a-b)(f(a)-f(c)) - (a-c)(f(a)-f(b))} \tag{5}$$

Now take a=X_{tb} (let it is X_{r1}), b=X_{r2} and c=X_{r2}. and find a new vector (say X_Q) using by Equation-5. All other steps of SPI-DE are same as defined in IQI-DE.

Graphical description of SPI and IQI is given in Figure 1.

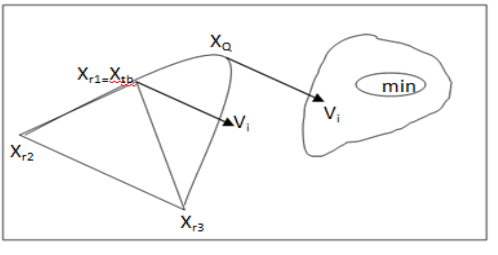


Fig. 1. Differentiate between perturbed vector generated by simple DE and generated by interpolation based DE

3.3 Proposed Variants: IQI-DE and SPI-DE

The computational steps of the proposed variants are same as that of basic DE except in the selection of base vector. Further, in order to provide more diversity to the algorithm and to maintain a balance between exploration and exploitation we fixed a probability (pr) between X_{tb} and X_Q to be selected as a base vector.

Pseudo Code of IQI-DE and SPI-DE Algorithm

```
1   BEGIN
2   Generate uniformly distributed random population P= {X_i^G, i=1,2,...NP}.
    X_i^G = X_lower +(X_upper -X_lower)*rand(0,1), where i =1, 2,..,NP
3   Evaluate f(X_i^G)
4   while (Termination criteria is met )
5   {
6       for i=1:NP
7           {
8            Select three vectors X_r1^G, X_r2^G and X_r3^G which are different from each other and also different from X_i^G
9   Find best vector X_tb^G among these X_r1^G, X_r2^G and X_r3^G
10  Put a= X_r1^G, b= X_r2^G and c= X_r3^G in Equation-4 and Equation-5 and Find X_Q^G
11    if (rand (0,1)< pr )
          X_r1^G= X_Q^G /* for IQI using Equation-4
                   and for SPI using Equation-5*/
12    Else
13        X_r1^G= X_tb^G
14    end if
15  Perform mutation operation as defined by Equation-1
16  Perform crossover operation as defined by Equation-2
17  Evaluate f(U_i^G)
18  Select fittest vector from X_i^G and U_i^G to the population of next generation by using Equation-6
19           }/* end for loop*/
20  Generate new population Q={X_i^G+1 ,i=1,2,...NP}
21  } /* end while loop*/
22  END
```

4 Experimental Setting and Simulated Results

To evaluate performance of proposed algorithms total 6 standard benchmark function which are taken from [3], [12] are used in the present study. Next in this section algorithms for comparison, performance criteria, experimental settings and numerical results are given.

4.1 Algorithms for Comparison

The algorithms for comparison are listed as below;

- The conventional DE algorithm: DE/rand/bin/1[1]
- DE with random localization (DERL) [14]
- Clustered based learning enhance DE (LeDE) [3]
- State of the art DE: jDE [9] SaDE [11], and JADE [12]

4.2 Performance Criteria

We have used 4 different criteria to evaluate performance and comparisons our proposed algorithms which are given below;

Error [10]: The error of a solution X is defined as $f(X)$-$f(X^*)$, where X^* is the global optimum of the function. The minimum error is recorded when the Max NFEs is reached in 50 runs. Also the average and standard deviation of the error values are calculated.

NFEs [10]: The number of fitness function evaluations (NFEs) is recorded when the VTR is reached before to reach maximum NFE. i.e we set the termination criteria as $|f_{opt} - f_{globqal}| \leq VTR$ and record average NFE of successful run over 50 runs.

Convergence graphs [7]: The convergence graphs show the mean fitness performance of the total runs, in the respective experiments.

Acceleration rate (AR) [7]: This criterion is used to compare the convergence speeds between algorithms A and B. It is defined as follows:

$$AR = \frac{NFE_{\mathrm{A}} - NFE_{\mathrm{B}}}{NFE_A}\%$$

4.3 Experimental Setting

All algorithms are implemented in Dev-C++ and the experiments are conducted on a computer with 2.00 GHz Intel (R) core (TM) 2 duo CPU and 2- GB of RAM. All results are calculated the average results over 50 independent runs for every algorithm.

The parameters used are set as follows [3], [12]

Table 1. Parameter Setting

Pop size (*NP*)	100
Dimension (*D*)	30
Scale Factor (*F*), Crossover rate (*Cr*),and Probability(*Pr*)	0.5,0.9, 0.1
Value to reach (*VTR*)	10^{-08} except for Noise function *of VTR* 10^{-02}
Max NFE	300000

4.4 Simulated Result and Comparisons

In Table-2 numerical results are given in term of NFE while in Table-3 results are given in term of average error. Table-2 shows the comparison of proposed IQI-DE and SPI-DE with SDE, DERL and LeDE in term of average NFEs over 50 runs. For LeDE the results have been taken from [3]. In Table-3 comparison with jDE, SaDE and JADE algorithms in term of average error over 50 runs is given. All results for these algorithms are taken from [12].

It is clear from Table-2 for both SPI-DE and IQI-DE the NFEs are less than other DE variants for every benchmark problem except Rosenbrock problem. In the case of Rosenbrock problem DERL obtained the accuracy in 255240 NFEs while SPI-DE takes 263740 NFEs and IQI-DE takes 258000 NFEs to achieve the accuracy. So there is a little difference to between the NFEs of DERL and proposed SPI-DE and IQI-DE. Otherwise both proposed algorithms give the better performance in the case of other benchmark problems. The total number of NFEs for all benchmark problems is 1055250, 585570, 569922, 540430 and 457620 by SDE, DERL, LeDE, SPI-DE and IQI-DE respectively. Hence the acceleration rate of SPI-DE over SDE is 48.78%, over DERL is 7.7% and over LeDE is 5.17% while acceleration rate of IQI is 56.63%, 21.85%, and 19.70% with respect to SDE, DERL and LeDE. Hence IQI-DE gives the better performance all algorithm and also from SPI-DE. IQI-DE is 15.32 % faster than SPI-DE.

In the Table-3 comparison is taken over some state of the art algorithm like SaDE, jDE and JADE algorithms. Here first we have fixed max-iteration and then find the average error and standard deviation over the 50 runs. It can be seen easily for all case IQI-DE gives minimum error and standard deviation in compare to all algorithm while SPI-DE gives minimum error for all problem except than Ackley problem when compare to SaDE and jDE while it gives the better performance only in two benchmark problem when it compare with JADE.

Table 2. Experimental results and comparison in term of average NFEs over 50 runs

Fun	**SDE**	**DERL**	**LeDE /Exp**	**SPI-DE**	**IQI-DE**
Sphere	96620	51320	49494	44680	**35970**
Schwefel 2.21	160400	84000	77464	70440	**52130**
Rosenbrock	416640	**255240**	282972	263740	258000
Noise	135290	63370	33302	48700	**25700**
Griewank	101060	52860	50579	45400	**39220**
Ackley	145240	78780	76111	67470	**46600**
Total	1055250	585570	569922	540430	**457620**

Table 3. Comparison with state of the art algorithms in term of average fitness error and standard deviation in 50 runs

Fun	Max-Gen	SaDE	jDE	JADE	SPI-DE	IQI-DE
Sphere	1500	4.5E–20	2.5E–28	1.8E–60	1.1E-034	**3.2E-070**
		(6.9E–20)	(3.5E-28)	(8.4E–60)	(2.2E-034)	**(2.9E-070)**
Schwefel	2000	1.9E–14	1.5E–23	1.8E–25	5.9E-024	**1.1E-048**
2.22		(1.1E–14)	(1.0E–23)	(8.8E–25)	(3.5E-024)	**(1.1E-048)**
Rosenbrock	3000	2.1E+01	1.3E+01	8.0E–02	0.0E+00	**0.0E+00**
		(7.8E+00)	(1.4E+01)	(5.6E–01)	(0.0E+00)	**(0.0E+00)**
Noise	3000	4.8E–03	3.3E–03	6.4E–04	1.3E-03	**2.7E–04**
		(1.2E–03)	(8.5E–04)	(2.5E–04)	(5.1E-04)	**(1.4E–04)**
Griewank	500	7.8E–04	1.9E–05	9.9E–08	9.6E-009	**8.2E-014**
		(1.2E–03)	(5.8E–05)	(6.0E–07)	(5.6E-009)	**(9.3E-014)**
Ackley	2000	4.3E–14	4.7E–15	4.4E–15	4.2E-013	**1.4E-016**
		(2.6E–14)	(9.6E–16)	(0.0E+00)	(1.9E-013)	**(0.0E+00)**

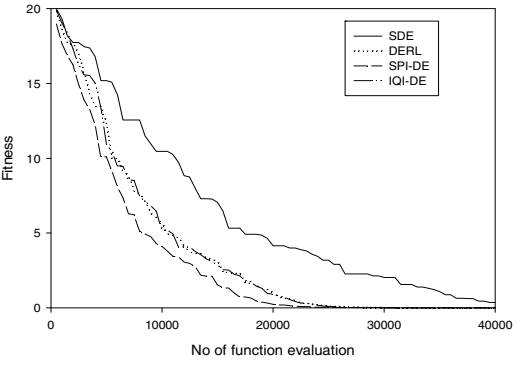


Fig 2(a)

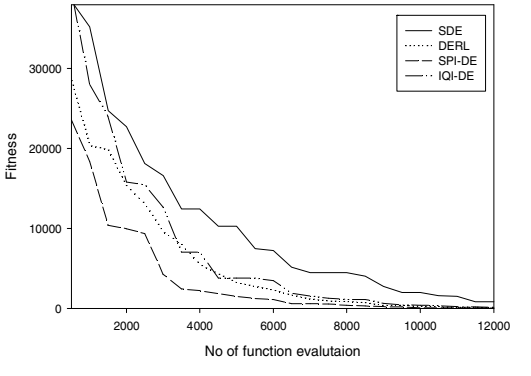


Fig 2(b)

Fig. 2. Convergence graphs of Ackley function and Sphere function

5 Conclusions

In the present study, two modified methods are suggested for selecting the base vector. These methods: IQI and SPI exploit the local information of the search domain for generating the base vector. The corresponding variants are named IQI-DE and SPI-DE. From the numerical results it is observed that the proposed modifications help in improving the performance of DE in terms of convergence rate besides maintaining the solution quality.

References

[1] Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous. Spaces, Berkeley, CA, Tech. Rep. TR-95-012 (1995)

[2] Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. In: Congress on Evolutionary Computation, pp. 980–987 (2004)

[3] Cai, Y., Wang, J., Yin, J.: Learning enhanced differential evolution for numerical optimization. Soft Computing (2011), doi:10.1007/s00500-011-0744-x

[4] Fan, H., Lampinen, J.: A trigonometric mutation operation to differentia evolution. Journal of Global Optimization 27, 105–129 (2003)

[5] Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local Search. IEEE Transaction of Evolutionary Computing 12(1), 107–125 (2008)

[6] Ali, M., Pant, M.: Improving the performance of differential evolution algorithm using cauchy mutation. Soft Computing (2010), doi:10.1007/s00500-010-0655-2

[7] Pant, M., Ali, M., Abraham, A.: Mixed mutation strategy embedded differential evolution. In: IEEE Congress on Evolutionary Computation, pp. 1240–1246 (2009)

[8] Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. Soft Computing Fusion Found Methodol. Appl. 9(6), 448–462 (2005)

[9] Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Transaction of Evolutionary Computing 10(6), 646–657 (2006)

[10] Rahnamayan, S., Tizhoosh, H., Salama, M.: Opposition based differential evolution. IEEE Transaction of Evolutinary Computing 12(1), 64–79 (2008)

[11] Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transaction of Evolutionary Computing 13(2), 398–417 (2009)

[12] Zhang, J., Sanderson, A.: JADE: adaptive differential evolution with optional external archive. IEEE Transaction of Evolutionary Computing 13(5), 945–958 (2009)

[13] Babu, B.V., Angira, R.: Modified differential evolution (MDE) for optimization of non-linear chemical processes. Computer and Chemical Engineering 30, 989–1002 (2006)

[14] Kaelo, P., Ali, M.M.: A numerical study of some modified differential evolution algorithms. European Journal of Operational Research 169, 1176–1184 (2006)
[15] Das, S., Abraham, A., Chakraborty, U., Konar, A.: Differential evolution using a neighborhood based mutation operator. IEEE Transaction of Evolutionary Computing 13(3), 526–553 (2009)
[16] Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell. Rev. 33(1-2), 61–106 (2010)
[17] Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Transaction of Evolutionary Computing 15(1), 4–13 (2011)

Genetic Algorithm in Data Capturing and Mining

M.K. Thanuja and C. Mala

Department of Computer Science and Engineering, National Institute of Technology, Thiruchrapalli, India
thanujaprakash@yahoo.in, mala@nitt.edu

Abstract. In the current business scenario, dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases .For each organization a process of centralized data management and an efficient data retrieval method has become essential. Centralization of data is needed to maximize user access for knowledge capturing analysis, business improvement and for future use. Dramatic technological advances are making this vision a reality for many companies. This paper presents a design of a tool for extracting the data from a centralized vault, indexing, retrieving domain-specific data and optimizing the result using genetic algorithm.

Keywords: ***Cosine*** similarity, Fitness function, Genetic Algorithm, Information Retrieval, domain-specific.

1 Introduction

With large amount of information resources available through various resources in organizations, the information overload for the user has become overwhelming. There has been dramatic increase in thirst for finding the best and the newest information through the world. Nowadays the thirst is more on the domain-specific information, as most of current information systems are, the user looking for some topic and the Information Retrieval System(IRS) retrieves too much information. The systems provide no qualitative distinction between the relevant and irrelevant documents [1].Genetic algorithms (GAs) are not new to information retrieval. Gordon suggested representing a posting as a chromosome and using genetic algorithms to select good indexes [4][10]. Yang *et al*. suggested using GAs with user feedback to choose weights for search terms in a query [5][10].Morgan and Kilgour suggested an intermediary between the user and IR system employing GAs to choose search terms from a thesaurus and dictionary

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 589–599.
springerlink.com

[6][10]. Boughanem et al. [7][10] examine GAs for information retrieval and they suggested new crossover and mutation operators.

2 Motivation

BHEL is an engineering and manufacturing organization in which a large volume of documents are captured into a centralized vault through different applications. Due to lack of domain-specific information system, the past and the present knowledge in the documents are not readily available for the users. This paper aims at the design of a tool for extraction of the data from the documents in the vault, indexing, retrieval of domain-specific information and optimization of the result.

3 Proposed System Architecture

3.1 Text Extraction

The various methods of information retrieval that have been proposed in the literature are full text scanning, indexing, signature files and clustering [3].Among them each method has its own advantages and disadvantages. Indexing is the method considered in this design. In any Information Retrieval System (IRS), the keywords are extracted from the documents by parsing the documents. The parsing of a document involves the steps namely markup and format removal of tags, tokenization, filtration. The above steps help in extracting distinct and meaningful keywords from the documents. The type of documents considered for scanning in the proposed tool are text files, PDF files, MS Word files, MS Excel files, MS PowerPoint files, HTML files and XML files. An algorithm is developed to find the type of the document and parse accordingly. The output of the algorithm is the distinct keywords and their relevant attributes.

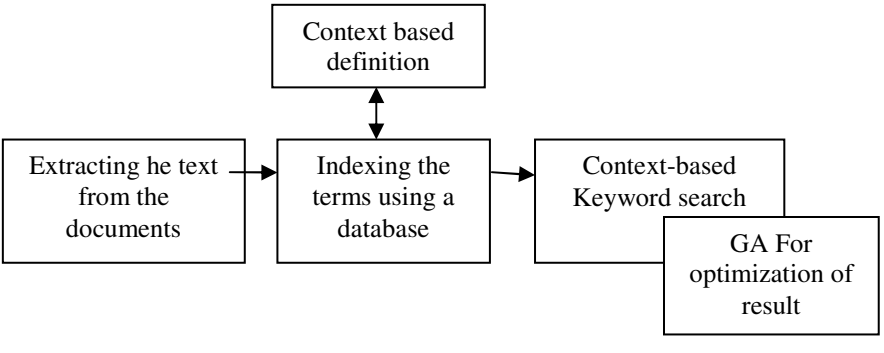


Fig. 1. Architecture of the proposed system

3.2 Indexing

Various indexing techniques are available in the field of IR, but the common and more efficient indexing technique found in the literature is inverted index. An inverted index is an index data structure storing a mapping from content, such as words, to its location in a document or a set of documents. It is proposed to use inverted index technique for the proposed system. It is the most popular data structure used in document retrieval systems [3].A general inverted file index consists of two parts: a set of inverted file entries, being lists of identifiers of the records containing each indexed word, and a search structure for identifying the location of the inverted file entry for each term. The search structure may contain the information like the document id, in which the term is located. In this system the inverted index is implemented as a database. And other attributes used in the inverted list are document number, the weight of the terms in the document and the position of the keyword in the document, the context of the document & the type of the document. In any IRS, the system first extracts keywords from documents and then assigns weights to the keywords by using different approaches. This research presents the following weighting scheme for calculating the weight of the terms[3].

$$W(t_i,d_j)=f(t_i,d_j)/\text{sqrt}(|d_j|) \text{ x } \ln(D/f(t_i)) \quad (1)$$

where $f(t_i,d_j)$ is the no. of times term t_i appears in document d_j, $|d_j|$ is the no. of terms in d_j, $f(t_i)$ is the no. of documents. containing t_i and D is the no. of docs.in the collection.

3.3 Context Based Definition

The context of the document is captured as one of the attributes of the inverted list. The context here is considered as the name of the directory in which the document is located.

3.4 Document Retrieval and Relevancy

In this paper it is proposed to use the vector model of the IRS. In this model a document is viewed as a vector in n-dimensional space (where n is the number of distinguishing keywords used to describe contents of the documents in a collection) and each term represents one dimension in the document space[10]. A query is also treated in the same way and constructed from the terms and weights provided in the user request. Document retrieval is based on the measurement of the similarity between the query and the documents. In this method, the retrieved documents can be orderly presented to the user with respect to their relevance to the query [9].

It is proposed to use Cosine similarity function in this work

$$Sim(Q, D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}} \quad (2)$$

3.5 Optimization Using Genetic Algorithm

In GA, the search space is contains solutions to the problem represented by a string is known as a chromosome. Each chromosome has an objective function value, called fitness. A population is set of chromosomes together with their associated fitness function. This population, at a given iteration of the GA, is called a generation.

3.5.1 The Genetic Approach

Once significant keywords are extracted from test data (relevant and irrelevant documents) weights, frequency are assigned to the keywords and , domain and type of the document are included the inverted list. The binary weights of the keywords are formed as a query vector. The GA has been applied for a fitness function to get an optimal or near optimal query vector.

3.5.1.1 Representation of the chromosomes

These chromosomes use a binary representation. The representation is based on the relevancy of the documents with the given query. The no. of genes will be equal to the no. of documents in the collections. In this genetic chromosomes are based on relevancy of the documents w.r.t to the query is given by the cosine similarity function mentioned in section 3.4 and is either 1 or 0. C_1={10 1 1 1 0 1 0 1 10 1 0 0 0 0 0 0 0 0} means the documents 1,3,4,5,7,9, 10&12 are relevant to the query for a collection of 20 documents. The no. of genes in a chromosome is the total no. of documents in the collection.

3.5.1.2 Selection

As the selection mechanism, the best chromosomes will on average achieve more copies, and the worst fewer copies. The algorithm is topped as the fitness value of all the chromosomes of a generation is equal and are equal to the maximum fitness of the previous generations[10].

3.5.1.3 Fitness Function

Fitness function is a performance measure or reward function, which evaluates how each solution, is good. As in the literature, in any GA used for optimization of text retrieval, the search time, the type of the document and the context specific documents are not considered ,it is proposed use these parameters in this work. The fitness function coined the GA with following fitness function is used

$$F=rel(d) + 1/T_d + S + T_s + d_{typ} + d_{dom} \quad (3)$$

where rel(d) is the total no. of relevant documents retrieved with respect to the given queryi.e $\sum_{k=1}^{n} x_k$ where x_k is k^{th} gene of the i^{th} chromosome,$T_s$is the time taken to search a relevant documents,T_d is the time taken to download the relevant documents,S total size of the relevant documents, $d_{typ}$is the type of the document and d_{dom} is the context of the document. The simulation results show that there are performance improvements in the optimisation of the results in inclusion of the proposed parameters. With increase in the search time the no. of relevant documents retrieved is expected to be more. With a fixed search time the no. of types of documents are expected to be more. Similarly with a fixed search time the no. of contexts of documents to be retrieved is expected to be more. Hence T_s,d_{typ} & d_{dom} are added to the fitness functions as a parameter of directly proportionate.

4 Performance Analysis

The GA with proposed FF is tested the documents in a centralized vault by including the parameters one by one

4.1 Search Time

At this step, the fitness function proposed is

$$F=rel(d) + 1/T_d + S + T_s \quad (4)$$

The fitness function without T_s but with the time taken to select the relevant document (T_r) by the user is already considered in the literature [10] and here the FF with T_s is proposed. The search time (T_s) is the total time taken to make the relevant documents available to the users considering the type of the document and based on the context. Initial generation and subsequent generation of a sample of 6 chromosomes are shown in Table 1. In the cross over process the chromosomes with maximum fitness are selected. The other chromosomes are crossed over with a probability of 0.8. The mutation probability is 0.0002. For the Chromosomes and the fitness for initial generation and subsequent generation [10] may be referred.

The information retrieval efficiency measures from precision.

$$Precision = \frac{\text{Number of documents retrieved and relevant}}{\text{Total Retrieved}} \quad (5)$$

The experiment results are shown for 200 queries and the experimental observation, the best values for this test documents collections at crossover probability Pc = 0.7 and mutation rate is Pm= 0.0002for the GA follows

Table 1. Queries and the precision of FF proposed in Eqn.(4)

Queries	Average Precision for 200 queries						
	Precision			Queries	Precision		
	Without GA	GA without T_s in FF	GA with T_s in FF		Without GA	GA without T_s in FF	GA with T_s in FF
1-10	0.58	0.6	0.5	101-110	0.62	0.63	0.65
11-20	0.52	0.55	0.5	111-120	0.57	0.57	0.6
21-30	0.6	0.625	0.664	121-130	0.54	0.58	0.6
31-40	0.7	0.734	0.8	131-140	0.62	0.67	0.69
41-50	0.48	0.5	0.5	141-150	0.66	0.69	0.69
51-60	0.5	0.51	0.61	151-160	0.71	0.7	0.71
61-70	0.57	0.6	0.61	161-170	0.71	0.71	0.71
71-80	0.61	0.64	0.59	171-180	0.65	0.68	0.7
81-90	0.52	0.55	0.55	181-190	0.62	0.65	0.7
91-100	0.56	0.58	0.6	191-200	0.65	0.691	0.691

The comparison of performance of the search algorithm optimization without GA, with GA with and without the search time as parameter in the FF and shown in Fig.2.

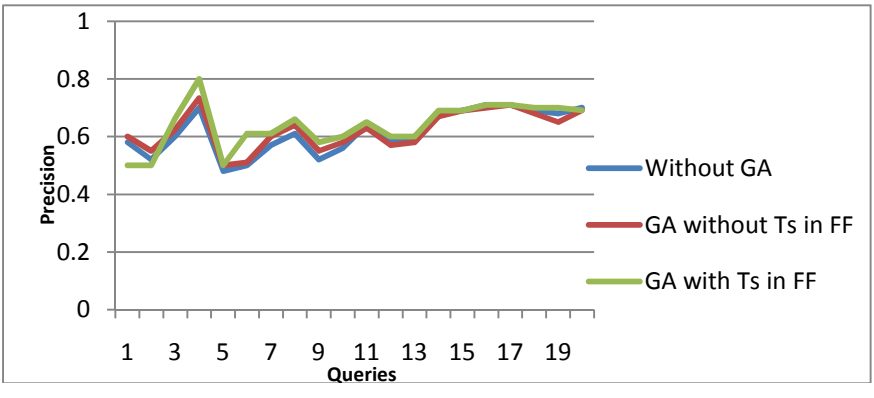


Fig. 2. Comparison of performance without GA, with GA with and without search time

4.2 Document Type

The second parameter included in the proposed FF is document type as follows

$$F=rel(d) + 1/T_d + S + T_s + d_{Typ} \quad (6)$$

The fitness function without Document type is proposed in section 4.1 and here d_{typ} is included. Different weightages are assigned to different types of documents as more weightage is given to the editable type of documents and least weightage is given to the non-editable type of documents. The results are shown in the

Table 2.In the cross over process the chromosomes with maximum fitness are selected. The other chromosomes are crossed over with a probability of 0.8. The mutation probability is 0.0002.The experiment results are shown for 200 queries

Table 2. Queries and the precision of FF proposed in Eqn.(6)

Average Recall Precision for 200 queries							
Queries	Precision			Queries	Precision		
	Without GA	GA without d_{typ} in FF	GA withd_{typ} in FF		Without GA	GA without d_{typ} in FF	GA withd_{typ} in FF
1-10	0.63	0.7	0.65	101-110	0.59	0.61	0.65
11-20	0.56	0.57	0.6	111-120	0.69	0.7	0.71
21-30	0.6	0.625	0.664	121-130	0.52	0.52	0.55
31-40	0.59	0.61	0.7	131-140	0.6	0.61	0.65
41-50	0.6	0.6	0.6	141-150	0.67	0.69	0.7
51-60	0.6	0.61	0.65	151-160	0.67	0.67	0.7
61-70	0.59	0.62	0.61	161-170	0.71	0.72	0.73
71-80	0.61	0.63	0.6	171-180	0.68	0.7	0.71
81-90	0.59	0.6	0.65	181-190	0.67	0.68	0.7
91-100	0.58	0.6	0.7	191-200	0.67	0.69	0.69

The performance of the search algorithm optimization is compared without GA,with GA with and without the document type and shown in Fig.3.

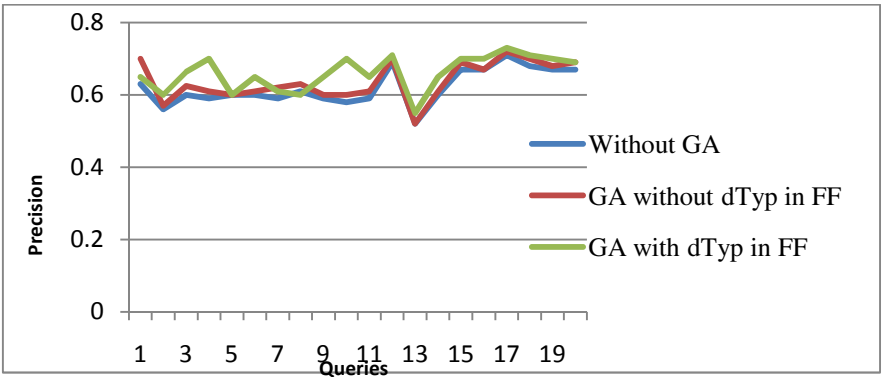


Fig. 3. Comparison of performance without GA, with GA with and without document type

From a test data of 20 documents and for 200 queries the various types of documents retrieved are shown in Table 4 and Fig 4.

Table 3. Retrieval of relevant documents based on type without and with GA

Type of documents	DOC	EXCEL	TEXT	PDF
Average no. of relevant documents without GA	10	2	2	10
Average No. of relevant documents with GA	25	9	8	30
Average total no. of relevant documents	30	10	10	60

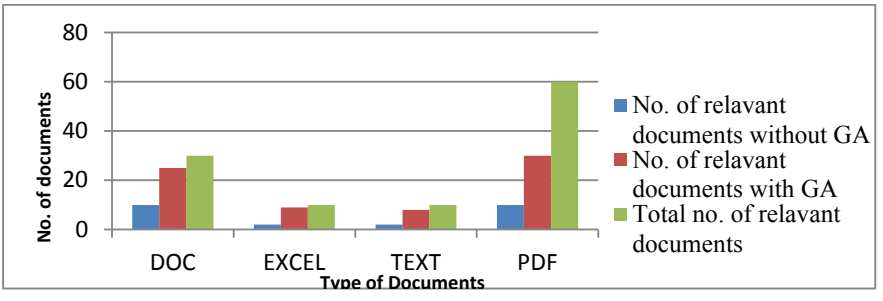


Fig. 4. Comparison of retrieved document based on type without and with GA

4.3 Domain Specific Documents

The final fitness function proposed is

$$F=rel(d) + 1/T_d + S + T_s + d_{Typ} + d_{dom} \quad (7)$$

The fitness function without domain-specific documents is considered in section 4.2 and here d_{dom} is included. The domain of the document relevancy is obtained

Table 4. Queries and the precision of FF proposed in Eqn.(7)

Average Recall Precision for 200 queries							
Queries	Precision			Queries	Precision		
	Without GA	GA without d_{typ} in FF	GA withd_{typ} in FF		Without GA	GA without d_{typ} in FF	GA withd_{typ} in FF
1-10	0.3	0.3	0.4	101-110	0.4	0.42	0.46
11-20	0.3	0.37	0.4	111-120	0.41	0.43	0.41
21-30	0.34	0.36	0.46	121-130	0.39	0.4	0.41
31-40	0.4	0.41	0.5	131-140	0.37	0.39	0.4
41-50	0.39	0.4	0.41	141-150	0.38	0.4	0.41
51-60	0.37	0.39	0.45	151-160	0.4	0.41	0.456
61-70	0.4	0.4	0.41	161-170	0.39	0.392	0.41
71-80	0.38	0.41	0.42	171-180	0.65	0.7	0.71
81-90	0.4	0.42	0.43	181-190	0.4	0.42	0.44
91-100	0.41	0.46	0.47	191-200	0.45	0.46	0.46

from the user. A weightages of 1 is assigned context relevant documents and 0 to context irrelevant documents. The results are shown in the Table 4. In the cross over process the chromosomes with maximum fitness are selected. The other chromosomes are crossed over with a probability of 0.8.The mutation probability is 0.0002.The experiment is done for 200 queries and the comparative precisions are listed in Table 4.

The performance of the search algorithm optimization is compared without GA, with GA with and without the domain specific documents and shown in Fig.5.

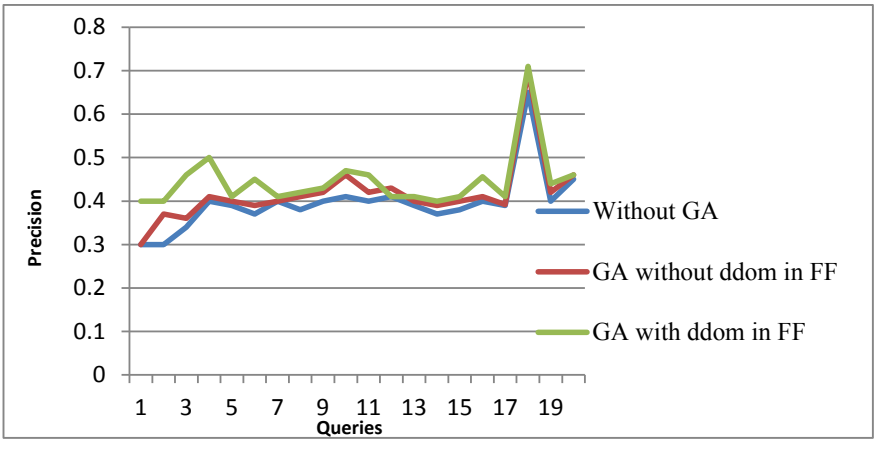


Fig. 5. Comparison of performance without GA, GA with and without document type

From a test data of 20 documents and for 200 queries the various documents retrieved based on different contexts are shown in Table 5 and Fig 6.

Table 5. Retrieval of relevant documents based on context

Type	Context1	Context2	Context3
No. of relevant documents without GA	5	5	6
No. of relevant documents with GA	10	12	10
Total no. of relevant documents	30	20	20

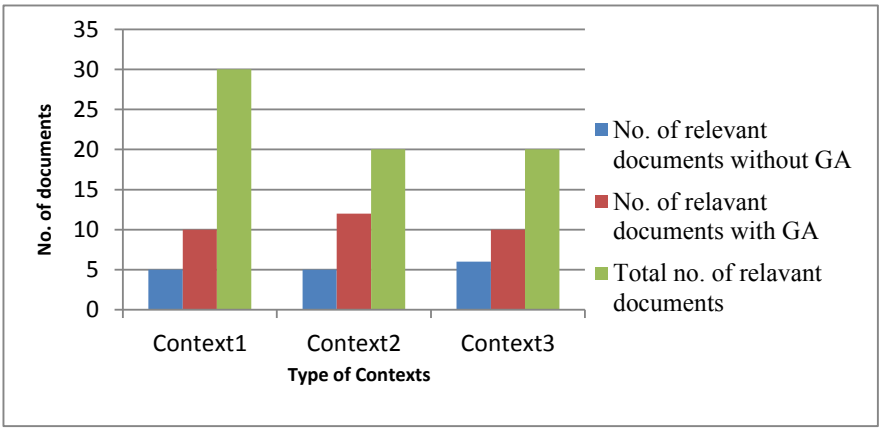


Fig. 6. Comparison of retrieved document based on context without and with GA

5 Conclusion

Comprehensive data warehouses that integrate operational data with customer, supplier, and market information have resulted in an explosion of information. Competition requires timely and sophisticated analysis on an integrated view of the data. However, there is a growing gap between more powerful storage and retrieval systems and the users' ability to effectively analyze and act on the information they contain. The data mining tools can make this leap. Quantifiable business benefits have been proven through the integration of data mining with current information systems, and new products are on the horizon that will bring this integration to an even wider audience of users.

References

[1] Faloutsos, C., Christodoulakis, S.: An access Method for Documents and its Analytical Performance Evaluation, Signature files. ACM Transactions on Information Systems (TOIS) 2(4), 267–288 (1984)

[2] Leake, D.B., Scherle, R.: Towards context-based search engine selection. In: Proceedings of IUI 2001, pp. 109–112 (2001)

[3] Martin-Bautista, M.J., Vila, M.-A., Larsen, H.L.: A Fuzzy Genetic Algorithm Approach to an Adaptive Information Retrieval Agent. Journal of the American Society for Information Science 50(9), 760–771 (1999)

[4] Gordon, M.: Probabilistic and genetic algorithms in document retrieval. Communications of the ACM 31(10), 1208–1218 (1988)

[5] Yang, J., Korfhage, R., Rasmussen, E.: Query improvement in information retrieval using genetic algorithms–a report on the experiments of the TREC project. In: Proceedings of the 1st Text Retrieval Conference (TREC-1), pp. 31–58 (1992)

[6] Morgan, J., Kilgour, A.: Personalising on-line information retrieval support with a genetic algorithm. In: Moscardini, A., Smith, P. (eds.) PolyModel 16: Applications of Artificial Intelligence, pp. 142–149 (1996)
[7] Boughanem, M., Chrisment, C., Tamine, L.: On using genetic algorithms for multimodal relevance optimization in information retrieval. Journal of the American Society for Information Science and Technology 53(11), 934–942 (2002)
[8] Salton, G., McGill, M.H.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)
[9] Pathak, P., Gordon, M., Fan, W.: Effective information retrieval using genetic algorithms based matching functions adaption. In: Proc. 33rd Hawaii International Conference on Science (HICS), Hawaii, USA (2000)
[10] Thanuja, M.K., Mala, C.: A Search Tool Using Genetic Algorithm. In: Das, V.V., Thomas, G., Lumban Gaol, F. (eds.) AIM 2011. CCIS, vol. 147, pp. 138–143. Springer, Heidelberg (2011)

Enhancing Scout Bee Movements in Artificial Bee Colony Algorithm

Tarun Kumar Sharma and Millie Pant

Indian Institute of Technology, Roorkee, India
taruniitr1@gmail.com, millifpt@iitr.ernet.in

Abstract. In the basic Artificial Bee Colony (ABC) algorithm, if the fitness value associated with a food source is not improved for a certain number of specified trials then the corresponding bee becomes a scout to which a random value is assigned for finding the new food source. Basically, it is a mechanism of pulling out the candidate solution which may be entrapped in some local optimizer due to which its value is not improving. In the present study, we propose two new mechanisms for the movements of scout bees. In the first method, the scout bee follows a non-linear interpolated path while in the second one, scout bee follows Gaussian movement. Numerical results and statistical analysis of benchmark problems indicate that the proposed modifications enhance the performance of ABC.

Keywords: Artificial Bee Colony, Quadratic Interpolation, Gaussian distribution.

1 Introduction

Artificial bee colony algorithm (ABC) is a new computational technique developed by Karaboga [1] based on simulating the foraging behavior of honey bee swarm. The performance of ABC is competitive to other population-based algorithms with an advantage of employing fewer control parameters [2–3]. Due to its simplicity and ease of implementation, ABC has captured much attention and has been applied to solve many practical optimization problems [4–6] since its invention in 2005. However, similar to other evolutionary algorithms, ABC also has some drawbacks which hamper its performance. For example, the convergence speed of ABC is typically slower than some other population-based algorithms like differential evolution (DE) [7] and PSO [8] when handling unimodal problems [6]. Also, ABC can easily get trapped in the local optima when solving complex multimodal problems [6]. The search equation of ABC used to generate new candidate solutions based on the information of previous solutions is good at exploration it is poor at exploitation [9], which results in the imbalance between exploration and exploitation. Therefore, accelerating convergence speed and avoiding the local optima have become two important and appealing goals in ABC

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 601–610.
springerlink.com

research. A number of ABC variants have, been proposed to achieve these two goals [9–11].In this paper a variation is made in the movement of scout bees. In the basic ABC algorithm are assigned a random location for determining the new food location. This is generally done with the help of computer generated random numbers following uniform distribution, which may not prove to be very efficient for locating new food sources. In the present study we focus on enhancing the movement of scout bees in order to get more efficient food locations. Two enhancements are proposed in the scout bee phase (1) quadratic interpolation and (2) Gaussian movements. The first variant is named as QABC, while the second variant is named as GABC.

The rest of the paper is organized as follows: section 2 outlines the overview of ABC algorithm. In section 3, the proposed QABC and GABC are described. In section 4, experimental settings, evaluation criteria and results are given. Finally the paper concludes with section 5.

2 Artificial Bee Colony

ABC classifies the foraging artificial bees into three groups, namely, employed bees, onlooker bees and scout bees. Half of the colony consists of employed bees, and the other half includes onlooker bees. In the foraging process of honeybee colonies, initially, some bees search randomly for food in a given area around the hive. After finding a food source, these bees take some nectar back to the hive, deposit the nectar and share the nectar information of the food sources with other bees waiting at the dance area (where waggle dance is performed) within the hive. The bee colony then enters a new cycle of iterations. At each iteration, following steps take place: (1) after sharing the information, an employed bee will either become an onlooker after abandoning the food source or continue to forage its previously visited site; (2) some onlookers in the hive will simultaneously follow some employed bees based on the received information in order to further forage on some specific memorized food sources; and (3) some scouts will spontaneously start a random search. An important stage of the ABC algorithm, from which in fact the collective intelligence arises, is the sharing of information. This is achieved by influencing the behavior of onlookers which will select their food source according to following probability:

$$P_i = f_i / \sum_{k=1}^{SN} f_k \tag{1}$$

where f_i is the fitness value of a i, food source (position in parameter space). In other words onlookers will explore promising locations with higher probability than others. Candidate food sources are generated from memorized ones according to:

$$V_{ij} = x_{ij} + \phi_{ij}(x_{i,j} - x_{k,j}) \tag{2}$$

where $i, k = 1,..., SN$, $j = 1,..., n$, and v_i is the new food source generated by using both, the current food source x_i and a randomly chosen food source x_k from the population and $-1 \le \phi_{ij} \le 1$ (generated at random every time it is used) determines the step size of the movement. Both, *i* and *j* are generated at random but $k \neq i$. When a source does not improve after a certain number of iterations, it is abandoned and replaced by one found by a scout bee, using Eq. 2: which involves the generation of a new solution at random.

$$X_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{3}$$

Where $i = 1,2,..,SN$. $j = 1,2,\ldots,n$. $x_{max,j}$ and $x_{min,j}$ are upper and lower bounds of parameter *j*, respectively. These food sources are randomly assigned to *SN* number of employed bees and their finesses are evaluated.

Basic steps of Artificial Bee Colony:

Initialization of food sources (Population): The initial population of solutions is filled with *SN* number of randomly generated *n*-dimensional real-valued vectors (i.e., food sources). Let $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ represent the i^{th} food source in the population, and then each food source is generated by equation (3).

Employed bee initialization: In this phase each employed bee X_i generates a new food source V_i in the neighborhood of its present position by using solution search equation (2). Once V_i is obtained, it will be evaluated and compared to X_i. If the fitness of V_i is equal to or better than that of X_i, V_i will replace X_i and become a new member of the population; otherwise X_i is retained. In other words, a greedy selection mechanism is employed between the old and candidate solutions.

Probabilistic Selection: An important stage of the ABC algorithm, from which in fact the collective intelligence arises, is the sharing of information. This is achieved by influencing the behavior of onlookers which will select their food source according to probability equation (1)

Onlooker bee phase: An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source X_i depending on its probability value P_i. Once the onlooker has selected her food source X_i, she produces a modification on X_i by using Equation (2). As in the case of the employed bees, if the modified food source has a better or equal nectar amount than X_i, the modified food source will replace X_i and become a new member in the population.

Scout bee phase: If a food source X_i cannot be further improved through a predetermined number of trials limit, the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. The scout produces a food source randomly using equation (3).

3 QABC and GABC: The Proposed Variants

The proposed QABC and GABC differ from the basic ABC in terms of the movements of the scout bees. Instead of assigning a random movement for the scout bees, in the proposed variants the scout bees are moved in order to find a better location for them. For this purpose we have used the method of interpolation and method following Gaussian distribution.

Quadratic Interpolation: In the present study, the three initial solution vectors r_1, r_2, and r_3 are selected randomly between 0 and 1, distinct from each other. From these three points, the coordinates of the new Food Location $V_{i,G+1} = (v_{1,G+1}, v_{2,G+1}, \ldots, v_{n,G+1})$, are determined as:

$$v_{i,g+1} = 0.5 * \frac{(x_{r_2,g}^2 - x_{r_3,g}^2) * f(x_{r_1,g}) + (x_{r_3,g}^2 - x_{r_1,g}^2) * f(x_{r_2,g}) + (x_{r_1,g}^2 - x_{r_2,g}^2) * f(x_{r_3,g})}{(x_{r_2,g} - x_{r_3,g}) * f(x_{r_1,g}) + (x_{r_3,g} - x_{r_1,g}) * f(x_{r_2,g}) + (x_{r_1,g} - x_{r_2,g}) * f(x_{r_3,g})} \tag{4}$$

Gaussian movement: Gaussian distribution also called a "bell shaped curve" as a mode of perturbation (or mutation) has been used for generating new candidate solutions. The PDF for Gaussian function is given as:

$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$; with mean 0 and standard deviation 1, i.e. N (0, 1).

Proposed Variants: After initialization QABC starts like the usual ABC as iscussed in previous section and when the food source X_i cannot be further improved through a fixed trial limit, the food source get abandoned, and the corresponding employed bee act as *scout bee*. Than scout bee produces a new food source using, the quadratic interpolation given in equation (4) for QABC. In GABC the scout bee produces a new food source using, Gaussian random numbers and the equation is given as:

$$x_{i,j} = x_{\min,j} + Gauss(-1,1)(x_{\max,j} - x_{\min,j}) \tag{5}$$

where, *Gauss* is a random number following Gaussian distribution.

Fig. 1 explains the pseudocode of QABC and GABC & flow graph is shown in Fig. 2.

Begin
1. Initialize the population of food sources x_i, $i = 1,\ldots,SN$
2. Evaluate each food source x_i, $i = 1,\ldots,SN$
3. $cycle = 1$

Repeat
For each food source x_i in the population
4. Generate a new food source v_i by its corresponding employed bee (*Eq. 2*)
5. Evaluate v_i
6. Keep the best solution between x_i and v_i

End
7. Select, based on fitness proportional selection, the food sources to be visited by onlooker bees

For each food source x_i chosen by an onlooker bee
8. Generate a new food source v_i by its corresponding onlooker bee (*Eq. 2*)
9. Evaluate v_i
10. Keep the best solution between x_i and v_i

End
11. Use the scout bee to replace those abandoned food sources using (*Eq. 4 for QABC and Eq. 5 for GABC)*
12. Keep the best solution between x_i and v_i
13. Save in memory the best food source so far
14. $cycle = cycle + 1$

Until *cycle*
End

Fig. 1. Pseudocode of Proposed QABC & GABC

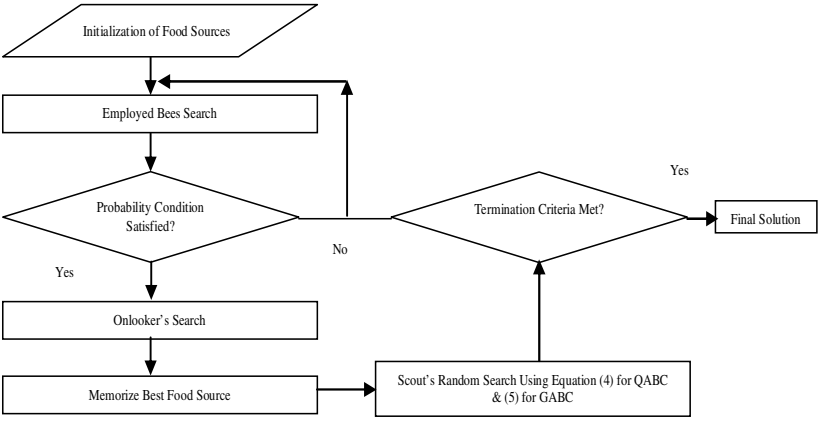


Fig. 2. Flow Graph of Proposed Variants

4 Experimental Settings, Performance Criteria and Results

Experimental Settings

The proposed algorithms are tested on 4 benchmark problems and 2 engineering problems given in Appendix. The following control parameters are: colony size (SN), MCN (Maximum Cycle Numbers) and "*limit*". ABC and the proposed variants are implemented on Dev-C++ and the experiments are conducted on a computer with 2.00 GHz Intel (R) core (TM) 2 duo CPU and 2- GB of RAM. For each problem, all the algorithms independently run 30 times. The parameter setting is taken as follows:

Population size	*100 (i.e SN=50)*
limit	*100*
Value to Reach (VTR)	10^{-15}
Maximum MCN	*5000, 8000*

Performance Criteria

Mean Fitness, Standard Deviation, Best and Worst: The average of function fitness value that an algorithm can find, using predefined MCN, is recorded in each run and then average of the function fitness values are calculated. Also the average, standard deviation, best and worst of the fitness values are calculated.

MCN: The MCN (Maximum Cycle Number) is recorded when the VTR is reached before to reach maximum MCN. i.e. we set the termination criteria as $\left|f_{optimal} - f_{global}\right| \le VTR$ and record MCN over 30 runs.

Acceleration rate (AR) in %: This criterion is used to compare the convergence speeds between ABC, GABC and QABC. It is defined as follows:

$$AR = \frac{MCN_{one\ algorithm} - MCN_{other\ algorithm}}{MCN_{one\ algorithm}}\%$$

Results

In Table-1 we have taken the results on the basis of average error. In this case MCN is fixed at 5000 and 8000, to estimate the average of minimum fitness function value in 30 runs. From the Table-1 it can be clearly observed that for all benchmark functions QABC gives better results than GABC and ABC. A two tail sample *t-test* is also applied to analyze the statistical significance of the proposed algorithm. We have checked the significant difference of QABC with respect to GABC and ABC at 5% level of significance. The calculated *t-value* (Table 3) of all function is greater than *t-table* value that shows the significantly better performance in the comparison of ABC. In the Table-2, we fixed *VTR* as given in experimental setting and then calculated the MCN of 30 runs. From Table-2 we can see that the proposed QABC gives the better results for every function in the comparison to the other algorithms. From the Table-3 it is clear that the proposed QABC is faster than GABC by 13.50% and ABC by 16.50%, when MCN=5000, D=30 &

SN=20 and in other case when MCN=8000, D=50 & SN=50, QABC is again faster than GABC by12.27% and ABC by 10.57%. Best and worst function values in 30 runs are also presented in Table-1.The results of two engineering design problems [12-13] are presented in Table 4. We see that in terms of average fitness function value and standard deviation all the algorithms gave more or less similar results although in some cases the proposed algorithms gave a marginally better performance than basic ABC and GABC.

Fig. 3(a) & 3(b) shows the convergence graph Rosenbrock's and Griekwank function. MCN taken to estimate the average of minimum fitness function value in 30 runs are also presented graphically in Fig. 4.

Table 1. Mean, Standard Deviation,Best, Worst, Mean and values obtained by ABC, GABC and QABC through 30 independent runs on function from f_1 to f_4

F	Algorithm	MCN=5000/D=30/SN=20		MCN=8000/D=50/SN=50	
		Mean	Best	Mean	Best
		Std. Dev	Worst	Std. Dev	Worts
f_1	ABC	1.162e-015	0.062e-015	2.841e-014	1.507e-014
		1.896e-015	2.235e-014	3.867e-014	4.654e-014
	GABC	8.295e-016	1.460e-016	3.159e-015	1.679e-016
		2.531e-017	3.310e-014	9.268e-016	5.130e-017
	QABC	1.437e-016	8.377e-016	2.551e-014	1.962e-014
		4.448e-017	4.626e-014	0	4.818e-014
f_2	ABC	3.750e-015	9.277e-016	1.379e-012	1.001e-013
		6.297e-014	5.245e-012	4.167e-013	1.712e-014
	GABC	1.053e-015	2.894e-016	1.313e-014	1.065e-015
		7.870e-015	5.277e-013	4.578e-015	1.579e-016
	QABC	4.256e-016	2.976e-016	1.036e-014	1.075e-015
		1.406e-015	6.041e-015	3.838e-015	1.633e-016
f_3	ABC	2.016e-002	1.953e-002	1.075e-001	1.056e-002
		8.446e-003	3.337e-002	0	3.458e-003
	GABC	2.738e-004	1.199e-004	1.181e-002	0.326e-002
		8.152e-004	7.972e-004	0	2.707e-003
	QABC	2.264e-004	9.163e-005	9.753e-003	2.808e-004
		8.389e-005	5.065e-001	0	2.983e-003
f_4	ABC	1.514e-016	8.289e-017	1.147e-015	0.472e-016
		4.309e-017	1.514e-014	3.265e-016	1.147e-017
	GABC	3.176e-017	1.982e-018	7.155e-017	3.673e-018
		9.039e-017	3.691e-016	2.036e-018	7.155e-019
	QABC	8.206e-018	2.549e-020	1.670e-017	6.006e-018
		1.333e-019	1.355e-017	2.183e-018	7.670e-019

Table 2. MCN taken by functions for VTR (NC-Not Converge)

F	**D=30/SN=20**			**D=50/SN=50**		
	ABC	**GABC**	**QABC**	**ABC**	**GABC**	**QABC**
f_1	1148	1102	1069	1737	1770	1255
f_2	4011	3903	1928	7023	6552	5790
f_3	NC	NC	NC	NC	NC	NC
f_4	1083	1009	1005	1674	1636	1610

Table 3. T-test And AR(%), (here 3/1 implies QABC /ABC, and 3/2 implies QABC/ GABC)

F	t-test				AR			
	SN=20/D=30		SN=50/D=50		SN=20/D=30		SN=50/D=50	
	3/1	3/2	3/1	3/2	3/1	3/2	3/1	3/2
f_1	1.611	40.20	0.224	72.35	6.9	3.0	27.7	29.1
f_2	0.16	0.235	9.853	1.391	51.9	50.6	17.6	11.6
f_3	7.080	0.173	1.#NF	10.385	NA	NC	NC	NC
f_4	9.969	0.781	1.#NF	55.123	7.2	0.4	3.8	1.6

1.#NF indicates std. dev. = 0.

Table 4. Mean of fitness function value and standard deviation (Std) to solve the engg. design problems by all algorithms

F	D	ABC	GABC	QABC
F_1	2	169.849 (0.00801)	169.843 (1.74e-016)	169.838 (1.670.e-016)
F_2	3	4.21319 (3.1904e-016)	4.19546 (1.0841e-016)	4.19002 (.000010)

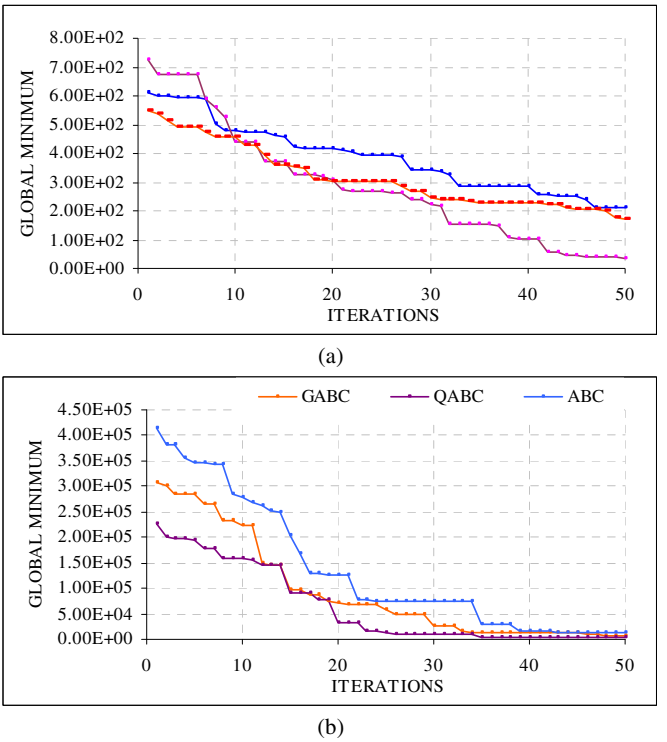
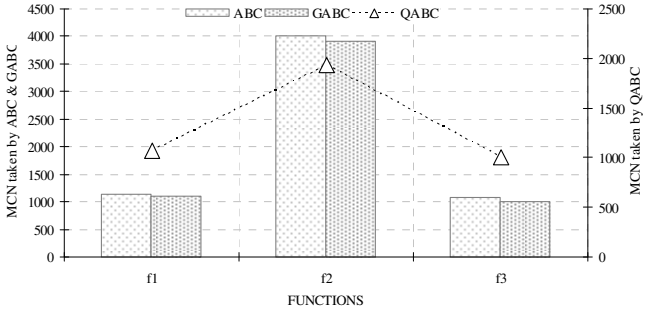
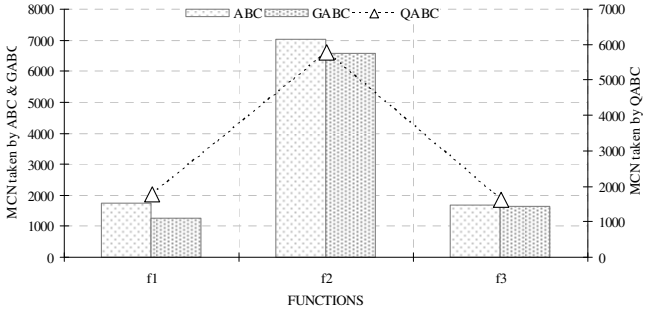


Fig. 3. Convergence plot of (a) Griekwank (b) Rosenbrock



(a)



(b)

Fig. 4. MCN taken by function f_1, f_2 & f_3 (a) D=30, SN=20 (b) D=50, SN=50

5 Conclusion

In the present study enhancements are suggested in the scout bee phase of the ABC algorithm. Instead of providing a random movement for scout bees, interpolation and Gaussian movements are embedded to generate more efficient food locations. The corresponding variants named as QABC and GABC are tested on 4 benchmark problems and two real life engineering design problems. Numerical results indicate that the proposed enhancements improve the performance of basic ABC in terms of convergence rate and fitness function.

References

1. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Kayseri, Turkey: Erciyes University (2005)
2. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39, 171–459 (2007)
3. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)

4. Singh, A.: An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied Soft Computing 9, 625–631 (2009)
5. Kang, F., et al.: Structural inverse analysis by hybrid simplex artificial bee colony algorithms. Computers & Structures 87, 861–870 (2009)
6. Samrat, L., et al.: Artificial bee colony algorithm for small signal model parameter extraction of MESFET. Engineering Applications of Artificial Intelligence 11, 1573–2916 (2010)
7. Storn, R., Price, K.: Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 23, 689–694 (2010)
8. Kennedy, J., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan-Kaufmann Publishers (2001)
9. Zhu, G.P., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation (2010), doi:10.1016/j.amc.2010.08.049
10. Akay, B., Karaboga, D.: A modified artificial bee colony algorithm for real- parameter optimization. Information Sciences (2010), doi:10.1016/j.ins. 2010.07.015
11. Alatas, B.: Chaotic bee colony algorithms for global numerical optimization. Expert Systems with Applications 37, 5682–5687 (2010)
12. Beightler, C., Phillips, D.: Applied geometric programming. John Wiley and Sons, New York (1976)
13. Prasad, B., Saini, J.: Optimal thermo hydraulic performance of artificially roughened solar air heater. Journal Solar Energy 47, 91–96 (1991)

Appendix: List of Benchmark Problems and Engineering Design Problems

Benchmark problems			
Name of function	**Definition**	**Range**	**Known global optimum**
Griekwank	$f_1(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]	$f_1(0,\ldots,0) = 0.$
Restrigin's	$f_2 = 10\,n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos\left(2\pi x_i\right)\right)$	[-5.12, 5.12]	$f_2(0\ldots,0) = 0.$
Rosenbrock's	$f_3(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1}^2 - x_i^2\right) + \left(1 - x_i^2\right)\right]$	[-30, 30]	$f_3(1,\ldots,1) = 0.$
Sphere	$f_4(x) = \sum_{i=1}^{n} x_i^2$	[-5.12, 5.12]	$f_4(0,\ldots,0) = 0.$
Engineering Design Problems			
F_1: Optimal Capacity of Gas Production Facilities [12]			
F_2: Optimal Thermohydraulic Performance of an Artificially Roughened Air Heater [13]:			

Economic Load Dispatch with Prohibited Operating Zones Using Genetic Algorithms

Piyush Jain and K.K. Swarnkar

M.I.T.S., Gwalior (M.P.)
piyushjainn89@gmail.com, kuldeepkumarsony@yahoo.co.in

Abstract. This paper present an efficient and reliable Genetic Algorithm for solving Economical Load Dispatch problem with prohibited operating zones in power plant. The main objective is to minimize the total fuel cost of generating unit having quadratic cost characteristic subjected to limits on generator true and reactive power output, transmission losses, prohibited operating zones. This paper present an application of genetic algorithms to economic load dispatch problem with prohibited operating zones of power plants for 3 generator test case system [3]. Economic load dispatch problem is applied and compared its solution quality and computational efficiency to GA other optimization techniques. The simulation results show that the proposed algorithms outperform previous optimization method.

Keywords: Genetic Algorithms, Economic Load Dispatch, Prohibited Operating Zones.

1 Introduction

One of the most relevant problems in power system is economic load dispatch (ELD). Economic load dispatch problem is a constraint optimization problem that minimizes the total operating cost of a power system while meeting the total load plus transmission load within generator limits. The operating cost plays an important role in the economic scheduling. A simplified input-output curve of thermal unit known as heat-rate curve as shown in fig1. Converting the ordinate of heat rate curve from Btu/h to Rs/h by multiplying the fuel input by the cost of fuel in $/Btu which result in fuel cost curve as shown in fig2.

Basically, the fuel cost function of generator is represented by single quadratic function. Various conventional method such as lambda iteration, gradient method, participation factor method etc which are only suitable for fuel cost curve with linear and monastically increasing function has been applied to solve ELD problem with quadratic cost function [2-7] .But in the case, if the input-output fuel cost characteristic of generator are highly non-linear and discontinuous due to

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 611–619.
springerlink.com

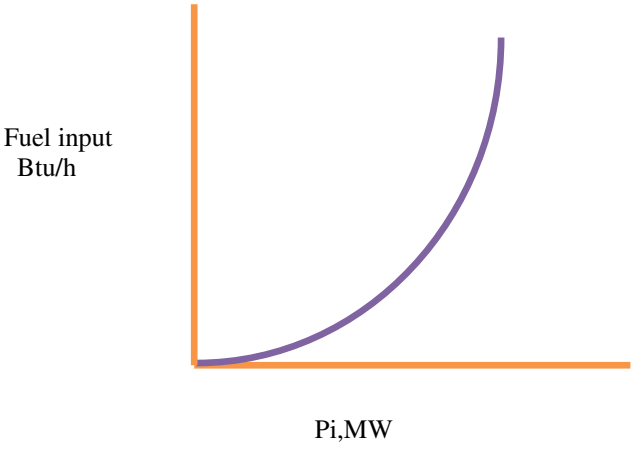


Fig. 1. Heart Rate Curve

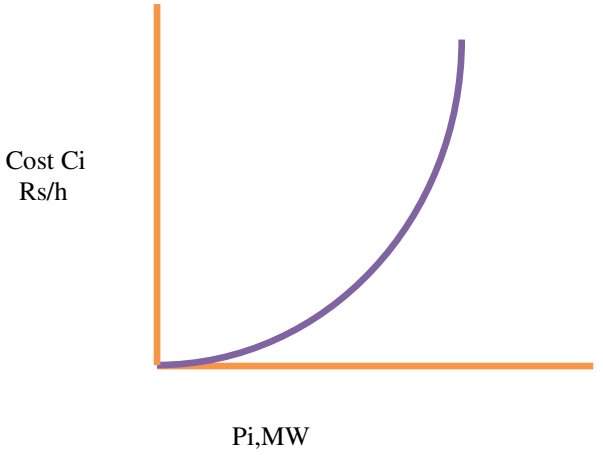


Fig. 2. Fuel Cost Curve

prohibited operating zones. Some of these techniques may not be able to produce good solution in large practical thermal generation. Even in this case, the solution could be far from optimal and very time consuming.

In practice, all the operating zone of generator unit is not always available for load allocation due to physical operation limitation. Fault in generation machine or their associated auxiliaries occasionally exist. So prohibited operating zones could be formed between maximum and minimum generation limits. A unit with these prohibited zones results in unavailable amplification of shaft bearing vibration. a unit with prohibited operating zones transform the ordinary economic dispatch to

a non convex optimization problem where the conventional based method cannot be applied. The discontinuous input output power generation characteristic provided by the unit with prohibited operating zones is shown in fig3.

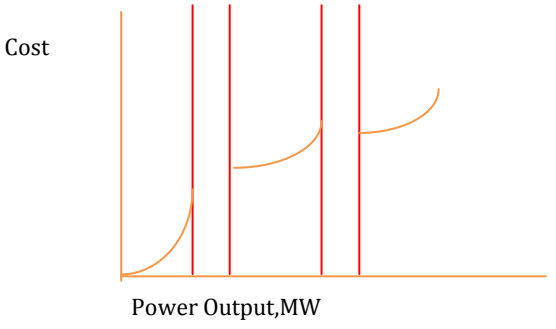


Fig. 3. Unit input/output characteristic

Recently, modern Meta heuristic and behavioral random search algorithms such as particle swarm optimization (PSO), ant colony optimization, differential evolution, simulated annealing etc have been applied successfully to ELD problem.

In this paper, genetic algorithm is investigated for determine the optimal loading of generator in power system where some of the units have prohibited operating region. This new optimization technique was first introduced by john Holland. Like PSO, a genetic algorithm has a way of sharing information between solutions. This paper present a more robust genetic algorithms model applied to a practical operational problem. In a power system, economic operation of generating unit required the units to share the total load demand in an optimal manner in order to minimized fuel costs.

The remainder of this paper is organized as follow-section 2 provides a brief description and mathematical formulation of different type of ELD problem. In section 3, the concept of genetic algorithms is discussed. The section 4 shows the result. Finally, the conclusion and future work of research are outlines in section 5.

2 Problem Formulation

The economic load dispatch may be formulated as a non linear constrained problem. Both convex and non convex ELD problem have been modeled in this paper. The convex ELD problem assumes quadratic cost function along with system power demand and operational limit constraint. The practical non convex ELD problems, in addition, consider generator non linearity's as prohibited operating zones. The objective function F_t of ELD problem may be written as

$$\text{Min } F_t = \sum_{i=1}^{m} Fi(Pi) \quad (1)$$

In (1), the fuel cost function $F_i(P_i)$ of the i^{th} generating unit is usually expressed as a quadratic polynomial:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \quad (2)$$

Where a_i, b_i, and c_i, i=1, 2, 3….,m, are the fuel cost coefficient of i^{th}generating unit, P_i is the power output of the i^{th} generating unit. m is the total number of generating units.

Subjected to the constraints:

1. Real Power Balance Constraint:

$$\sum_{i=1}^{m} Pi = P_D + PL \quad (3)$$

Where P_D is the total system demand and P_L is the total transmission loss and sit's expressed as

$$P_L = \sum_{i=1}^{m} * \sum_{j=1}^{m} P_i B_{ij} P_j + \sum_{i=1}^{m} B_{0i} P_i + B_{00} \quad (4)$$

2. Generator Capacity Constraint: the power generated by each generator will be within their lower limit P_i^{min} and upper limit P_i^{max}. so that

$$P_i^{min} \le P_i \le P_i^{max} \quad (5)$$

3. Prohibited Operating Zones: The Prohibited Operating Zones in the input–output performance curve for a typical thermal unit are because of a steam valve operation or vibration in the shaft. For unit i with POZs, the feasible operating zones can be described as follows:

$$P_i^{min} \le P_i \le P^l_{i,1} \quad (6)$$

$$P^u_{i,j-1} \le P_i \le P^l_{i,j} \; j=2,3,\ldots\ldots n_i \quad (7)$$

$$P^l_{i,ni} \le P_i \le P_i^{max} \quad (8)$$

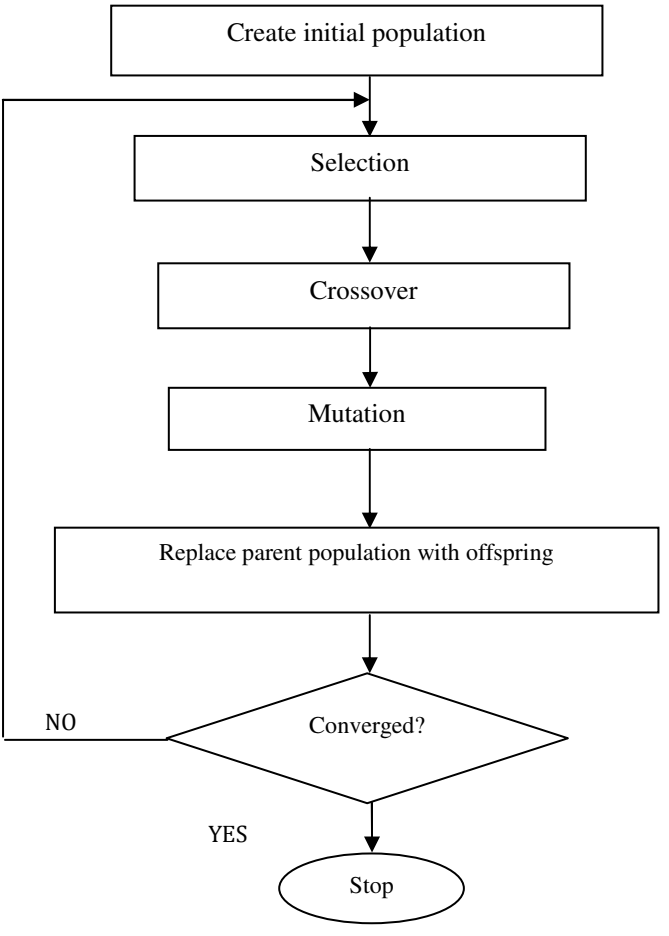
Where j represents the number of prohibited operating zones of unit i. $P^l_{i,j}$ is the lower limit of j^{th} prohibited operating zone and $P^u_{i,j-1}$ is the upper limit of $(j-1)^{th}$ prohibited operating zone of i^{th} unit. n_i is the total number of POZ of i^{th}unit.

It is noted that unit i with n_i prohibited zones will have n_i +1 disjoint operating regions. These disjoint regions form a nonconvex set.

3 Genetic Algorithms

Genetic algorithms are adaptive algorithms for finding the global optimal solution for an optimization problem. The genetic algorithms develop by john Holland is characterized by binary representation of individual solution, simple problem independent crossover and mutation operators and a proportional selection rule. Genetic algorithms operate on a population of potential solution applying the principal of survival of fittest to produce better and better approximation to a

solution. At each generation, a new set of approximation is created by the process of selecting individual according to their level of fitness in problem domain. A flowchart for a genetic algorithm is shown in fig.4.



Representation of parameter: In genetic algorithms, the parameters are represented as a string of binary number 0 and 1. So each parameter of given problem is coded with string of bits. The individual bit is called gene. The total string of such gene of all parameter written in a sequence is called chromosome. For example, if each parameter are x_i ,i=1, 2,3……n is coded in a string of length q, a parameter vector is represented using a string of total length nq. This string of total length nq is called chromosome. The general scheme of genetic algorithms starts from a population of randomly generated candidate solution (chromosome). If variable x is represented by a string of q binary digit and x_{max} and $x_{min}$are lower and upper bound of variable x, then its decimal value can be calculated by using this formula

$$X=x_{min}+ [b*(x_{max}-x_{min})/2^{q}-1]$$

Where b is a discrete value.

After selecting string representation, a random population of solution is generated. Then fitness is given to each population member, which represent the goodness of each solution .the string is evaluated in the context of objective function and constraints. If the constraint is absence, then objective function is treated as fitness function.

GA operator: - Genetic operators are used to generate new population from the previous population. These genetic operators are reproduction, crossover, and mutation.

Reproduction is the operator used to copy the old chromosome into matting pool according to their fitness. Chromosome with a high level of fitness is to be retained while the one with low level of fitness is discarded. The various method include roulette-wheel selection, tournament selection, rank selection, Boltzmann selection etc are used to select chromosome for parent to crossover. After reproduction, crossover operation is implemented. Crossover is the basic operator for producing new chromosome. In this, information is exchanged among string of matting pool to create new string. Crossover operator is proceeding in three steps. First, two individual string are selected from the matting pool generated by the reproduction operation. These are called parents chromosome. Then crossover sites are selected at random along the string length and then, the position value are swapped between two parents chromosome at crossover point. These are called child chromosome. This process is repeated until population is filled with new chromosomes. The final genetic operators are mutation. Mutation is a random process where one ales of gene is replaced by another to produce new genetic structure. Thus in mutation, 0 is changed to 1 and vice versa, at a random location.

Genetic algorithms is a very simple and straight-forward .reproduction select best string, cross-over operator recombine them to produce two better string and the mutation operator alter the string locally.

4 ELD Using Genetic Algorithms

For solution of ELD using GA, incremental fuel cost of the generators i.e. lambda is encoded in the chromosome. The algorithm for implementing ELD without losses using genetic algorithm is as follows.

1. Read population size, chromosomelength, unitdata, Pdemand, Probability of Elitism,crossover and mutation.
2. Randomly generate population of chromosomes.
3. Decode the chromosomes using(6).
4. Lambda_act=Lambda_min +(Lambda_max-lambda_min)*Decoded lambda
5. Use the Lambda_act and cost coefficients of the generators,and calculate real power output of the generators(P_{gen}).
6. Calculate the error of each chromosome as(SUM of P_{gen})-pdemand.
7. Fitness(i) of each chromosome is calculated as 1/(1+error(i)/Pdemand).
8. Arrange the chromosomes in the descending order of their fitness.

9. Check if error(1),≤0.0001*Pdemand
10. If yes STOP and calculate Optimal fuel cost and P_{gen} of units Else
11. Check if Fitness(1)=Fitness(last chromosome)
12. If yes print 'All chromosomes have equal value',calculate Optimal fuel cost and $P_{gen}$of units and STOP.
    Else
13. Apply elitism,Reproduction(RWS),crossover and mutation and generate new population from old one.
14. Update generation count.
15. Check if Generation count>maximum generations?
16. If yes,print 'Problem not converged in maximum number of generations', STOP.
    Else
17. Repeat from step 3.

5 Simulation Results

The develop algorithm is tested on standard 3 generator. For every case the chromosome length, population size, probability of crossover, mutation considers with GA and conventional method.

The test problem is based on a 3 unit power system [3] with three prohibited operating zones. The genetic parameter considered for this case are

Population size: 30,
Chromosome length: 16bits,
Max no of generations: 100,
Crossover probability: 0.9,
Mutation probability: 0.01.

Table 1. Unit characteristics

unit	a (Rs/H)	b (Rs/MWH)	c (Rs/MWH²)	$P_{max(MW)}$	$P_{min(MW)}$
1	561	7.92	0.001562	600	150
2	310	7.85	0.001942	400	100
3	78	7.97	0.00482	200	50

Table 2 represent the unit prohibited operating zones and Table 3 represent the transmission losses coefficient.

Table 2. Unit prohibited operating zones

Unit	Prohibited regions		
	Zone 1	Zone 2	Zone 3
1	[185,225]	[305,335]	[240,450]

Table 3. Transmission loss coefficient matrix

Ploss	P1	P2	P3
	0.00003	0.00009	0.00012

Table 4. Output power generation without loss

GA	
Pdemand(MW)	850
Pg1(MW)	393.26
Pg2(MW)	334.67
Pg3(MW)	122.25
FC($/Mwh)	8196.0
Time(sec)	0.078

Table 4 present the result of standard 3 generator system for a power demand of 850 without considering transmission losses.

Table 5. Output power generation with loss

GA	
Pdemand(MW)	850
Pg1(MW)	435.219
Pg2(MW)	299.983
Pg3(MW)	130.667
FC($/Mwh)	8344.96
Loss	15.8304
Time(sec)	0.28

Table 5 present the result of standard 3 generator system for a power demand of 850 while considering transmission losses. From table 6 and table 7,it can be seen that the GA model perform better than the other conventional method. With enhancement such as multiple resolutions search mechanisms' added to the conventional method. The performance improves s and its almost as good as the GA.

Table 6. Comparison the result of 3 variable methods with loss

	GA	Lamda iteration	Gradient method
Pdemand(MW)	850		
Pg1(MW)	435.219	432.459	431.986
Pg2(MW)	299.983	297.562	296.898
Pg3(MW)	130.667	127.935	126.795
FC($/Mwh)	8344.96	8434.56	8596.34
Loss	15.8304	16.2956	15.9828
Time(sec)	0.28	1.96	1.21

Table 7. Comparison the result of 3 variable methods without loss

	GA	Lamda iteration	Gradient method
Pdemand(MW)	850		
Pg1(MW)	393.26	393.14	393.07
Pg2(MW)	334.67	334.49	334.46
Pg3(MW)	122.25	122.17	122.11
FC($/Mwh)	8196.0	8347.01	8211.47
Time(sec)	0.078	0.96	0.83

6 Conclusion

The paper presented a Genetic Algorithm for Economic Load Dispatch problem. For the same Genetic operator and parameter, the proposed algorithm show better and faster convergence .Large computational times involved with Gas can be overcome by assessing the variation of error with the solution at hand. A number of cases have been studied and the algorithm showed reliable convergence. The algorithm can be used for real time ELD problem as the solution accuracy is achieved in minimum time. The only limitation of the algorithm is, if the solution represent a single variable (lambda in this case), the error variation cab be easily assessed with respect to this single variable. But if the chromosomes represent a number of variables, the error variation with respect to all the variables cannot be predicted and it became difficult to restrict the search space to small regions.

References

[1] Chowdhary, B.H., Rahman, S.: A Review of recent advances in economic load dispatch. IEEE Trans. Power Systems PWRS 5(4), 1248–1257 (1990)
[2] Huneault, M., Galiana, F.D.: A Survey of the optimal power flow Literature. IEEE Trans. Power System PWRS 6(2), 762–770 (1991)
[3] Wood, A., Wollenberg, B.F.: Power Generation Operation and control. A wiley-interscience publication, John Wiley & Sons. INC. (1996)
[4] Laoufi, A., Hazzab, A., Rahli, M.: Economic Dispatch using Fuzzy Genetic Algorithm. International Journal of Applied Engineering Research 1(3) (2006) ISSN 0973-4562
[5] Chen, C.-H., Chen, Y.-P.: Real Coded ECGA for Economic Dispatch. In: GECCO 2007, London, England, United Kingdom, July 7-11 (2007)
[6] Chen, P.-H., Chang, H.-C.: Large-Scale Economic Dispatch by Genetic Algorithm. IEEE Trans. On Power System 10(4) (November 1995)
[7] Haupt, R.L., Haupt, S.E.: Practical genetic algorithms, 2nd edn. John Wiley and Sons, Inc., Publication Hoboken, New Jersey (2004)

Optimization of QoS Parameters for Channel Allocation in Cellular Networks Using Soft Computing Techniques

Narendran Rajagopalan and C. Mala

Department of Computer Science and Engineering, National Institute of Technology, Trichy, 620015
narenraj1@gmail.com, mala@nitt.edu

Abstract. The usage of mobile communications systems has grown exponentially. But the bandwidth available for mobile communications is finite. Hence there is a desperate attempt to optimize the channel assignment schemes. In this work, some of the quality of service parameters such as residual bandwidth, number of users, duration of calls, frequency of calls and priority are considered. Genetic Algorithm and Artificial Neural Networks is used to determine the optimal channel assignment considering the quality of service parameters. The simulation results show that Genetic Algorithm performs better than Heuristic Method. But application of Artificial Neural Networks outperforms Genetic Algorithm and Heuristic method by a considerable margin. Channel allocation can be optimized using these soft computing techniques resulting in better throughput.

Keywords: Genetic Algorithm, Channel Allocation, Quality of Service, Artificial Neural Network, Throughput.

1 Introduction

The evolution of Mobile Communication System has been rapid. The number of cell phones has exceeded the number of PSTN phones globally. But unfortunately the wireless bandwidth available for mobile communication is fixed and cannot be increased. Thus making optimization of channel assignment schemes an important area of research in the wireless community.

A very successful technology in the Mobile Communication Systems is Global System for Mobile Communication (GSM). GSM divides geographical regions into cells. The adjacent cells cannot use the same frequency at the same time which would result in interference and low quality of service. There has been considerable amount of research literature into channel assignment [1,4,5,10] schemes sharing the frequencies among cells minimizing interference. Not many literatures exist which concentrates on quality of service parameters like available residual

K. Deep et al. (Eds.): Proceedings of the International Conf. on SocProS 2011, AISC 130, pp. 621–631.
springerlink.com

bandwidth, Number of Users in a cell, Periodicity of calls, Priority of calls, time of call and frequency of calls. In this paper, these Quality of Service parameters are taken into account. Genetic Algorithm and Artificial Neural Networks is applied to these parameters to optimize the channel assignment.

1.1 Introduction to Genetic Algorithms

Genetic Algorithms[8,9] are heuristic adaptive search algorithm based of the evolutionary idea of natural selection and genetic, used to simulate processes in natural system necessary for evolution. Introduced by John Holland in 1960, Genetic Algorithms can be applied in many fields in engineering.

The operations of Genetic Algorithms are simple such as, copying of strings, swapping partial strings, etc. The parameters are represented as strings of bits. Initial population is created randomly and subjected to the fitness function for evaluation. A simple set of operations is performed on the initial population to generate successive population which closes towards optimized results.

The simple set of operations performed in Genetic Algorithm is:

Reproduction : is a process of selecting strings with better fitness value compared to strings with lower fitness value for the next generation. Selecting an individual string according to its fitness value leads to a higher probability of it contributing a better offspring in the next generation.

Crossover : is a process of exchanging information among strings selected during reproduction. The set of individuals are selected at random and mated. Mating is the process of selecting a position called crossover point, within the length of the string and exchanging the bits between the two strings. Crossover together with reproduction gives optimized results with recombination of individuals.

Mutation : is the process of changing a single bit information of the string. Mutation takes place rarely at the rate of one in one thousand reproductions. It reflects the biological mutation taking place over a long period of time.

1.2 Introduction to Artificial Neural Networks

Artificial Neural Network (ANN)[2,3,6,7] is similar to biological neural network in its structure and some features. Artificial neural network is mainly used in optimization problems, it may be either minimization or maximization problem. ANNs are implemented on computers. The capabilities of an ANN involve

- Processing of Parallel operations
- Flexibility in adjustment of the system weights while training the system so that the ANN can be adjusted to the new environment.

Structure of ANN:

The ANN is composed of an input unit, processing unit and an output unit. All units are interconnected according to some topologies such as bidirectional, auto associative, group of instar, group of outstar, etc. The processing unit performs

summing operations & updating weights based on the desired output. The input and output units may be continuous, discrete, deterministic or stochastic.

The structure of ANN consists of interconnected processing unit [5] as shown in Fig.1 which contains Summation Unit. The output of the summation part is calculated based upon the given input values (may be continuous or discrete data) and the weights (either positive or negative) assigned to those input units. The computed value (weighted sum) is called as the activation value. The output unit (continuous or discrete) produces a signal from the activation value. The weights in the network are adjusted in order to store the output pattern in a network. The process of adjusting the weights is known as learning. The learning law or algorithm follows the procedure for updating each of its weights. Learning can be performed either as supervised or as an unsupervised manner.

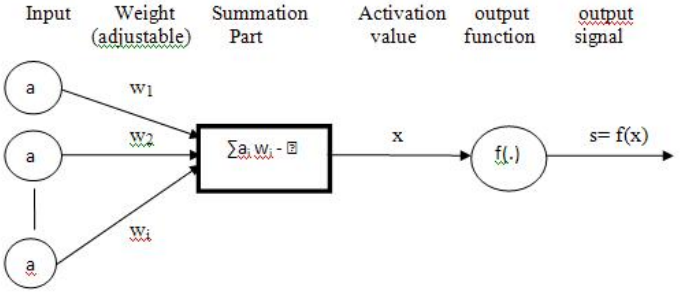


Fig. 1. Basic Model of ANN

$$\text{Activation value } (x) = \sum_{i=1}^{M} a_i w_i - \theta \tag{1}$$

Where a_i is set of inputs, w_i is set of weights and θ is bias term

$$\text{Output signal } (s) = f(x) \text{ [output function]} \tag{2}$$

Learning requirements

The requirements of learning laws[2] are given below

- Learning leads to convergence of weights.
- Learning time should be small (for capturing the pattern information from samples)
- Learning uses only local information (change in the weight on a connecting link between 2 units depends on the states of these 2 units only). Due to this, learning can be done in parallel. Hence it speeds up the learning process.
- All possible patterns in the network should be captured by learning.

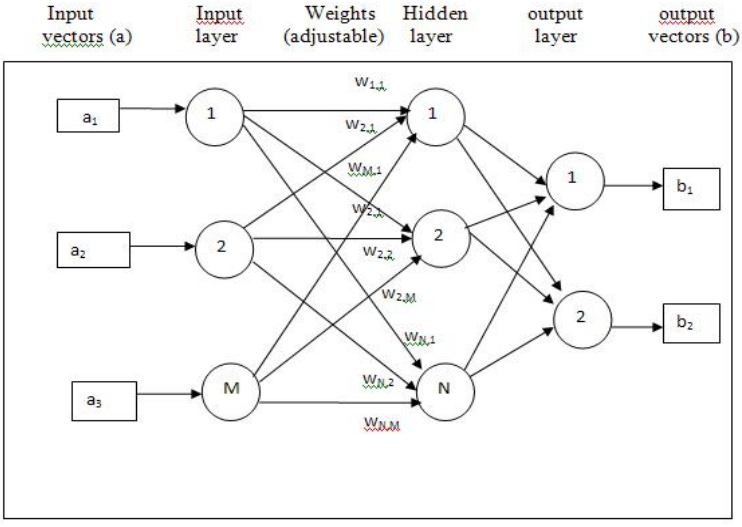


Fig. 2. Basic Feedforward Neural Network

2 Proposed Scheme

There are two distinct classifications of channel allocation schemes called as Fixed Channel Allocation and Dynamic Channel Allocation. Frequency allocation is static in fixed channel allocation. But the dynamically selected channel minimizes interference in dynamic channel allocation. Whenever there are call requests and lesser number of channels free for assignment to all requests, some of the quality of service parameters must be considered to decide on who should be allotted the channel. The quality of service parameters such as duration of calls, number of users, residual bandwidth, priority of calls, time of call and frequency of calls which can be considered for decision making of channel allocation. Genetic Algorithm, heuristic method and Artificial Neural Network is applied to these parameters for optimal allocation.

2.1 Quality of Service Parameters Considered for the Proposed Problem

Duration of Calls(a) : Parameter 'a' is referred to as the duration of a call by a user. If there are a number of call requests, the call with maximum duration must be given priority so that it yields maximum profit to the service providers. The range of call times is obtained using recorded history of usage by users over a period of time to find the average range of time that users hold the channel allocated to them.

Number of Users(b) : The number of Users is a cell making a call request is another important parameter affecting the quality of service. If the number of users is less than the number of available channels, then the call request can be

granted without affecting service of others. But when the number of users are more than the channels available, it would lead to forced termination of some calls, hence affecting the quality of service.

Residual Bandwidth(c) : Residual Bandwidth or the number of free channels available in a cell is another Quality of Service parameter which can affect the service quality.

Priority of calls (d) : depending upon whether it is a handoff request or a new call request, the priority parameter can be set. The handoff call must be given the highest priority to avoid disconnection of ongoing connection.

Time of Call (e) : Time of the day can be classified into number of slots and assign different priorities to each slot.

Periodicity(f) : Periodicity is defined as the frequency of calls by a user. If the frequency of calls from a user is high, then denying a call request will not affect as much as denying a call request from an infrequent user.

3 Application of Genetic Algorithm

In order to apply genetic algorithm to the problem, The parameters have to be encoded to strings. The parameters are represented as eight bit strings. Some of the parameter values are obtained from the history of the caller. The time duration of a call by a user is represented by Parameter 'a', with 10 seconds assumed as one unit. A value of a = 5, would imply the periodicity of the call to be 50 seconds. The parameter priority is higher if the value of d is higher. Hence if d = 100, then it has a higher priority than $d_2 = 65$, but lesser than $d_3 = 150$. Depending upon the density of users in a cell, a unit is selected like 10 users as one unit. Hence a value of b=45 would indicate 450 users. A fitness function is needed and initial population as seed is required. The fitness function is given by,

$$F(X) = a + 1/b + c + d + e + 1/f \qquad (3)$$

The fitness function for this application is a maximization function. The seed values are randomly generated and tabulated as shown in table below.

4 Application of Artificial Neural Network

Radio spectrum is limited in wireless communication systems. Therefore the system capacity is improved by means of effective use of available channels. The channel allocation problem [3,4,5] is used to allocate channels by means of minimizing the probabilities of call blocking or dropping and by maximizing the quality of service. So the channel allocation problem is an optimization problem also Np- hard. Hence it uses neural network for solving the problem.

The feedforward neural network as shown in Fig. 2 is used for optimized channel allocation problem proposed in this paper. The assignment of channel is based on the output units of neural network.

Table 1. Randomly Generated Seed Sample

Sl. No	Duration of call(a)	Number of Users(b)	Residual Bandwidth(c)	Priority(d)	Time of call(e)	Periodicity(f)	F(x)
1	225.0	18.0	228.0	226.0	255.0	255.0	934.06
2	251.0	10.0	199.0	248.0	255.0	255.0	953.10
3	252.0	110.0	225.0	248.0	255.0	255.0	980.013
4	70	62	9	34	133	27	246.05
5	121	195	203	90	226	62	640.02
6	20	155	167	182	178	55	547.02
7	78	203	75	185	184	53	522.02
8	138	136	37	160	244	186	579.01
9	189	161	76	214	108	158	587.01
10	56	241	58	70	102	51	286.02
11	107	147	82	16	151	207	356.01
12	36	41	190	72	146	149	444.03
13	136	96	217	125	66	161	544.02
14	74	4	19	206	242	84	541.26
15	199	14	104	151	20	143	474.08
16	103	159	244	221	148	249	716.01
17	92	73	94	165	239	153	590.02
18	50	43	16	172	18	214	256.03
19	193	111	223	85	60	9	561.12
20	238	216	73	251	39	83	601.01

The input unit contain 6 parameters for the proposed method. Parameters 'a', 'c', 'd', 'e' are directly proportional and 'b' and 'f' are inversely proportional to the output. The probability of the usage per day indicates weight in the network. The unsupervised learning method is adopted in the proposed method. The method uses Instar (Winner take all) learning law [6,7] which is unsupervised learning process. Here the initial weight assigned to the network is random prior to learning and vectors are normalized during learning. In Instar learning method, all the inputs are connected to each of its output layer (1, 2 … N) in a feedforward manner. The summing part has to be calculated using input vector $a = (a_1, a_2, \ldots a_M)$ and weights ($w_{i,j}$ indicates weight assigned in the link connecting i^{th} output node with j^{th} input node).

The output for each unit i is computed as $w_i^T a$. The maximum output of the unit (k) is found out by using

$$w_k^T a = \max (w_i^T a) \tag{4}$$

The weight vector is adjusted to the k^{th} unit as follows

$$\Delta w_k = \eta (a - w_k) \tag{5}$$

where η is learning rate parameter (that affects the convergence speed and stability of the weights during learning)

$$\Delta w_{kj} = \eta (a_j - w_{kj}) \quad \text{for } j = 1 \text{ to } M \tag{6}$$

The error measure depends on the parameter. If there is any change in the parameter then that affects the error measure. Hence there exists a inter relationship between parameter and error measure.

Table 2. Input vector (10 different set of values generated randomly from 1 to 255 for 6 input parameters)

a	70	121	20	78	138	189	56	107	36	136
(1/b)	0.0161	0.0051	0.0064	0.0049	0.0073	0.0062	0.0041	0.0068	0.0243	0.0104
c	9	203	167	75	37	76	58	82	190	217
d	34	90	182	185	160	214	70	16	72	125
e	133	226	178	184	244	108	102	151	146	66
(1/f)	0.0370	0.0161	0.0182	0.0189	0.0053	0.0063	0.0196	0.0048	0.0067	0.0062

Table 3. Six Different Weights considered

wi \ j	1	2	3	4	5	6
w1	0.1	0.1	0.2	0.3	0.1	0.2
w2	0.3	0.1	0.1	0.15	0.25	0.1
w3	0.241	0.123	0.248	0.176	0.058	0.154
w4	0.2	0.3	0.1	0.2	0.1	0.1
w5	0.345	0.124	0.045	0.164	0.151	0.171

Table 4. Activation Value for ten different set of Inputs

a	70	121	20	78	138	189	56	107	36	136
(1/b)	0.0161	0.0051	0.0065	0.004926	0.007353	0.006211	0.004149	0.0068	0.0244	0.0104
c	9	203	167	75	37	76	58	82	190	217
d	34	90	182	185	160	214	70	16	72	125
e	133	226	178	184	244	108	102	151	146	66
(1/f)	0.0370	0.0161	0.0182	0.018868	0.0053	0.006329	0.019608	0.0048	0.0067	0.0062
X (for w1)	32.3090	102.3037	107.8043	96.70427	93.6018	109.1019	48.40434	47.0016	77.8038	101.1023
X (for w2)	60.2553	126.6021	94.5024	104.6524	130.1013	123.4013	58.60238	80.4511	77.1031	97.7516
X (for w3)	32.8077	108.4561	88.5956	80.63351	84.7477	108.3267	46.11953	57.6985	76.9400	112.4222
X (for w4)	35.0085	85.1031	74.9037	78.50336	87.7027	99.0025	41.20321	47.9025	55.2079	80.5037
X (for w5)	50.2223	99.7694	71.1449	88.41284	112.3608	120.0309	48.81587	66.0317	54.8281	87.1533

5 Performance Analysis with Heuristic Method,Genetic Algorithm and Neural Networks

Testing the performance of the system was done using Matlab. For each case study, any two parameters are constantly increased at a predetermined interval and the performance of Heuristic Method, Genetic Algorithm and Artificial Neural Network is recorded varying other parameters. Throughput in the figures is the percentage of best theoretical case to the throughput achieved by Genetic Algorithm. The graphs in Figure 3, Figure 4, Figure 5, Figure 6 and Figure 7 compare

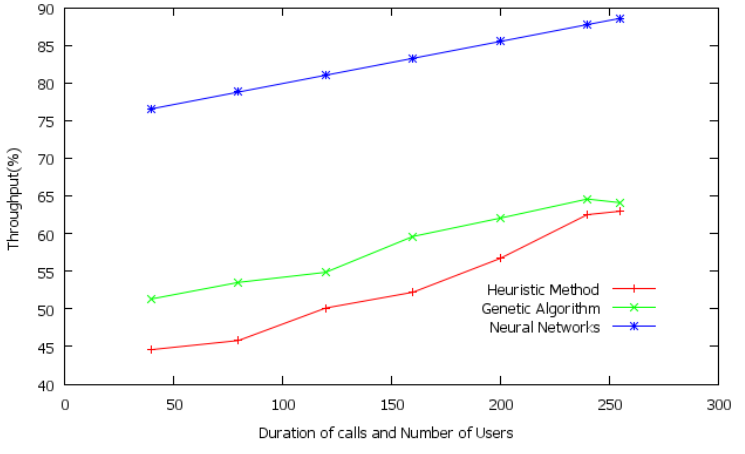


Fig. 3. Performance of Heuristic Method, Genetic Algorithm and Neural Networks with Parameters Duration of calls and Number of Users fixed at different Intervals

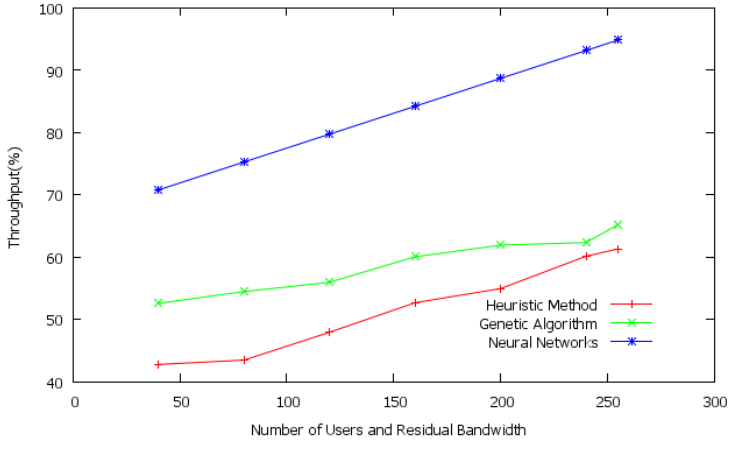


Fig. 4. Performance of Heuristic Method, Genetic Algorithm and Neural Networks with Parameters Number of Users and Residual Bandwidth fixed at different Intervals

the throughput performance of Heuristic method, Genetic Algorithm and Artificial Neural Network for different parameters. From the graphs, it is clear that Neural Networks outperform other methods.

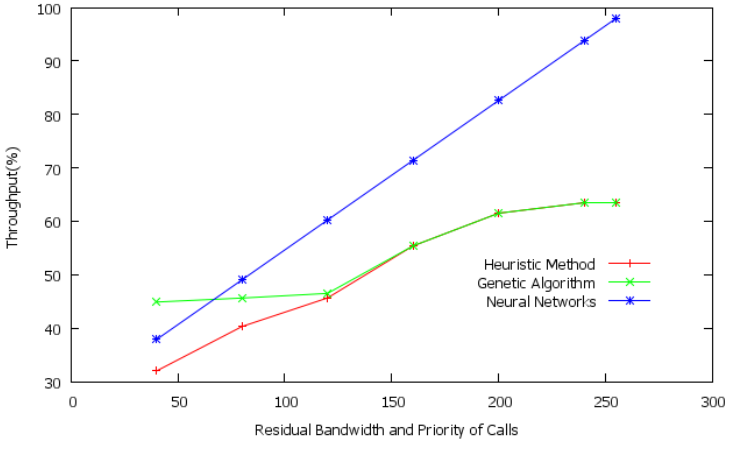


Fig. 5. Performance of Heuristic Method, Genetic Algorithm and Neural Networks with Parameters Residual Bandwidth and Priority of calls fixed at different Intervals

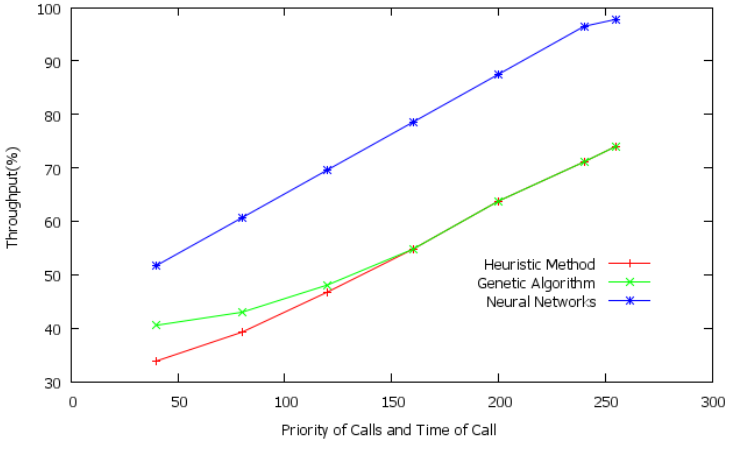


Fig. 6. Performance of Heuristic Method, Genetic Algorithm and Neural Networks with Parameters Priority of calls and Time of Call fixed at different Intervals

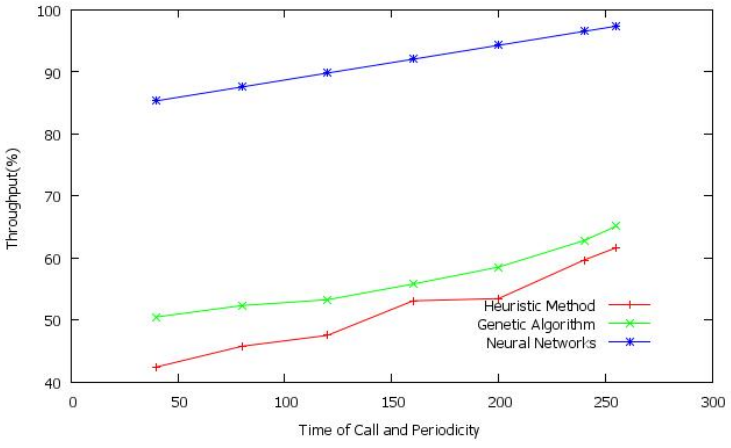


Fig. 7. Performance of Heuristic Method, Genetic Algorithm and Neural Networks with Parameters Time of Call and Frequency of Calls fixed at different Intervals

6 Conclusion

The quality of service parameters such as Duration of Call, Number of Users, Residual Bandwidth, Priority, Time of Call and Frequency of Calls are considered for channel allocation problem. Genetic Algorithm and Artificial Neural Networks are applied on these parameters for optimization of channel allocation. From the results obtained, it can be concluded that application of Genetic Algorithm increases the throughput compared to heuristic method. But application of Artificial Neural Network gives the maximum throughput compared to the other methods.

References

[1] Battiti, R., Bertossi, A., Cavallaro, D.: A randomized saturation degree heuristic for channel assignment in cellular radio networks. IEEE Trans. Veh. Technol. 50, 364–374 (2001)

[2] Yegnanarayana, B.: Artificial neural networks. Prentice Hall of India (2001)

[3] Vidyarthi, G., Ngom, A., Stojmenovic, I.: A hybrid channel assignment approach using an efficient evolutionary strategy in wireless mobile networks. IEEE Transactions on Vehicular Technology 54(5), 1887–1895 (2005)

[4] Smith, K., Palaniswami, M.: Static and dynamic channel assignment using neural networks. IEEE Journal on Selected Areas in Communications 15(2), 238–249 (1997)

[5] Elhachmi, J., Guennoun, Z.: Evolutionary neural networks algorithm for the dynamic frequency assignment problem. International Journal of Computer Science and Information Technology 3(3), 49–61 (2011)

[6] Chan, P.T.H., Palaniswami, M., Everitt, D.: Neural network – based dynamic channel assignment for cellular mobile communication systems. IEEE Transactions on Vehicular Technology 43(2), 279–288 (1994)

[7] Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro–fuzzy and soft computing–A computational approach to learning and machine intelligence. PHI Learning (2010)
[8] Wikipedia, http://www.wikipedia.org/
[9] Pandian, J., Murugiah, P., Rajagopalan, N., Mala, C.: Optimization of Dynamic Channel Allocation Scheme for Cellular Networks Using Genetic Algorithim. In: Nagamalai, D., Renault, E., Dhanushkodi, M. (eds.) PDCTA 2011. CCIS, vol. 203, pp. 628–637. Springer, Heidelberg (2011)
[10] Kaabi, F., Ghannay, S., Filali, F.: Channel allocation and routing in Wireless Mesh Networks:A Survey and qualitative comparison between schemes. International Journal of Wireless and Mobile Network 2(1), 132–151 (2010)

An ACO-GA Optimization Scheme for Route Discovery in Cellular Networks

C. Mala, A. Gokul, Anand Babu, R. Kalyanasundaram,
and Narendran Rajagopalan

Department of Computer Science and Engineering, National Institute of Technology, Trichy- 620015
mala@nitt.edu, {gokul.a,mail2anand,
kalyansundar.r,narenraj1}@gmail.com

Abstract. The recent advancements in mobile communication has triggered mankind in every part of the world to use wireless devices viz., cell phones, laptops, PDAs, etc., To provide excellent services to the mobile users, every service provider expands his network to cover a wide coverage area. To reduce the establishment cost of service providers, infrastructure sharing among service providers is becoming popular. This requires existing connectivity/ topology in a Cellular Network (CN) to be explored and discovered. In this paper, a novel method for topology discovery of CN using Ant Colony Optimization (ACO) is proposed. In this approach, the Base Station(BS)s simulate the way ants forage for food to find out the routes to other BSs. The route discovered by each ant is associated with trail (pheromone) strength which in turn decides whether the route is the best or not. ACO applied to CN gives all the existing routes between the various BSs in a CN. Genetic Algorithm (GA) based optimization is further applied to get the optimal path satisfying multiple constraints viz., number of hops, Poisson traffic distribution, buffer capacity, link delay, queuing delay, and residual bandwidth from the set of paths given by ACO. Our simulation results show that different network models viz., Random model, unidirectional ring, bidirectional ring, star, and tree are explored faster using ACO and our scheme using ACO-GA outperforms the scheme without GA with respect to Call Service Rate and Call Dropping Rate.

Keywords: Cellular network, ant colony optimization, pheromone, topology.

1 Introduction

The exponential growth in Wireless communication enables people in different parts of the world to participate in nomadic computing [1] with the help of a CN [2,3]. With the change in the lifestyle of people in recent years, wireless links enable Mobile Users (MU) of a CN to participate in real time applications, viz., video conferencing, multiparty games, online auctions, access to distributed databases,

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 633–643.
springerlink.com

group communication, etc., on the fly. In all these scenarios, routing of information has to be done faster. Existing routing protocols in a wired network when extended to mobile networks fail because of the inherent dynamism in a mobile network. Moreover, the drastic growth in the number of MU's demand increase in the size of the CN. Every service provider needs to know the existing connectivity in his network or other networks so that he can plan either for sharing of infrastructure or expansion in infrastructure to accommodate more MUs and to provide them excellent service. As the network models become more and more complex, it is required that the future routing protocols are more robust and adaptable to react effectively and favorably to changes in traffic, network connectivity etc. One such important model, the Ant Colony Optimization (ACO) meta-heuristic [4,5,7,9,10] is a generic framework for ant-based optimization algorithms. Additionally two sets of mobile agents are used for establishing call connections. With the increase in the number of MUs, these two mobile agents will be overloaded and it will be difficult to perform frequent computation of the optimal route faster to support the inherent dynamism in a CN. In this work, *Ant Agents* are modeled as mobile agents that aim at local optimization and can backtrack, if required, to get the global optimum. Ant agents lay pheromones for others to follow. The ant agents interact via stigmergy to discover the network topology. As the proposed ACO converges, it helps us to get all the paths between any two base stations. Genetic Algorithm [3,8] is further applied to get the optimum path over the solution space provided by ACO.

Taking into account the aforementioned facts, a novel routing scheme to compute the optimal route between any two base stations in a CN, taking into account different constraints viz., number of hops, Poisson traffic distribution, buffer capacity, link delay, queuing delay, and bandwidth available is proposed in this paper using ACO-GA. The remainder of the paper is organized as follows: Section 2 discusses the proposed approach, Section 3 about the models considered, Section 4 the performance analysis and Section 5 concludes the paper.

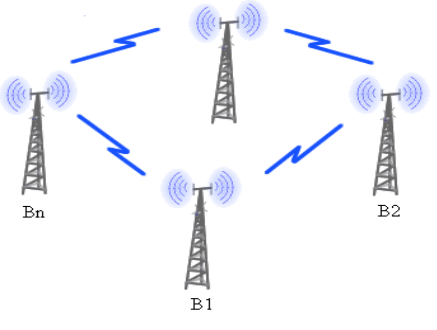


Fig. 1. Communication Network with Base Stations

2 Proposed Approach

The proposed approach considers a CN with the Base Stations (BS) numbered from B1 to Bn. Each BS maintains two data structures namely Base List and Route List as shown in Tables 1 and 2. Base list stores the list of BS in a route.

Route list stores the pheromone strength in each route. Both the data structures are initially empty for all BS. Either one of the base stations in the CN or a node external to the CN functions as a coordinator. The coordinator is responsible for distributing the ants and to configure the network topology after receiving response from the different BS. The proposed approach uses ACO to discover the network topology. In ACO, Ants are *agents* which move along by choosing the path between BS in a CN based on pheromone strength. An ant's path represents a specific candidate solution. When an ant has found a solution, pheromone is laid on its path. The pheromone strength is based on various factors such as bandwidth available, length of path etc. The amount of pheromone associated with each path is reduced as the time goes on. Thus the pheromone strength associated with a path keeps reducing if the path is not used frequently. This affects behaviour of other ants by 'Stigmergy'. As the pheromone strength increases on a path, it indicates that a large number of ants passed through this. This directly implies that the corresponding path is the best path between the respective BS in a CN.

Table 1. Baselist

Base Station	Pointer to Route List

Table 2. RouteList

Routes	Pheromone Strength

2.1 Proposed Algorithm for ACO

Any BS first receives an INITIALIZE message from either the coordinator or another BS. Then it sends Ant Agents to all outgoing links from this BS. The Ant Agents have the name of the initiating BS stored in them to identify the ant agents from different BSs. Once a BS receives an Ant Agent from another, it finds out the initiator of the agent. If the initiator is not present in the BASELIST, it is added. Then the BS executes the following steps:

Step 1: It forwards the Ant Agent received to its outgoing links, if this is the first time the BS receives an agent from the corresponding initiator.
Step 2: If it has not yet started sending Ant Agents of its own, it initiates own Ant Agents.
Step 3: When an Ant Agent discovers a base station, a backward thread is started that reports back to the initiating base station.

Step 4: The backward thread contains information regarding the route that was saved earlier.
Step 5: If the new route discovered corresponds to a known base station, a new entry is made in the corresponding RouteList.
Step 6: If the route discovered corresponds to an unknown base station, a new entry is made in the BaseList first. Then the RouteList is created and updated.

Now as an Ant Agent reaches a particular base station, information is reported back to the initiating base station using the information the ant would have stored along the way. The initiating base station then modifies the BaseList and the RouteList as per the latest information. Since Ants are propagated between all the base stations, all the possible routes are found. This helps us to find the topology. The ants also report back the pheromone strength associated with each route. The pheromone strengths allow us to find the best routes between the stations.

2.2 Proposed GA Based Approach

The Genetic Algorithm takes as input the solution set provided by the ACO. A fitness function based on traffic, number of requests etc. is used to evaluate each individual solution. The best solutions among them are considered for crossover. The process is repeated over and over. Repeated optimization at each step leads to the best solution. This gives the best path between different nodes taking into account the different constraints.

3 Network Models

The proposed approach is applied to different network models, which differ in their topology. The network topology is the physical arrangement of the elements of a network. To provide better service, it is required to find the best path to route among a set of routes taking into consideration the network traffic on the route and the length of the path. Hence the topology directly affects the routing decision and it is essential to configure the network topology faster. The different topologies considered in this paper are classified based on their degree of connectivity as loosely connected, normal connected and tightly connected.

3.1 Loosely Connected Graph

In a loosely connected graph, the number of paths between any two nodes is less as shown in Fig.2. A Tree network is an example of loosely connected graph. Here there is a single path between any two nodes. Hence it takes relatively very less time to discover the topology and the best route.

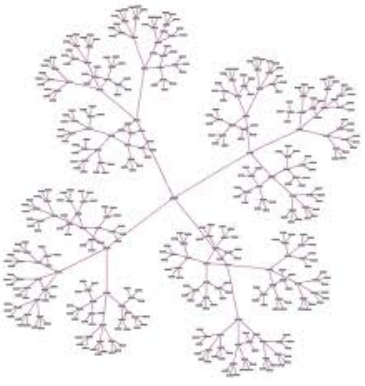


Fig. 2. Loosely Connected Graph

3.2 Normal Connected Graph

In a normal Connected Graph, the number of connections associated with the base stations is not greater than n/2. Fig.3 is an example of a normal connected graph. Here there are more paths to discover between any two communicating nodes. Hence it takes relatively longer time to discover the topology and the best route.

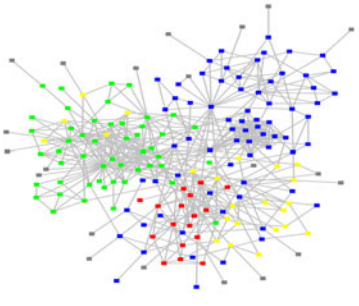


Fig. 3. Normal Connected Graph

3.3 Tightly Connected Graph

In a fully connected graph, the number of connections associated with the base stations is n-1as shown in Fig.4. There exists a single link between every node in the network. Hence it takes maximum amount of time to discover the complete topology and the best route.

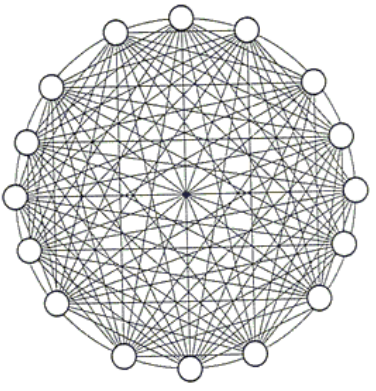


Fig. 4. Tightly Connected Graph

3.4 Pheromone Strength Calculation Using GA

The pheromone strength calculation to find out the best route is based on a number of parameters. The first parameter is the number of hops in the specified route.

$$F1 = N_h = m;\ 1 <= m <= N_b \tag{1}$$

N_b is the number of base stations.

The second parameter is the intensity of traffic on that particular route. The traffic distribution is assumed to be Poisson. The probability distribution is given by the equation

$$P(X = k) = (\lambda^k e^{-\lambda}) / \ k! \tag{1.1}$$

Where the mean arrival rate is given by λ.

Thus the probability of k requests arriving in a time interval can be given by

$$F2 = P(Tk) = (\lambda T)^k * e^{-\lambda T} / \ k! \tag{2}$$

The third parameter is the minimum buffer capacity along that route. It is better to go for routes that have better buffering capacity to avoid congestion. The minimum buffering capacity along the routes is given by

$$F3 = \text{Min}(B_i * C) \tag{3}$$

B_i is the number of buffers at the i^{th} base station.
C is the capacity of each buffer.

The fourth parameter is the delay in message transmission owing to the path length. Assume Erlang k distribution. The probability of delay over a path P of length m less than t is given by the expression

$$F4 = (\gamma^m t^{m-1} c^{-\gamma t}) / (m-1)! \qquad 1 <= m <= B_n - 1 \tag{4}$$

Where γ is the call arrival rate.

The fifth parameter is the queuing delay along the chosen route is given by

$$F5 = \Sigma(\delta_i * t) \tag{5}$$

δ_i is number of free buffers and t is the time taken to load one free buffer.

The sixth parameter is the residual bandwidth along a route as it affects the optimality of the route. The residual bandwidth is given by the formula

$$F6 = \text{Min } (BWi) \tag{6}$$

Where BWi is the bandwidth available at each hop in the route.

From the equations (1) to (6), the Fitness function, a minimization function to compute the best route between BS is coined as

$$F = F1 + F2 + F3 + F4 + F5 + 1/F6 \tag{7}$$

The value of F is calculated for every route between BSs. The route for which the value of F is minimum, is considered to be the best route between the BS.

4 Simulation and Performance Analysis

A set of N base stations of unknown topology is considered for our simulation. ACO Algorithm 1 is applied to this topology. The simulation is done using the Eclipse editor in Java. Eclipse permits to simulate the presence of multiple base stations. It is observed from the simulation that two main factors affect the performance of ACO. They are

- The number of base stations.
- The degree of connectivity.

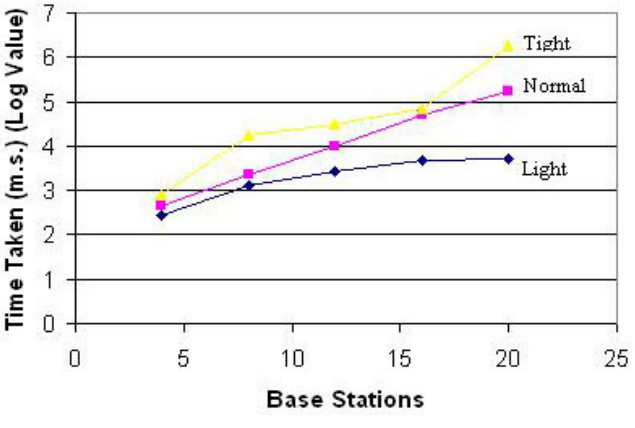


Fig. 5. Time taken for Networks with Different Levels of Connectivity

The performance of the algorithm under various conditions has been studied and results are plotted in the form of graphs shown in Fig.5 and Fig.6.

CASE 1

Consider a loosely connected computer network as shown in Fig 2. In this, each base station is connected to one or two other base stations only. The number of paths between any two base stations is very less. Algorithm 1 is applied to the loosely connected network consisting of 5,10,15,20 & 25 nodes and the time needed to discover the network topology is found out. A graph is plotted with these values as shown in Fig 5. In a lightly connected network, the probability of having a cycle is very small. So, the number of routes between any two base stations is very less. Thus the time taken for topology discovery gradually saturates after 15 base stations.

CASE 2

Consider the base stations are well connected as shown in Fig 3. Each pair of base stations has more than a few paths between them. Hence ACO takes longer time to discover the topology. In a well-connected graph, the number of cycles increases. Thus, the number of routes that have to be discovered increases rapidly. Thus, in normal connectivity, the time taken increases logarithmically.

CASE 3

Here each base station is connected to almost every other base station as shown in Fig 4. In this network, a number of cycles exist in the graph. As the number of nodes increase, the number of routes between any two nodes also increases exponentially with time. This situation is highly improbable in practical situations.

It is observed from the simulation that as the number of nodes and paths increase, Ant Colony Optimization takes more time to converge. But the time taken to discover the topology using this algorithm is considerably lesser than the traditional algorithms.

CASE 4: COMPARISON OF VARIOUS NETWORKS MODELS

Ant Colony Optimization is also applied to various standard network models such as Star, ring, tree network etc. The result of simulation is plotted in Fig 6.

From the graph it's clear that the time taken to find all the paths in a network doesn't increase too much as the number of nodes increases. Thus the ant colony optimization scales well when applied to above network models. It is observed from Fig. 6 that the ACO Algorithm 1 takes less time for standard network models. This is because, once we specify a network model, there is a restriction on how the model should behave. For example, if we consider a ring network, there is a restriction that each BS should be connected to only 2 other BSs. This restriction reduces the number of routes possible between base stations. Consider the case of a tree network wherein there is a restriction that it can never have a cycle. Thus the number of routes between any two BSs is always one. Thus time taken for ACO is minimum. Traditional algorithms such as Table driven and On-Demand Routing algorithm fail to discover the network topology before routing. Ant Colony Optimization is useful in situations where the connectivity of the network is not

known in advance but routing is to be done. Though all the existing paths between all base stations in a computer are found out by Ant Colony Optimization, it is up to the user to limit the number of paths to be discovered between any pair of nodes as it increases the processing overhead.

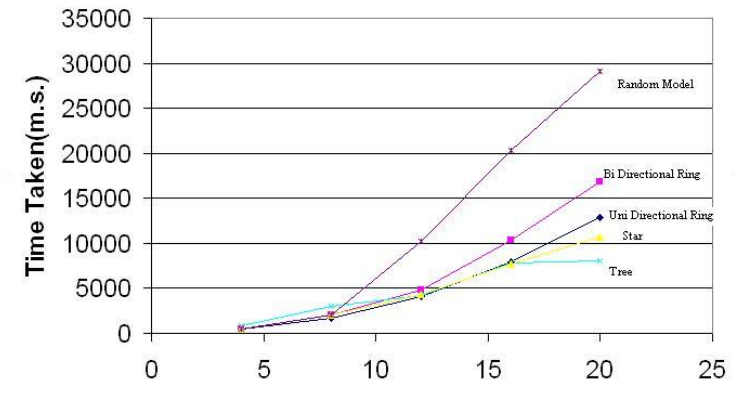


Fig. 6. Comparison of Various Network Models

But in case of general network models as in Fig 2 and Fig. 3, the connectivity is random and so the time taken by ACO increases. To reduce the processing time, the results obtained by ACO are given as input to GA based optimization technique to get the optimal routes between base stations faster. The results of the simulation are shown in the graphs in Figs. 7 to 9.

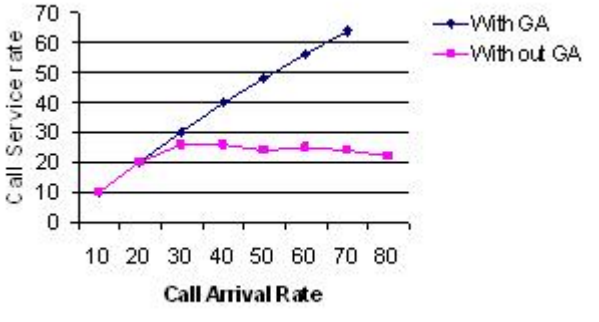


Fig. 7. CAR Vs CSR

The selection of the best path available from the set of existing solutions using the iterative approach, where each path is tested for optimality using the fitness function, is time consuming. Moreover, there is a restriction on the number of requests that can be serviced. Once the threshold is reached, the probability that future requests are dropped is very high as shown in the Fig 7. Hence to this

solution set, another optimization using GA which can deal efficiently with a huge solution set is applied. The operations in GA viz., selection, crossover, mutation take less time compared to iterative approaches. Thus for a given Call Arrival Rate (CAR), the Call Service Rate (CSR) is high in case of GA.

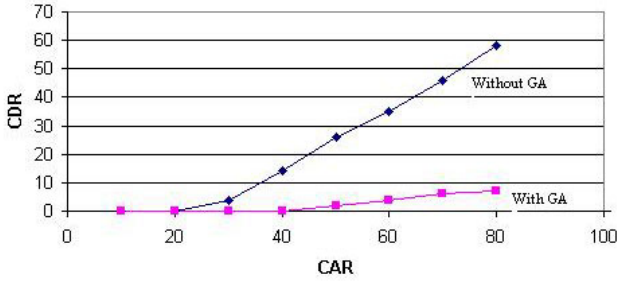


Fig. 8. Comparison Between GA and Normal Call Dropping Rates

Further, it can be observed from Fig 8. that using GA, only a few calls are dropped. This is because the time taken for selection of an optimal route using GA is very less. Hence, a large number of requests can be serviced without the problem of call drop. Thus, it is seen that GA aids ACO and leads to a better performance. Moreover, our proposed GA based approach aids to speed up the optimal route selection process when the base stations are tightly connected. The comparison between the time taken by GA and the iterative selection procedure is given in Fig. 9.

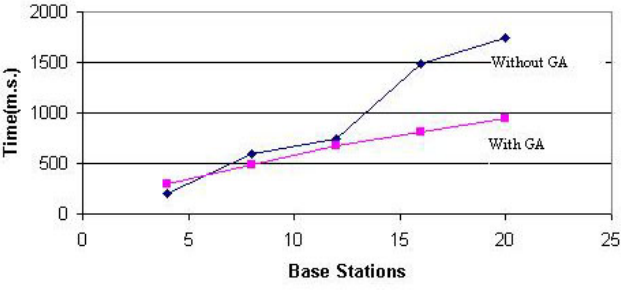


Fig. 9. Comparison Between GA and Iterative Route Selection

From the graph in Fig. 9, it can be inferred that the GA based approach takes less time to converge compared to the approach without GA. Thus, when the number of base stations increase, it is better to use the GA based approach to select the optimal solution from the solution pool.

5 Conclusion

Ant Colony Optimization algorithm proposed in this paper outperforms the existing algorithms viz., Table driven and on demand routing algorithms as it discovers the topology before routing. To find out the optimal routes between base stations faster, GA based optimization is further applied to the results given by ACO. Simulation results show that this combined ACO and GA based approach performs better than the approach with ACO. The time taken by our proposed approach to discover the topology depends on the connectivity of the CN and it can be significantly reduced either by limiting the number of paths to be discovered between the BSs or by using multiple ownership for each ant agent.

References

[1] Perkins, C.E.: Mobile networking in the Internet. Mobile Networks and Applications 3, 319–334 (1999)

[2] Schiller, J.: Mobile Communications, 2nd edn. Pearson Education (2003)

[3] Mala, C., Shriram, R., Agarwal, S.: Genetic Algorithm based Routing in Multi-cast Communication in Networks with Mobile Users. In: The Proceedings of International Conference on Information Technology (Asian Applied Computing Conference), Nepal (2005)

[4] Stutzle, T., Dorigo, M.: ACO Algorithms for the Traveling Salesman Problem. In: Miettinen, K., Makela, M., Neittaanmaki, P., Periaux, J. (eds.) Evolutionary Algorithms in Engineering and Computer Science. Wiley (1999)

[5] Ho, S.L., Yang, S., Ni, G., Machado, J.M.: A Modified Ant Colony Optimization Algorithm Modeled on Tabu- Search methods. IEEE Transactions on Magnetics 42(4) (2006)

[6] Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Efficient Graph Search by a Smell-Oriented Vertex Process. Annuals of Mathematics and Artificial Intelligence 24, 211–223 (1998)

[7] Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: ANTS: Agents, Networks, Trees, and Subgraphs. IBM Haifa Research Lab, Future Generation Computer Systems Journal 16(8), 915–926 (2000)

[8] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Pearson Education (2004)

[9] Ma, L., Tian, J., Yu, W.: Visual saliency detection in image using ant colony optimisation and local phase coherence. Electronics Letters 46(15), 1066–1068 (2010)

[10] Cecilia, J.M., Garcia, J.M., Ujaldon, M., Nisbet, A., Amos, M.: Parallelization Strategies for Ant Colony Optimisation on GPUs. In: IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp. 339–346 (2011)

Optimized Channel Allocation Using Genetic Algorithm and Artificial Neural Networks

Narendran Rajagopalan, C. Mala, M. Sridevi, and R. Hari Prasath

Department of Computer Science and Engineering, National Institute of Technology, Trichy, 620015
{narenraj1,hariprasathnitt}@gmail.com,
{mala,msridevi}@nitt.edu

Abstract. As the spectrum for wireless transmission gets crowded due to the increase in the users and applications, the efficient use of the spectrum is a major challenge in today's world. A major affecting factor is the inefficient usage of the frequency bands. Interference in the neighboring cells affects the reuse of the frequency bands. Some of the quality of service parameters such as residual bandwidth, number of users, duration of calls, frequency of calls and priority are considered for optimized channel allocation. Genetic Algorithm and Artificial Neural Networks are applied to determine the optimal channel allocation considering the quality of service parameters. The simulation results show that using Genetic algorithm betters heuristic method and artificial neural networks performs better than Genetic Algorithm by a comfortable margin.

Keywords: Genetic Algorithm, Channel Allocation, Quality of Service, Artificial Neural Network.

1 Introduction

Channel allocation is critical in today's world due to the limited spectrum available. This is mainly due to the contribution of the increasing cost and the spectrum that is left to be allotted to the operators. Efficient use of the spectrum thus becomes a necessity. Channels can also be utilized more efficiently by reusing it when two stations do not interfere. Hence the operators can utilize the spectrum efficiently by dynamically allocating frequencies to their cells as and when required.

Dynamic channel allocation[1,2] is more efficient as compared to fixed channel allocation under low traffic conditions. But when there is a constant stream of calls, fixed channel allocation is much more efficient in not wasting the spectrum that is available. Optimization is not really required in such cases but to hope there will be a constant stream of calls every day is highly unlikely and hence there is a need to use dynamic channel allocation. While using dynamic channel allocation,

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 645–655.
springerlink.com

the channel is allocated depending on the requirements of the users demands in real time. But this arises the problem of how to manage the channel allocation efficiently. Genetic Algorithm, Artificial Neural Network are used to optimize the dynamic nature of the channel in this work.

1.1 Introduction to Genetic Algorithms

Genetic Algorithm[3, 9] is based on the theory of natural selection proposed by Charles Darwin. It is a subset of a set of algorithms known as Evolutionary Algorithms which are mostly used to find optimum solutions to problems. Genetic Algorithms are commonly used as optimization algorithms. It uses a combination of survival of the fitness and a randomized information exchange procedure. It is the same procedure that is followed by the theory of natural selection of using the best of the previous generation to create a new generation with a better set and hence optimization or evolution.

Initially, for implementing Genetic Algorithm[3,9], there are two basic requirements – an initial set of values (known as individuals) and a fitness function for evaluating the individual. The processes involved in Genetic Algorithm are simple and basic operations such as copying of strings, swapping strings, etc. Initial population, a set of individuals, maybe generated using a random function or using a set of previously known or assumed data. The value of the individual is determined by the parameters. Some parameters may have a higher priority than others, some may have a wider range of states or variations than others, etc. This is used to decide the number of bits to be allotted to each parameter in an individual. If a higher level of optimization is required, a larger string length for the individual can be used. Increasing the number of individuals in the population may also give a higher level of optimization.

The amount of optimization required usually decides the way in which the initial population is generated.

Once the population is generated, a set of operations is done in succession so as to optimize the population. They are,

(a) Reproduction
(b) Crossover
(c) Mutation

Reproduction is the process where the individuals are copied over to the next generation based on their fitness defined by the fitness function. Individuals with a higher fitness are reproduced while the ones with the least fitness are neglected. This is usually achieved by finding out the percentage contribution of each individual to the fitness of the entire population.

Crossover is the process where information exchange or evolution occurs. After reproduction, a set of random individuals are paired and mated. Mating is when a part of the string from one individual is swapped with its corresponding ones from its mate. The part of the string to be swapped is decided by randomly choosing a point at which the swapping occurs, and this point is called as the crossover point.

When combined with reproduction, this leads to effective optimization by recombination of individuals.

Mutation is process where an individual's string is changed randomly by changing the value of one of the parameters randomly. This is done to ensure that optimization is carried out if by chance all the individuals in the population have 0's or 1's in a particular location which may lead to incomplete optimization. As mutation is a very rare event in the natural selection procedure, the same probability of one in a thousand cycles is implemented.

1.2 Introduction to Artificial Neural Networks

Artificial neural networks (ANN)[4,5,6] have been developed as generalizations of mathematical models of biological nervous systems. A first wave of interest in neural networks (also known as *connectionist models* or *parallel distributed processing*) emerged after the introduction of simplified neurons.

The main objective of Artificial Neural Networks is to understand a new style of computation which is inspired by biological neurons and their adaptive connections. It is a very different style from sequential computation. Applications of neural networks are basically to solve practical problems by developing novel learning algorithms. An ANN consists of inputs and corresponding weights, which when computed by a mathematical function gives the activation of neuron. One more function, which can be identity, will compute the output of neuron. Usually a threshold value is associated with every ANN. The weights associated can be either positive or negative. Weights can be adjusted to get desired output for specific input values. ANN has the ability to learn how to accomplish the tasks based on the data provided for training. An ANN can create its own representation of the information it received during its learning time. There are two types of ANN's, Feed-forward networks and Feedback networks. Feed-forward ANNs allows signals to travel in only one way, from input to output, no sense of time. There are no feedback loops. Feed-forward ANN's directly associate inputs with outputs. They are extensively used in pattern recognition. Feedback networks are designed to have signals travelling in both directions by introducing loops in the network. Feedback networks are dynamic in nature. Their state changes continuously until

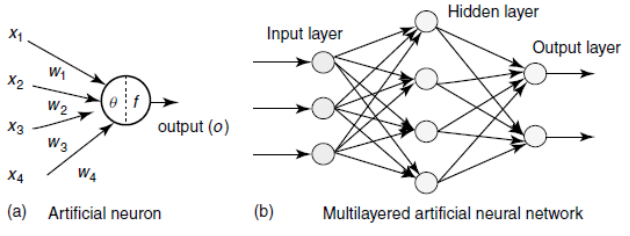


Fig. 1. Architecture of an Artificial neuron and a multilayered neural network

they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. In feed-forward ANNs, the back propagation algorithm is used. The signals are send forward and errors will be propagated backwards. ANNs are organised in layered format. Input layer, output layer and a set of intermediate hidden layers.

It is used in applications such as robotics, diagnosing, forecasting, image processing and pattern recognition. There are two types of ANN's, Feed-forward networks and Feedback networks**.** In this paper feedback network is used for the application.

Feedback method

Feedback networks in Figure. 1(b) are designed to have signals travelling in both directions by introducing loops in the network. Feedback networks are dynamic in nature. Their state changes continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

A Self-Feedback Neural Network (SFNN) [7,8,10] is a recurrent, two-layer network. The output layer of an SFNN contains several self-feedback units. In the feedback layer of this model, there is no interlinking among the units. The output of SFNN contains the past information about the system which is made possible by the self-feedback connection between the units. As there is no interlinking among the units in feedback layer, the SFNN has very few weights when compared to a fully recurrent ANN and the network is very much simplified.

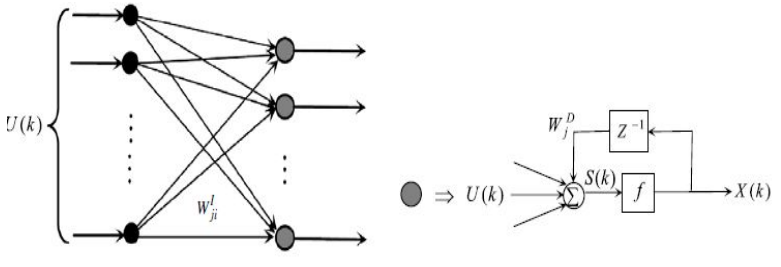


Fig. 2. Structure of SFNN Model

The architecture of the SFNN model is depicted in Fig. 2. The mathematical description is as follows

$$S_j(k) = \sum_{i=1}^{n} W_{ij}^{1} u_i(k) + W_j^{D} X_j(k-1) \tag{1}$$

$$X_j(k) = f\left(S_j(k)\right) \tag{2}$$

Where $u_i(k)$ (i=1,....,n) denotes the external input, and $X_j(k), S_j(k)$ (j=1,.....,m) are the state variable and output of the jth unit of the output layer, respectively. W_{ij}^{1}, W_j^{D} are connection weights from input to output layer and within the output layer, respectively. $f(\lambda)$ is the sigmoid activation function defined as

$$f(\lambda) = 1/(1 + e^{-\lambda}). \quad (3)$$

Feedback neural networks have advantages over feed-forward networks in that they have the ability to store previous state information and prevent the need to predict the model order of the time series.

2 Proposed Scheme

Channel assignment schemes are classified into Fixed Channel Assignment and Dynamic Channel Assignment. In Fixed channel assignment schemes, certain frequencies are assigned statically to cells. In dynamic channel assignment schemes, whenever there is a call request, a channel is selected such that it is optimal and minimizes interference. Experiments have proved that fixed channel assignment schemes perform better in high and uniformly distributed traffic conditions. But in case of non-uniform traffic, where call requests are more in one cell and less in the other, dynamic channel assignment is better suited. The quality of service parameters must be taken into consideration to decide on who should be allotted the channel. Some of the influencing parameters like Periodicity of calls, number of users, residual bandwidth, type of service, type of users, type of calls are considered in this work. For optimal channel allocation, Genetic Algorithm, heuristic method and Artificial Neural Network is applied.

2.1 Quality of Service Parameters Considered in This Work

Duration of Calls(a): It is defined as the amount of time a user holds on to a certain call. This affects the allocation as knowledge of the amount of time the spectra is in use is vital for optimal allocation of the spectra to the users. The range of call times can be obtained from recorded history of usage by users over a period of time to find the average range of time that users hold the channel allocated to them.

Number of Users(b): the number of users requesting allocation of frequency to make a certain voice or data call at a certain time. This is proportional to the number of users active in the given cell. This is also obtained using knowledge of the recent history of traffic in each cell owned by the operator.

Residual Bandwidth(c): Residual bandwidth is the remaining spectra available to be allocated in each cell. The advantage of DCA over FCA takes effect as in FCA each cell is allocated with a given set of frequencies while in DCA the total spectrum can be allocated to any of the cells. The spectrum available to each cell is much higher in DCA in comparison to FCA. Residual Bandwidth or the number of free channels available in a cell is another Quality of Service parameter which can affect the service quality.

Type of Service(g): depending upon whether it is a voice service or data service or multimedia service, this parameter can be used as a quality of service parameter to decide on channel allocation.

Type of Users(h): Type of Users is a parameter differentiating regular users from roaming users. Roaming users must be given higher priority during channel allocation, so it is considered as one of the parameters for decision making.

Type of Calls(i): Type of call is another parameter differentiating emergency calls from ordinary calls. It is one of the parameter which can affect the channel allocation decision.

3 Application of Genetic Algorithm

For the application of genetic algorithm to the problem, the parameters are encoded as eight bit strings. Some of the parameter values are obtained from the history of the caller. The parameter `a' is the time duration, a user has called. Every 10 seconds is assumed to be one unit. Hence a value of a = 25, would imply the duration of the call to be 250 seconds. The parameter 'Type of Service' is higher if the value of g is higher. Hence if g = 100, then it has a higher priority than $g_2 = 65$, but lesser than $g_3 = 150$. Depending upon the density of users in a cell, a unit is selected like 10 users as one unit. Hence a value of b=45 would indicate 450 users. A fitness function is needed and initial population as seed is required. The fitness function is given by,

$$F(x) = a + 1/b + c + 1/g + h + 1/i \qquad (4)$$

Table 1. Randomly Generated Seed Sample

Sl. No	Duration of Call(a)	Number of Users(b)	Residual Bandwidth(c)	Type of Service(g)	Type of Users(h)	Type of Calls(i)	F(x)
1	52	105	67	245	195	60	314.03
2	8	87	253	237	48	25	309.06
3	213	242	162	249	55	37	430.04
4	117	234	53	164	117	200	287.02
5	115	109	146	196	232	139	493.02
6	210	108	88	192	250	39	548.04
7	95	110	184	82	77	245	356.03
8	202	57	125	154	70	52	397.04
9	40	188	153	224	23	76	216.03
10	111	185	159	118	25	172	295.02
11	128	81	147	97	177	209	452.03
12	34	204	202	32	121	102	357.05
13	101	150	109	130	164	42	374.04
14	69	221	165	157	9	253	243.01
15	177	206	150	38	127	252	454.04
16	180	214	28	218	87	193	295.01
17	62	34	63	212	65	126	190.04
18	196	101	47	61	134	180	377.03
19	206	166	243	67	158	118	607.03
20	161	218	119	174	219	90	499.02

The fitness function is a maximization function. The initial values for the strings are randomly generated and tabulated as shown in table below. Each parameter is represented as string of eight bits each.

4 Application of Artificial Neural Network

Feedback Neural Network[4, 7] is applied to the get the optimized solution. MATLAB was used for testing. The input parameters and assigned weights are processed to obtain the activation value. The feedback network uses bidirections. So, the calculation of Xj has to be performed for bidirection by make use of Eq. (1) & (2). The obtained output has to be learned using random supervised competitive learning by using the equations mentioned below:

$$W_{ij}(t) = S_i [S_j - W_{ij}(t)] \quad \text{(for deterministic competitive learning)} \tag{5}$$

$$W_{ij}(t) = r_i (a) S_i [S_j - W_{ij}(t)] + n_{ij}(t) \quad \text{(for stochastic competitive learning)} \tag{6}$$

$$S_i = f_i (x_i(t)) \text{ (output signal of the unit i)} \tag{7}$$

$$S_j = f_j (x_j(t)) \text{ (output signal of the unit j)} \tag{8}$$

Where $n_{ij}(t)$ is zero mean Gaussian white noise process & $r_i (a)$ is reinforcement function.

The parameters or data considered for the problem is given below:

Input parameter: (a, b, c, g, h, i) are taken as random value generated in between 1 to 255.

Table 2. Sample Input to the Neural Network

Sl.No	Input Data for the Neural Network						Activation Value
	A	b	C	g	h	i	
1.	70	62	9	34	133	27	22.12
2.	121	195	203	90	226	62	75.31
3.	20	155	167	182	178	55	53.206
4.	78	203	75	185	184	53	41.206
5.	138	136	37	160	244	186	45.604
6.	189	161	76	214	108	158	44.903
7.	56	241	58	70	102	51	27.409
8.	107	147	82	16	151	207	42.22
9.	36	41	190	72	146	149	56.208
10.	136	96	217	125	66	161	63.605

5 Performance Analysis with Heuristic Method, Genetic Algorithm and Neural Networks

MATLAB was used for conducting performance analysis. For each case study, any two parameters are constantly increased at a predetermined interval and the

performance of Heuristic method, Genetic Algorithm and Artificial Neural Networks is recorded varying other parameters. Throughput in the figures is the percentage of best theoretical case to the throughput achieved by different methods. The graphs in Figure 3, Figure 4, Figure 5, Figure 6 and Figure 7 show the performance of Heuristic Method, Genetic Algorithm and Artificial Neural Network. It is clear that Artificial Neural Network perform better than Genetic Algorithm and Heuristic Method.

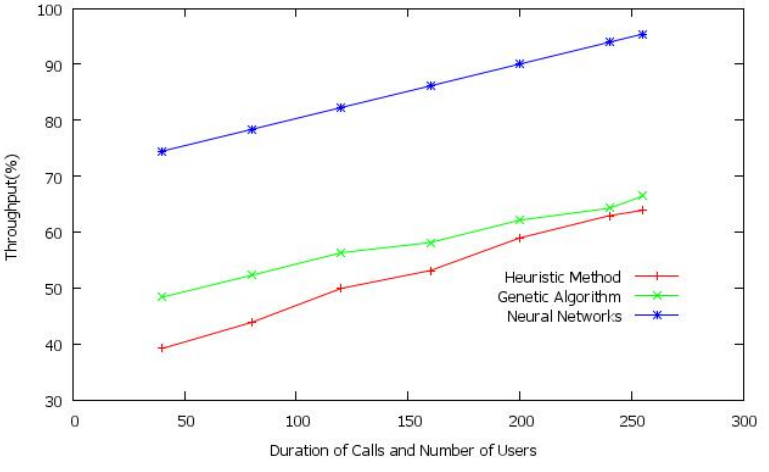


Fig. 3. Performance of Heuristic Method, Neural Networks, Genetic Algorithm with Parameters Duration of Calls and Number of Users fixed at different Intervals

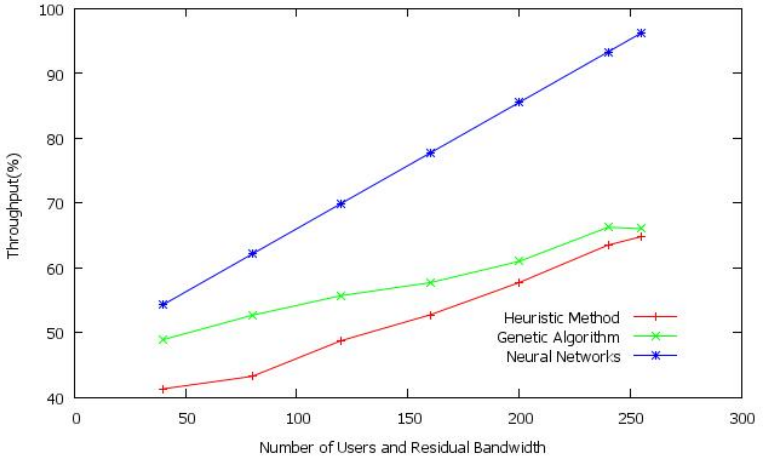


Fig. 4. Performance of Heuristic Method, Neural Networks, Genetic Algorithm with Parameters Number of Users and Residual Bandwidth fixed at different Intervals

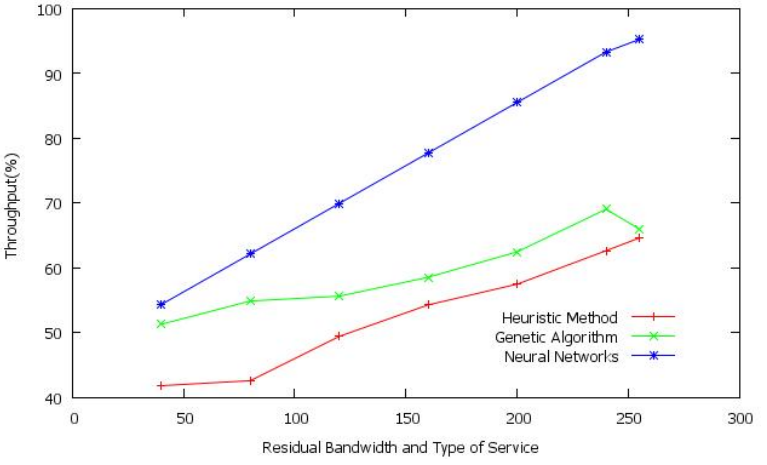


Fig. 5. Performance of Heuristic Method, Neural Networks, Genetic Algorithm with Parameters Residual Bandwidth and Type of Service fixed at different Intervals

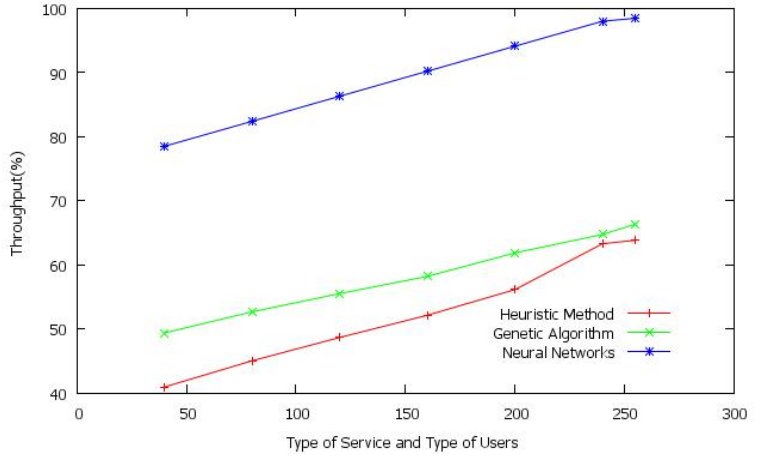


Fig. 6. Performance of Heuristic Method, Neural Networks, Genetic Algorithm with Parameters Type of Service and Type of Users fixed at different Intervals

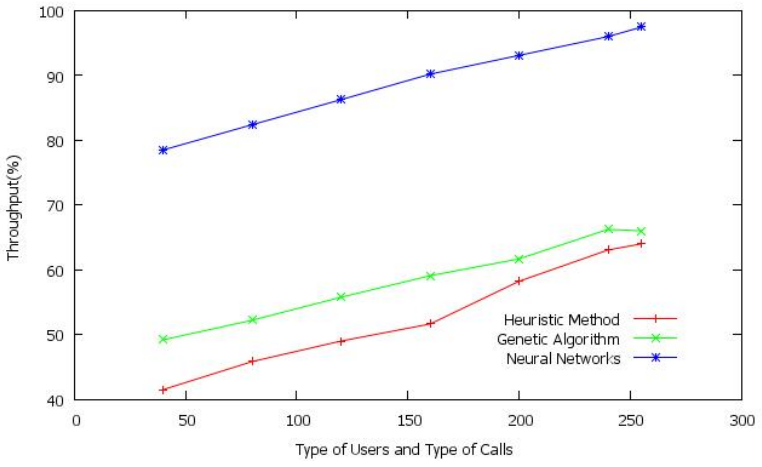


Fig. 7. Performance of Heuristic Method, Neural Networks, Genetic Algorithm with Parameters Time of Users and Type of Calls at different Intervals

6 Conclusion

Parameters such as Duration of Calls, Number of Users, Residual Bandwidth, Type of Users, Type of Service and Type of Calls are considered for Channel allocation. Optimization of these parameters is carried out using Genetic Algorithm and Artificial Neural Networks. It can be observed from the results obtained that throughput of Artificial Neural Network is better than throughput of Genetic Algorithm and Heuristic method.

References

[1] Granelli, F., Przemyslaw, P., Venkatesha, P., Hoffmeyer, J.: Standardization and Research in Cognitive and Dynamic Spectrum Acess Networks: IEEE SCC41 efforts and Other Activities. IEEE Communications Magazine 48(1), 71–79 (2010)

[2] Martinez, D., Andrade, A.G., Martinez, A.: Interference-Aware Dynamic Channel allocation scheme for cellular networks, pp. 295–300. IEEE Press, Los Alamitos (2010)

[3] Goldberg, D.E.: Algorithms in Search, Optimization and Machine Learning. Pearson Education, India (2004)

[4] Yegnanarayana, B.: Artificial neural networks. Prentice Hall of India (2001)

[5] Vidyarthi, G., Ngom, A., Stojmenovic, I.: A hybrid channel assignment approach using an efficient evolutionary strategy in wireless mobile networks. IEEE Transactions on Vehicular Technology 54(5), 1887–1895 (2005)

[6] Elhachmi, J., Guennoun, Z.: Evolutionary neural networks algorithm for the dynamic frequency assignment problem. International Journal of Computer Science and Information Technology 3(3), 49–61 (2011)

[7] Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro–fuzzy and soft computing–A computational approach to learning and machine intelligence. PHI Learning (2010)
[8] Wikipedia, http://www.wikipedia.org/
[9] Pandian, J., Murugiah, P., Rajagopalan, N., Mala, C.: Optimization of Dynamic Channel Allocation Scheme for Cellular Networks Using Genetic Algorithim. In: Nagamalai, D., Renault, E., Dhanushkodi, M. (eds.) PDCTA 2011. CCIS, vol. 203, pp. 628–637. Springer, Heidelberg (2011)
[10] Kaabi, F., Ghannay, S., Filali, F.: Channel allocation and routing in Wireless Mesh Networks:A Survey and qualitative comparison between schemes. International Journal of Wireless and Mobile Network 2(1), 132–151 (2010)

Technical and Relative Efficiency Assessment of Some Private Sector Hospitals in India

Sandeep Kumar Mogha[1], Shiv Prasad Yadav[1], and S.P. Singh[2]

[1] Department of Mathematics, Indian Institute of Technology, Roorkee, Roorkee-247667, India
moghadma@gmail.com
[2] Department of Humanites and Social Sciences, Indian Institute of Technology, Roorkee, Roorkee-247667, India
{yadavfma,singhfhs}@iitr.ernet.in

Abstract. This paper measures technical and relative efficiencies of some private sector hospitals in India. The study makes an attempt to provide an overview of the general status of the hospitals. Data Envelopment Analysis (DEA)-CCR and BCC models have been applied for the Input-output data collected from 15 hospitals for the year 2009-2010. The study found that out of 15 hospitals, 2 hospitals are technically efficient and 8 hospitals are pure technically efficient. Assessment of technical efficiency concludes that the performance of hospitals is good but still very far from the optimal level. On average, a technical inefficient hospital may reach on the efficiency frontier if it is able to increase its output by 30.20%. The mean value of OTE (60.80%) indicates that on average a hospital to be efficient has to produce 39.20% more output with the same level of inputs.

Keywords: DEA, Efficiency and Hospitals.

1 Introduction

Over the last 5 decades India has built a sound healthcare infrastructure at primary, secondary and tertiary levels in both the public and the private sectors. However, during the last decade, role of the public sector in the healthcare services has been shrinking and that of private sector has been expanding. Inadequate infrastructure, non-availability of competent doctors and paramedical staff, poor maintenance, lack of motivation among the staff, and insufficient supply of drugs and medicines are the major problems with the public sector healthcare institutions. The policy of structural adjustment programs and liberalization being initiated by the Government of India since 1991 has also affected the public sector health services, which is evident from the fact that public sector expenditure on health in India has decelerated from 1.3% of GDP in 1990 to 0.9% in 1999 [9]. In 2001, the target was to increase the spending from 0.9 % to at least 2 % of the GDP on healthcare.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 657–666.
springerlink.com

Private sector healthcare services range from those provided by large corporate hospitals, small hospitals/nursing homes to clinics and dispensaries. The growth of this sector has been triggered by a number of factors including government policies of concessional land allotment and relaxation in custom duties on import of medical equipment, rapid influx of medical technology, growing deficits of public sector hospitals, and an increasing middle - income class. Private healthcare institutions can transform India into a major medical tourism destination. Big corporate hospitals in the country, which constitute about 1% of total healthcare services, have already started treating patients from other countries. However, in order to become globally competitive and also to meet out the growing domestic demand for healthcare services, these hospitals have to upgrade their infrastructure and adopt cost-effective medical technology to provide services of international standard at affordable prices.

Initially DEA was proposed by Charnes, Cooper and Rhodes [3]. Since early 1980s, DEA has been extensively used for efficiency analysis of health care organizations. Work done by Sherman [8] and Nunamker [6] were among the first that applied DEA to measure hospitals and nursing services efficiency. Since then, DEA has been used widely in the assessment of health care organizations all over the world. Despite the availability of other efficiency measurement methods, the DEA has become the dominant approach for measuring efficiency in healthcare and many other sectors in the economy [4]. DEA determines relative efficiencies of the DMUs on the basis of their inputs/outputs. Keeping this in view, we plan to examine the relative performance of some hospitals of India through DEA methodology. The paper attempts to estimate technical efficiency of the hospitals, set benchmark for inefficient hospitals, and suggest alternative actions that would make them relatively efficient. The paper is organized as follows: section 2 contains data and variables, section 3 models used and results and discussion are given in section 4, followed by conclusion in the last.

2 Data and Variables

In India, PROWESS and CAPITALINE databases include data on a large number of companies, including hospitals. These sources have balance-sheet based financial data. The firm level panel-data is only available with PROWESS and CAPITALINE databases. Both the databases are similar one and based on annual reports. Earlier studies have also used the PROWESS and CAPITALINE data to estimate the productivity and efficiency of hospitals. Keeping all these limitations in view, we have obtained the data from CAPITALINE database. This study covers 25 private sector hospitals of India for the sample. The data upon which our study is based is collected for the year 2009-2010.The variables used in the study represent financial status of the hospitals. Attempts are made to incorporate a fairly comprehensive list if inputs and outputs, which reflect the general and informative results. However some consideration have been specified for the selection of number of DMUs. As the thumb rule the number of DMUs is expected to be larger than the product of number of inputs and outputs in order to discriminate effectively between efficient and inefficient DMUs.

2.1 Variables Selection

Initially, we have considered 1 output "Net Profit" and four inputs NFC, EE, SW, and Raw Material. But Net Profit figures for some of the hospitals are found to be negative, and DEA strictly requires value of all inputs and outputs to be positive. Hence, it becomes imperative to drop Net Profit variable from the analysis. Most of the hospitals reported raw material as zero, therefore this variable is not considered for DEA analysis. This variable is replaced by Capital Employed. Finally, in the calculation of efficiency measure, the input variables are NFC, EE, SW, and CA, and the output variable is OI. The observed data for the selected hospitals are given in Table1, all inputs and outputs are measured in crores of rupees.

Table 1. Observed data of the sample Hospitals in India (2009-2010)

		Inputs				Output
S.No.	Hospital Name	NFC	EE	SW	CA	OI
H1	A D S Diagnostic Ltd (New Delhi)	1.62	0.04	0.46	3.86	3.32
H2	Apollo Health & Lifestyle Ltd (Chennai)	2.85	0.04	2.63	21.01	6.41
H3	Alps Hospital Ltd (New Delhi)	39.95	1.39	8.48	24.22	38.86
H4	Meenakshi Multi -speciality Hospital Ltd (Chennai)	11.53	0.54	3.11	8.26	10.45
H5	Dhanvantri Jeevan Rekha Ltd (Meerut)	4.42	0.12	0.89	4.9	3.48
H6	Dolphin Medical Services Ltd (Hyderabad)	12.15	0.04	0.25	24.65	3.35
H7	HSCC (India) Ltd (New Delhi)	6.32	0.29	12.97	72.39	21.38
H8	KMC Speciality Hospitals India Ltd (Tiruchirappalli)	10.42	0.32	1.97	16.64	8.54
H9	Lotus Eye Care Hospital Ltd (Coimbatore)	42.65	0.36	1.96	54.19	14.17
H10	Medinova Diagnostic Services Ltd (Hyderabad)	5.76	0.65	2.44	15.34	12.13
H11	N G Industries Ltd (Kolkata)	4.62	0.28	0.46	11.71	10.11
H12	Prerana Hospital Ltd (Kolhapur)	7.8	0.17	2.51	16.35	8.4
H13	Secunderabad Healthcare Ltd (Hyderabad)	5.02	0.17	0.80	7.65	18.43
H14	Sharma East India Hosp & Medical Res. Ltd (Jaipur)	8.34	0.39	2.11	13.1	8.81
H15	Regency Hospital Ltd (Kanpur)	32.3	1.30	5.7	44.64	39.72
Mean		13.05	0.41	3.12	22.59	13.84

The inputs and outputs variables are defined as follow:

Net Fixed Capital (NFC): Net fixed capital is that portion of the total capital invested in fixed assets.

Energy Expenses (EE): These expenses are mainly electric charges, and also include fuel charges, if any.

Salary & Wages (SW): It includes total annual expenses incurred by a hospital on all employees with management. Contributions to employee's provident fund, payment of bonus and staff welfare expenses are also included under this variable.

Current Assets (CA): The value of assets that are reasonably expected to be converted into cash within one year. Current assets include cash, accounts receivable, inventory.

Operating Income (OI): It refers to the revenue generated by a hospital from its main service activities during a given accounting period.

Correlation and regression analysis have been worked out to know the extent of variation in OI caused by the variation in the mentioned input variables. The results given in Table 2 shows that OI has good correlation with these input variables.

Table 2. Correlation Matrix of Input-Output variables

	NFC	EE	SW	CA	OI
NFC	1				
EE	0.705	1			
SW	0.314	0.502	1		
CA	0.491	0.252	0.723	1	
OI	0.686	0.89	0.665	0.477	1

3 Models Used

The DEA models are divided into two orientations, one is input orientation and another is output orientation. The input orientated model determines by how much inputs quantities can be proportionally reduced without changing the outputs' quantities produced. In contrast, the output-orientated model determines by how much outputs' quantities can be proportionally expanded without altering the quantities of inputs. The choice of model orientation, input-oriented or

output-oriented, depends on the extent to which the health institution has control on its inputs or outputs (Agarwal *et al.* [1] & Ismile [5]) also used output-orientation. In this study, an output orientation has also been used for the following reasons:

i. Because of the data is financial, and in private sector the purpose is to maximize its annual budget so output-orientation is used.
ii. The availability of health resources is restricted by the annual budget allocated for the health sector.

Output Oriented CCR (envelopment) model [2]	Output Oriented BCC (envelopment) model [7]
$Max\, Z_k = \phi_k$ $+\varepsilon\left(\sum_{r=1}^{s} s_{rk}^{+} + \sum_{i=1}^{m} s_{ik}^{-}\right)$ Subject to: $\sum_{j=1}^{n} \lambda_{jk} y_{rj} - s_{rk}^{+} = \phi_k y_{rk} \quad \forall r = 1,..s,$ $\sum_{j=1}^{n} \lambda_{jk} x_{ij} + s_{ik}^{-} = x_{ik} \quad \forall i = 1,..m,$ $\sum_{j=1}^{n} \lambda_{jk} \geq 0 \quad \forall j = 1,2,..,n,$ ϕ_k is unrestricted is sign $s_{ik}^{-}, s_{rk}^{+} \geq 0 \quad \forall i = 1,2,..m,$ $r = 1,2,..s,$	$Max\, Z_k = \phi_k$ $+\varepsilon\left(\sum_{r=1}^{s} s_{rk}^{+} + \sum_{i=1}^{m} s_{ik}^{-}\right)$ Subject to: $\sum_{j=1}^{n} \lambda_{jk} x_{ij} + s_{ik}^{-} = x_{ik} \quad \forall i = 1,..m,$ $\sum_{j=1}^{n} \lambda_{jk} y_{rj} - s_{rk}^{+} = \phi_k y_{rk} \quad \forall r = 1,..s,$ $\sum_{j=1}^{n} \lambda_{jk} = 1 \quad \forall j = 1,2,..,n,$ ϕ_k is unrestricted in sign, and $\lambda_{jk}, s_{ik}^{-}, s_{rk}^{+} \geq 0 \quad \forall i, j, r,$
$Max\, Z_k =$ Output Oriented CCR and BCC efficiency of k^{th} DMU.	

where, n is the number of DMUs (i=1,…,n), p is the number of inputs x_{ij} (j=1,2…..,p), s is the number of outputs y_{rj} (r=1,2…….s), ϕ_k is the output-oriented efficiency score, λ_{jk} (j=1,2…n) are non-negative reals, and s^- & s^+ represent input excesses and output shortfalls respectively. A hospital is said to be fully efficient if and only if ϕ^* =1 and slacks $s_j^{-*} = s_r^{+*} = 0$, $\forall j, r;$. Otherwise if $\phi^* > 1$, the hospital is called inefficient.

3.1 Input-Output Targets for Inefficient Hospitals

DEA allows to set the input and output targets for inefficient hospitals [11, 12], so that they improve their performance, and each of the inefficient hospital become

efficient. According to the model the targets for the inefficient hospitals are as follows:

For outputs $\overline{y_{rk}} = \theta_k^* y_{rk} + S_{rk}^{+*} = \sum_{j=1}^{n} \lambda_{jk}^* y_{rj}$	For inputs: $\overline{x_{rk}} = x_{ik} - S_{ik}^{-*} = \sum_{j=1}^{n} \lambda_{jk}^* x_{ij}$

where $\overline{y_{rk}}$ (r= 1) and $\overline{x_{ik}}$ (i= 1,2,3,4) are the output and input targets respectively for the k[th] hospital; y_{rk} and x_{ik} are the actual output and inputs respectively of the k[th] hospital; θ_k^* is the optimal efficiency score of the k[th] hospital; s_{ik}^{-*} and s_{rk}^{+*} are the optimal input and output slacks of the k[th] hospital for (i=1,2,3,4) and (j=1). The optimal input and output targets for inefficient hospitals are given in Table 3:

The target values for all inputs and outputs of inefficient hospitals along with percentage reduction in inputs and augmentation in outputs are given in Table 3. The slacks in inputs and outputs are given in Table 4, the observed Table 3 shows that on average hospitals have significant scope to reduce inputs and augment outputs relative to the best performing hospital. On average 30.20 % of OI can be augmented and 22.91 % NFC, 14.63 % EE, 10.58 % SW, and 14.34 % CA can be reduced, if all the inefficient hospitals operate at the efficient level.

Table 3. Target values of input and output variables under BCC output oriented model

.	Inputs				Outputs
S.No.	NFC	EE	SW	CA	OI
H1	1.62(0)	0.04(0)	0.46(0)	3.86(0)	3.32(0)
H3	39.95(0)	1.39(0)	8.48(0)	24.22(0)	38.86(0)
H4	6.31(45.31)	0.22(60.19)	1.08(65.18)	8.26(0)	19.18(83.56)
H5	2.55(42.24)	0.08(36.67)	0.55(37.87)	4.09(0)	7.47(114.54)
H6	12.15(0)	0.04(0)	0.25(0)	24.65(0)	3.35(0)
H7	6.32(0)	0.29(0)	12.97(0)	72.39(0)	21.38(0)
H8	8.57(17.57)	0.32(0)	1.97(0)	15.27(8.22)	21.29(149.26)
H9	9.56(77.58)	0.36(0)	1.96(0)	15.62(71.17)	22.03(55.47)
H10	5.76(0)	0.21(68)	2.2(9.84)	15.34(0)	19.21(58.39)
H11	4.62(0)	0.28(0)	0.46(0)	11.71(0)	10.11(0)
H12	5.02(35.64)	0.17(0)	0.8(68.13)	7.65(53.21)	18.43(119.40)
H14	8.34(0)	0.31(20.77)	1.58(25.31)	13.10(0)	21.05(138.93)
H15	32.3(0)	1.30(0)	5.70(0)	44.64(0)	39.72(0)
Mean	10.06(22.91)	0.35(14.63)	2.79(10.58)	19.35(14.34)	18.02(30.20)

Figures in braces are the percentage reduction in the corresponding inputs and percentage addition in corresponding outputs to make the hospital efficient.

3.2 Pure Technical Efficiency (PTE)

Since CCR model works on the bases of CRS in which scale size of the DMU is not considered, so it is relevant to assess technical efficiency. Therefore in order to know whether inefficiency in any hospital is due to inefficient production

Table 4. Slacks in Inputs and Outputs

	Input Slacks				Output Slacks
S.No.	NFC	EE	SW	CA	OI
H1	0	0	0	0	0
H2	0	0	0	0	0
H3	0	0	0	0	0
H4	5.22	0.33	2.03	0	0
H5	1.87	0.04	0.34	0	0
H6	0	0	0	0	0
H7	0	0	0	0	0
H8	1.85	0	0	1.37	0
H9	33.08	0	0	38.56	0
H10	0	0.44	0.24	0	0
H11	0	0	0	0	0
H12	2.78	0	1.71	8.7	0
H13	0	0	0	0	0
H14	0	0.08	0.53	0	0
H15	0	0	0	0	0
Mean	2.99	0.06	0.32	3.24	0

operation or due to unfavorable conditions displayed by the size of hospital, BCC model is also applied. Table 4 also shows details about DEA results calculated by this model. It is evident from the Table4 Column 7 & 8, that out of 15 hospitals only 2 are technically efficient, and 8 hospitals are pure are technical efficient (BCC score = 1), i.e., none of these have scope to further augment outputs (maintaining same input level) while remaining 7 hospitals are relatively efficient (score < 1). PTE measures that how efficiently inputs are converted in to outputs. The overall average of PTE comes out to be 0.771 this means that given the scale of operation, on average a hospital can reduce its inputs by 22.9% of its observed level without detriment to its output levels. PTE is concerned with the efficiency in converting inputs to outputs for the given scale size of the hospitals as we observe that H1, H3, H6, H7, H11 and H15 are CCR technically inefficient while they are PTE.

3.3 Scale Efficiency (SE)

Scale efficiency (SE) is the ratio of the OTE and PTE scores. If the value of the SE score is one, then the hospital is apparently operating at optimal scale size. If the value is less than one, then the hospital appears either small or big relative to its optimum scale size. In Table 4, the SE score of the hospitals is at the 9th column. Result shows that out of 15 hospitals, 2 hospitals are SE, while the remaining 13 hospitals are scale inefficient. The average of the SE is 0.781 which indicates that on average a hospital may be able to decrease its input by 21.9% beyond its best practice average targets under VRS, if we were to operate at CRS.

4 Results and Discussion

Table 5 presents the information on OTE, reference set, peer weights and reference count (peer count) of the sample hospitals for the year 2009–10. The DEA analysis evaluates the set of hospitals which construct the production frontier. The hospitals having values of the OTE score equal to 1 form the efficient frontier and those having the values less than 1 are less efficient relative to the hospitals on the frontier.

Table 5. OTE, PTE, SE and RTS

S.No.	OTE	PTE	SE	RTS
H1	0.728	1	0.728	IRS
H2	1	1	1	-
H3	0.666	1	0666	DRS
H4	0.525	0.545	0.964	DRS
H5	0.295	0.466	0.632	IRS
H6	0.763	1	0.763	IRS
H7	0.921	1	0.921	DRS
H8	0.244	0.401	0.609	DRS
H9	0.361	0.643	0.561	DRS
H10	0.574	0.631	0.909	DRS
H11	0.954	1	0.954	IRS
H12	0.434	0.456	0.952	DRS
H13	1	1	1	-
H14	0.288	0.419	0.678	DRS
H15	0.369	1	0.369	DRS
Mean	0.608	0.771	0.781	

The results indicate that out of 15 hospitals, 2 hospitals (13.33%) are relatively efficient (efficiency score =1) while remaining 13 hospitals (86.67%) are relatively inefficient (efficiency score <1). These two efficient hospitals are (H2), and (H13). These hospitals are on the efficient frontier and thus form the "reference set", i.e., these hospitals can set an example of best operating practice for the remaining 13 inefficient hospitals to emulate. H8 is the most technical inefficient hospital. Among the inefficient hospitals, 5 hospitals have the efficiency scores above the average efficiency scores. The average of OTE scores comes out to be 0.608. This reveals that on average hospital can reduce its resources by 39.20% to obtain the existing level of outputs.

We also use the frequency of efficient hospitals in the reference set (i.e. peer count) to discriminate among them. The higher peer count represents the extent of robustness of that hospital compared with other efficient hospitals. In other words, a hospital with higher peer count is likely to be a hospital which is efficient with respect to a large number of factors and is probably a good example of a "global leader" or a hospital with a high robustness. Efficient hospitals that appear seldom in the reference set are likely to possess a very uncommon input/output mix so

when the peer count is low, one can safely conclude that the hospital is somewhat of an odd unit and cannot be treated as a good example to be followed. On the basis of robustness of efficiency scores, the hospitals on the frontier are classified as:

1. **High robustness:** H13(peer count =7) is considered as high robust hospital as it has maximum peer count, so it can be considered as global leader in terms of technical efficiency.
2. **Middle robustness:** H7 and H15 (peer count =4) are classified in the middle robust group.
3. **Low robustness:** H1 and H3 (peer count = 1), are graded in the low robust group in terms of OTE.

Table 6. Resulting efficiency score of hospitals by DEA model

S.N.	Efficiency Scores	Reference Sets	Peer Weight	Peer Count
H1	1	H1	1	1
H2	1	H2	1	0
H3	1	H3	1	1
H4	0.545	H3,H13	0.037,0.963	0
H5	0.466	H1,H13	0.274,0.726	0
H6	1	H6	1	0
H7	1	H7	1	4
H8	0.401	H7,H13,H15	0.045,0.128,0.827	0
H9	0.643	H7,H13,H15	0.29,0.165,0.806	0
H10	0.631	H7,H13,H15	0.022,0.106,0.872	0
H11	1	H11	1	0
H12	0.456	H13	1	0
H13	1	H13	1	7
H14	0.419	H7,H13,H15	0.015,0.121,0.864	0
H15	1	H15	1	4

5 Conclusion

In this paper, we measure technical and scale efficiencies of some private sector hospitals in India using DEA. The study finds that out of 15 hospitals 2 hospitals (13.33 %) have maximum degree of efficiency. The overall mean technical efficiency of hospitals is 60.80%, indicating that on average 39.20% of the technical potential of hospitals is not in use. This implies that these hospitals have the scope of producing the same output with the inputs 39.20 % less than the existing level. H1 and H13 have scored the technical efficiency score of unity and thus they form the efficient frontier. Among the efficient hospitals H13 is the most efficient hospital. On the contrary H8 is the most inefficient hospital.

The results of BCC model shows that out of 15 hospitals, 8 hospitals (53.33%) are PTE as they efficiently convert their inputs in to outputs. It is also observed that out of 8 BCC efficient hospitals 2 have CRS, 3 have DRS and 3 have IRS, while out of remaining 7 projected hospitals 6 have DRS and 1 have IRS. The model suggested that on average, inefficient hospitals may be able to reduce its NFC by 22.91%, EE by 14.63%, SW by 10.58%, and CA 14.34% relative to the best performing hospital.

Despite the fact the DEA results in this paper give an indication on the degree of hospital's efficiency in the process of transforming inputs in to outputs, the conclusion on the efficiency of hospitals need to be taken with some carefulness. The results of this study are dependent upon the choice of the inputs and outputs.

References

[1] Agrarwal, S., Yadav, S.P., Singh, S.P.: Assessment of Relative Efficiency of Private Sector Hospitals in India. J. of Maths. Sys. Sci. 2, 1–22 (2006)
[2] Banker, R.D., Charnes, A., Cooper, W.W.: Some Models for the Estimation of Technical and Scale Inefficiencies in Data Envelopment Analysis. Management Sci. 30, 1078–1092 (1984)
[3] Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the Efficiency of Decision Making Units. Euro. J. of Operation Research 2, 429–441 (1978)
[4] Hollingsworth, B.: Non-parametric and Parametric Applications Measuring Efficiency in Health Care. Health Care Management Sci. 6, 203–218 (2003)
[5] Ismail, M.A.: Technical Efficiency of Sudan Health Institutions: A State Level Analysis. Sudanese J. of Public Health 5, 122–129 (2010)
[6] Nunamaker, T.R.: Measuring Routine Nursing Service Efficiency: A Comparison of Cost per Patient Day and Data Envelopment Analysis Models. Health Services Research 18, 183–205 (1983)
[7] Ozcan, Y.A.: Health Care Benchmarking and Performance Evaluation: An Assessment Using Data Envelopment Analysis. Springer, New York (2008)
[8] Sherman, H.D.: Hospital Efficiency Measurement and Evaluation: Empirical Test of a New Technique. Medical Care 22, 922–938 (1984)
[9] Srinivasan, R.: Healthcare in India Vision 2025, issue and prospects, Planning Commition, Government of India

Gender Classification Using Artificial Neural Networks through Independent Components

Sunita Kumari and Banshidhar Majhi

National Institute of Technology, India
sunita.nitrkl@gmail.com, bmajhi@nitrkl.ac.in

Abstract. In this paper, an efficient technique for gender classification is developed. It uses the information maximization approach of independent component analysis for extracting the features from the face images. Further, these features were tested using back propagation neural network (BPNN) and radial basis function neural network (RBFNN). The analysis were carried out on FERET database. The main objective of the paper is to build up an optimum classifier using neural networks. The performance of the classifier is estimated through confusion matrix and measured in terms of accuracy, sensitivity and specificity.

Keywords: ICA, infomax, BPNN, RBFNN, Accuracy, Sensitivity, Specificity.

1 Introduction

Neural network is effective tool in the area of pattern recognition. This is achieved through two stages, firstly the learning (approximation) stage where neural network extracts the features from the input data and secondly the recognition (generalization) stage where the network distinguishes the pattern of the input data based on features. The network mainly consists of three layers, namely input, hidden and output layer. The result of recognition is greatly influenced by the hidden layer. The learning of neural network can be specified as a function approximation problem where the goal is to learn an unknown function or have an approximation of it from a set of input-output pairs. The learning of neural network can be supervised or unsupervised. If the input are given with the known labels then learning is referred as supervised otherwise unsupervised learning. Every instance present in any dataset used by pattern recognition algorithm constitutes some set of features. These features can be binary, real coded, categorical or continuous.

Gender classification is one such issue of pattern recognition where the aim is to determine that a given face image belongs to male or female. Gender classification is required for reducing the search space at the time of

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 667–676.
springerlink.com

identification. Gender classification is a two fold problem (i) the development of a classifier which can determine male and female classes correctly and (ii) finding out the set of feature to be given to the classifier. The authors in [6] have discussed about few classifiers but finding an optimum classifier is still an open issue. In[1, 5, 8], the authors have discussed about some of the feature extraction techniques. In this paper, a thrust is given on the classifiers based the neural network which uses the features derived from independent component analysis.

The rest of the paper is organized as follows: Feature extraction technique and classifiers are discussed in Section 2 and 3 respectively. Section 4 deals with the tool for measuring the performance of the classifier. The database and preprocessing of face images are discussed in Section 5. Simulation results are drawn in Section 6. Finally, Section 7 gives concluding remarks.

2 Feature Extraction Technique

Every classifier needs suitable information for classification. Thus, the data presented to the classifier should not be redundant. As the face image contains a lot of redundant information within itself. Hence, a technique is required to remove this spurious information. Independent Component Analysis (ICA) is one of such approach [2, 4]. ICA is a computational method for separating a multivariate signal into additive subcomponents.
It can be defined as:

For the given set of input sample x, ICA finds a linear transform in such a way that

$$s = Wx \tag{1}$$

such that the components, s are as independent as possible. Here, W is an unknown separating matrix and needs to be determined. There exists several algorithms for determining W like *Jade*, *Information Maximization (infomax)* and *Fast fixed point (fast)* ICA. The scheme proposed in the paper is based on the infomax ICA which is described in the following subsection.

2.1 Information Maximization ICA

Infomax is a gradient based neural network which maximizes information from input to the output network as proposed by Bell and Sejnosky [2]. The information maximization is achieved by maximizing the joint entropy of transformed vector $z = g(WX)$, where g is a sigmoidal function. The joint entropy is:

$$H(y) = -E[ln(f(y))] \tag{2}$$

where $f(y)$ is the multivariate joint density function of y.

$$f(y) = \frac{f(x)}{|J_W|} \tag{3}$$

Here, $|J_W|$ denotes the absolute value of Jacobian matrix J_W, which is defined as:

$$J_W = \det[\frac{\partial y_i}{\partial x_i}]_i \tag{4}$$

On combining the above equations, $H(y)$ can be written as:

$$H(y) = E[ln|J_W|] + H(x) \tag{5}$$

Maximization of $H(y)$ can be achieved by adapting W and can be achieved using only the first term.

3 Artificial Neural Network

An artificial neural network (ANN), is a mathematical model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation and can be used as a classifier for classifying gender from face images. Section 3.1 and 3.2 discusses the classification of face images using two well known ANN i.e. BPNN and RBFNN.

3.1 Back-Propagation Neural Network

It is a feedforward neural network with backpropagation algorithm. The input layer neurons contains face image values that constitutes inputs to the next layer neurons. There may be several such hidden layers. The final layer is the output layer where there are two nodes: one for male and another for female class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to whichever class's node has the highest value. A back-propagation algorithm is applied to a feedforward multi-layer neural network. Here, the functional signals flows in forward direction and error signals propagate in backward direction. That's why the name is given as back-propagation neural network. Sigmoidal activation function is chosen for hidden and output layer computational neurons. The architecture of feedforward neural network is shown in Fig. 1.

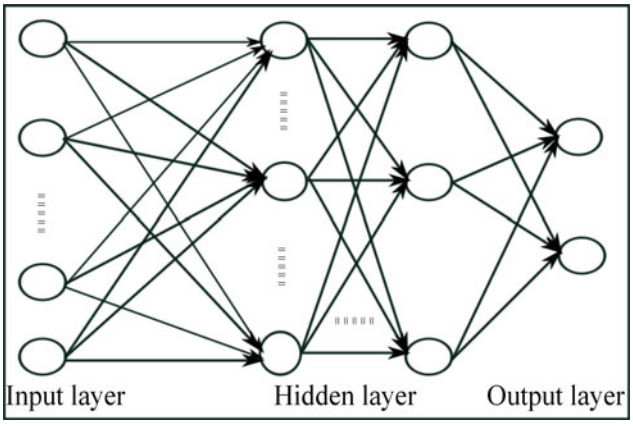


Fig. 1. Architecture of feedforward neural network

3.2 Radial Basis Function Neural Network

A radial function network is an artificial neural network which uses radial basis function as an activation function. It consists of three layers, the input layer, the hidden layer and the output layer. The input layer broadcasts the coordinates of the input vector to each neurons of the hidden layer and then hidden layer produces an activation based on associated radial basis function. Finally, each neuron present in the output layer computes a linear combination of the activation of the hidden units. Here, the gaussian kernel function is used as the radial basis function. Fig. 2 shows the architecture of RBFNN.

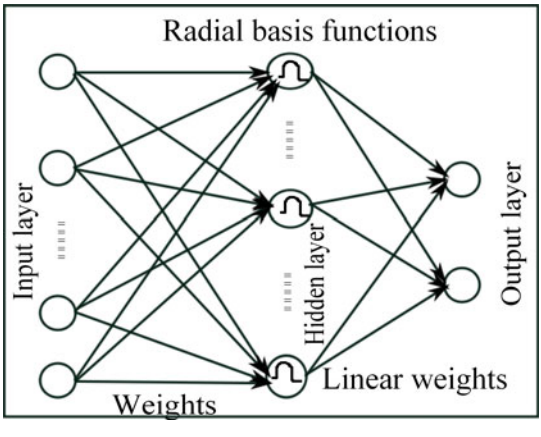


Fig. 2. Architecture of radial basis function neural network

Training RBF networks: The following parameters are determined by the training process.

- The number of neurons in the hidden layer.
- The coordinates of the center of each hidden-layer RBF function.
- The spread (radius) of each RBF function in each dimension.
- The weights applied to the RBF function outputs.

Learning strategies: The design of an RBF network follows three learning strategies depending on the centers of RBFs.

- Fixed centers selected at random
- Self-organized selection of centers
- Supervised selection of centers

This paper follows the supervised selection of centers as it is better than other two approaches as described in [9]. Here, the centers of the radial basis functions and all other learning parameters are changed by supervised learning process. The error correction learning uses gradient descent procedure that represents the generalization of the least mean square error algorithm [3].

4 Performance Measure of Neural Network as a Classifier

Performance of any classifier can be measured using data values present in confusion matrix. A confusion matrix contains information about actual and predicted classifications done by a classification system as shown in Table 1.

Table 1. Confusion Matrix

		Predicted	
		Class–1	Class–2
Actual	Class–1	a	b
	Class–2	c	d

The measures used are:

- Accuracy is the proportion of total number of predictions that were correct. Accuracy $= \frac{a+d}{a+b+c+d}$.
- Sensitivity is the proportion of actual positives that were correctly identified. Sensitivity $= \frac{a}{a+c}$.
- Specificity is the proportion of actual negatives that were correctly identified. Specificity $= \frac{d}{b+d}$.

5 Database Description and Input Source

Analysis were carried out using FERET database [7]. Some of the sample images from this are shown in Fig. 3. The database consists of 500 images consisting 250 male and 250 female. 280 images (140 male and 140 female) were randomly selected to form training set and 120 face images (60 male and 60 female) are chosen for testing. The original image size was 384×256. These images has been aligned first and then conventional triangularization approach is applied to get region of interest as shown in Fig. 4. Image size of each cropped images are 293×241. Further, to reduce the complexity of the system, images were resized to 118×97. ICA is applied on these face images to give input to the classifier. ICA construction is based on 50 principal components. Basis images obtained from ICA are shown in Fig. 5.

Fig. 3. Images from FERET database

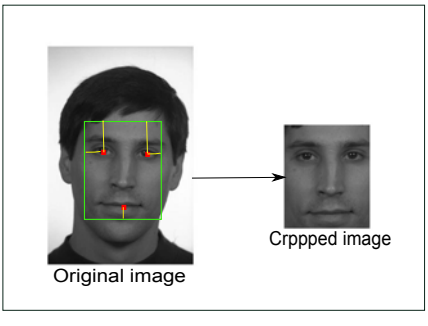


Fig. 4. Conventional Triangularization Approach

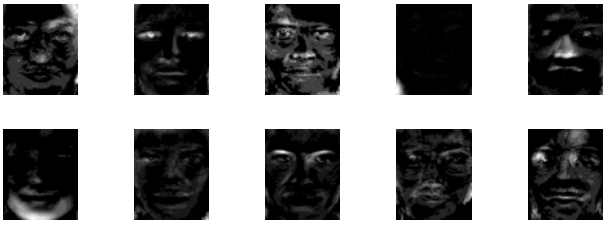


Fig. 5. Basis images obtained from ICA

6 Simulation Results and Discussions

The confusion matrix obtained from the classifiers BPNN and RBFN are shown in Table 2 and their discussions are provided in the following subsections.

Table 2. Confusion Matrix obtained from BPNN and RBFNN

BPNN				RBFN			
		Predicted				Predicted	
		Male	Female			Male	Female
Actual	Male	55	5	Actual	Male	58	2
	Female	3	57		Female	5	55
Accuracy = 93.33%				Accuracy = 94.17%			

6.1 Back Propagation Neural Network

Learning rate (η) and momentum (α) are the controlling parameters for feed-forward BPNN. These two parameters are varied within a range from 0.1-0.9. The simulations are carried for all the combinations of η and α and the results are recorded. The maximum accuracy is found to be 93.33% at $\eta = 0.35$ and $\alpha = 0.5$. The distribution of predicted classes of test images over actual class are shown in Fig. 6. Mean squared error (mse) obtained and performance measures are shown in Fig. 7.

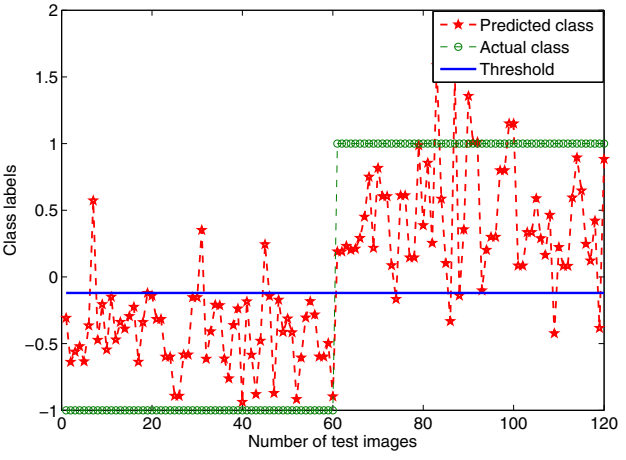


Fig. 6. Plot of actual class vs. predicted class in case of BPNN: LHS shows the distribution of female images and male distributions are shown in RHS

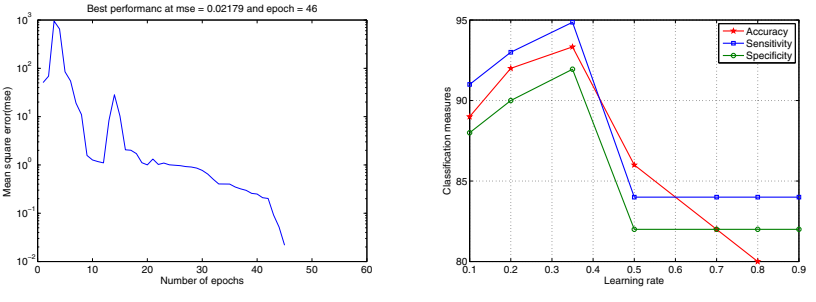


Fig. 7. Plot from BPNN showing mse and classification measures (a) Plot of mse vs. number of epochs in case of BPNN and (b) Plot of accuracy, sensitivity, and specificity vs. learning rate

6.2 Radial Basis Function Neural Network

In case of RBFNN, there are three controlling parameters, $\eta 1$, $\eta 2$ and $\eta 3$ which corresponds to number of neurons in the hidden layer, position of the center in the hidden layer and the spread of centers in the hidden layer respectively. These parameters are varied accordingly and the maximum accuracy obtained is 94.17%. This is achieved at 50 neurons and spread = 1. The distribution of predicted class over actual class is shown in Fig. 8. The mean squared error obtained during training and the plot of different classification measures like accuracy, sensitivity and specificity are drawn in Fig. 9.

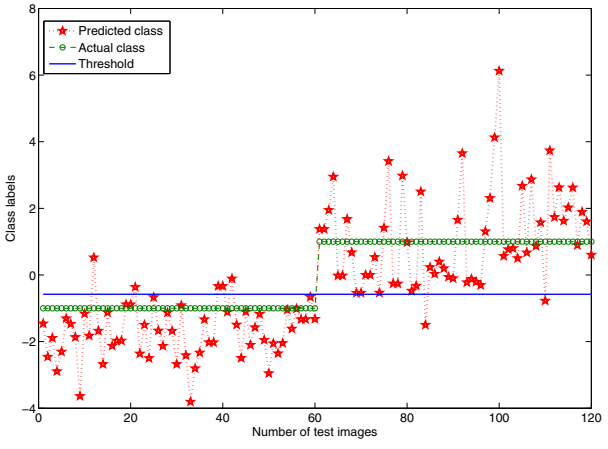


Fig. 8. Plot of actual class vs. predicted class in case of RBFNN: LHS shows the distribution of female images and male distributions are shown in RHS

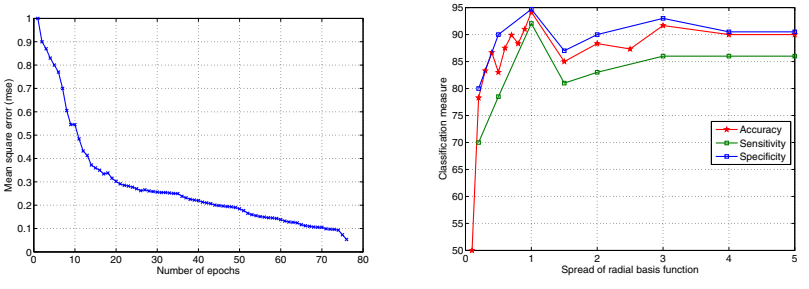


Fig. 9. Plot from RBFNN showing mse and classification measures (a) Plot of mse vs. number of epochs in case of RBFNN and (b) Plot of accuracy, sensitivity, and specificity vs. spread of radial basis functions

7 Conclusion

This paper gives the comparison between back propagation neural network and radial basis function neural network in case of gender classification. RBFNN works better than the BPNN. Infomax ICA is used for extracting features from face images.

References

1. Bartlett, M., Movellan, J., Sejnowski, T.: Face recognition by independent component analysis. IEEE Transactions on Neural Networks 13(6), 1450–1464 (2002)
2. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7, 1129–1159 (1995)

3. Haykin, S.: Neural Networks: A Comprehensive Foundation. IEEE (1999)
4. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Networks 13, 411–430 (2000)
5. Jain, A., Huang, J.: Integrating independent components and support vector machines for gender classification, vol. 3, pp. 558–561 (2004)
6. Moghaddam, B., Yang, M.-H.: Learning gender with support faces. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 707–711 (2002)
7. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face recognition algorithms. (a) Example 1 (b) Example 1 (c) Right ear (d) Mirroed Left ear
8. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognitive Neuroscience 3(1), 71–86 (1991)
9. Vipsita, S., Shee, B., Rath, S.: An efficient technique for protein classification using feature extraction by artificial neural networks. In: 2010 Annual IEEE India Conference (INDICON), pp. 1–5 (December 2010)

Design Optimization of Three Wheeled Motor Vehicle: A GA Approach

Manoj Thakur[1] and Kusum Deep[2]

[1] School of Basic Sciences, Indian Institute of Technology, Mandi, Mandi-175001, India
manojpma@gmail.com
[2] Department of Mathematics, Indian Institute of Technology, Roorkee, Roorkee-247667, India
kusumfma@iitr.ernet.in

Abstract. In this work the problem of finding the optimum design of suspension system for two of the most commonly used Indian commercial three-wheeled motor vehicles namely Bajaj rear engine (RE) and Vikram front engine (FE) vehicle is formulated as an nonlinear optimization problem having decision variables as spring stiffness, viscous damping force of the front and rear suspension, wheelbase and track width. A real coded genetic algorithm (RCGA) has been applied to optimize system parameters to minimize the root mean square acceleration spectral density. The results are compared with Random search technique (RST2). It is observed that the solutions obtained using both algorithms lie within the International Standard Organization (ISO) 2631 values (ISO I [1997]). In all the models RCGA performs significantly better than RST.

Keywords: three wheeled motor vehicle, genetic algorithms, Laplace crossover, Power mutation, RST2.

1 Introduction

The three-wheeled vehicle suspension system are designed to provide adequate vibration separation in different directions due to road disorders in order to achieve the required level of comfort for the driver and commuters. The most common parameter affecting the ride performance is the vehicle suspension characteristics, mass, inertia and other geometrical parameters etc. In order to improve the comfort level, the optimum values of these parameters which minimize the uneasiness of the driver and commuters are desired.

The stiffness and damping of front and rear suspension systems, Wheelbase of three wheeled vehicle, Track width of the vehicle have been found mainly to influence the ride behavior.

An optimization model may be modeled to minimize the vertical or horizontal root mean square acceleration response (RMSAR). Ramji [10] has shown that the horizontal RMSAR values for both Vikram and Bajaj vehicles always comes well

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 677–688.
springerlink.com

within the ISO limits in the whole frequency in the interval [0.1, 80] Hz, However vertical RMSAR surpasses the ISO limits in the mid frequency range. So it is required to get the optimal parameter setting only to minimize vertical RMSAR at the center of gravity of the sprung mass, resulting in slight penalty of the vehicle ride behaviour in the lateral direction.

In some of the earlier works in the field of vehicle dynamics (Duncan [1], Tamboli and Joshi [4], Castillo et al.[6], Afimiwala and Mayne [7], Demic [15], Elmadary and Dokainish [16], Pintado and Bentiz [18], Bhave and Kaul [19], Dahlberg, [20], Tan and Huston [21], Elmaraghy et al. [22], Nack [23], Peilu [24]) classical linear and nonlinear optimization techniques were used to find the RMSAR of the driver and commuters. Many soft computing based techniques such as Simulated Annealing approach and Genetic algorithm were also applied to get the global optimal solution of the non-linear optimization problems.

The previous work relating to optimization behavior of vehicles other than three wheeled vehicles gives some idea in the evaluation of optimum ride response of the vehicles. Only a small amount of work (Ramji and Goel [12, 13, 14], Tan and Huston [20]) dealing with ride behavior of three-wheeled vehicles is available in the literature. Recently, RST2 algorithm of Mohan and Shanker [3] has been applied to get the near global optimum solution of the suspension system of three-wheeled motor vehicle by Ramji et. al. [11].

2 Problem Formulation

In order to evaluate the performance of a vehicle, RMSAR has been used as criteria. The objective function is nonlinear function of design variables and computes the vehicle comfort level based on RMSAR. The objective function depends upon thirteen different vehicle and suspension parameters. Only vertical RMSAR has been considered as objective function to minimize from the ride comfort view point. The weighted root mean square acceleration response is given by:

$$\text{RMSAR} = \sqrt{\int_{\omega_1}^{\omega_2} \omega^4 \mathrm{B}(\omega)\, S_{\mathrm{q}_r}(\omega) d\omega} \tag{1}$$

where, $S_{\mathrm{q}_\mathrm{r}}(\omega)$ is the spectral density function for displacement. $\mathrm{q}_\mathrm{r}(\mathrm{t})$ and $[\omega_1, \omega_2]$ is the frequency interval in which it is computed and, $B(\omega)$ is weighting factor.

The ISO fatigue time is inversely proportional to RMSAR. Therefore, the objective function has been taken as to minimize square of RMSAR by using Eq. 1 and by taking appropriate weighing factors at different frequencies as proposed in ISO 2631 [16].

$$\textit{Min}\ \text{RMSAR} = \int_{\omega_1}^{\omega_2} \omega^4 \mathrm{B}(\omega)\, S_{\mathrm{q}_\mathrm{r}}(\omega) d\omega \tag{2}$$

A recently developed RCGA which uses Laplace crossover and Power mutation (LX-PM) is used to find the optimum values of the design variables for getting

minimum vertical RMSAR in the frequency range of 0.1 to 80 Hz. Selecting a multi-objective function and allowing suspension and vehicle parameters to vary over a wide range with imposition of constraints, the optimal values of the design variables have been obtained from a passenger ride comfort viewpoint for a rough road excitation.

The problem modeled is a nonlinear multi-objective optimization problem with objective function of the type

$$\begin{aligned} &Min\{f_1(x), f_2(x),..., f_t(x)\},\ x=(x_1, x_2,..., x_n)\in R^n \\ &Suject\ to:\ g_i(x)\ \geq\ 0, \quad i\ =\ 1,2,...,m\ ; \\ &x_k^{(L)} \leq x_k \leq x_k^{(U)}, Where\, k=1,\ 2,\ 3,...,\ n\ . \end{aligned} \tag{3}$$

where, $f_1, f_2,..., f_t$ are t number of objective functions that have to be minimized simultaneously.

Ramji [11] have used two different methods to solve the multi-objective optimization problem given by equation (3).

In the first approach the sum of the values of vertical RMSAR in the frequency range of 0.1 to 80 Hz is minimized.

$$\begin{aligned} &Min\ f(x)=\sum_{\omega_i} f_i(x)\ ,\ \ x=(x_1, x_2,..., x_n) \\ &\text{Subject to: } g_j(x)\ \geq\ 0, \quad j\ =\ 1,2,...,m \\ &x_k^{(L)}\ \leq\ x_k \leq x_k^{(U)}, \quad \text{where } k=1,\ 2,...,n\ . \end{aligned} \tag{4}$$

In the second approach the maximum of RMSAR value is minimized i.e.

$$\begin{aligned} &Min[Max(f_i(x))],\ i=1, 2, ...,t\ ,\ \ x=(x_1, x_2,..., x_n) \\ &\text{Subject to:}\quad g_j(x)\geq 0, \quad j\ =\ 1,2,...,m\ ; \\ &x_k^{(L)}\ \leq\ x_k \leq x_k^{(U)}, \quad k=1,\ 2,...,n \end{aligned} \tag{5}$$

In the above two approaches, multi objective optimization problem give rise to a set of optimal solutions of equal importance as far as all objectives are concerned. By taking parametric analysis into consideration, out of the number of available solutions, an appropriate design vector solution is recommended.

3 First Genetic Algorithms

GAs are population based heuristic search techniques which mimic Darwin's principle of natural selection and genetic inheritance. The idea of GAs was introduced by Holland [5]. A detailed description of GA could be found in Goldberg [2]. In GA, a population of potential solutions (termed as chromosomes) is evolved over successive generations using a set of genetic operators called selection, crossover

and mutation. Every solution is assigned a fitness value based on some criteria, and the selection operator is applied to choose comparatively 'fit' chromosomes to be a part of reproduction process. In reproduction phase new individuals are created through crossover and mutation operators. Crossover operator blends the genetic information between chromosomes to explore the search space, whereas mutation operator is used to maintain adequate diversity in the population of chromosomes to avoid premature convergence.

We now state the Laplace Crossover Operator and the Power Mutation Operator used in LX-PM.

3.1 Laplace Crossover Operator

The Laplace Crossover Operator (LX) by Deep and Thakur [8] is a parent centric operator that uses Laplace Distribution. The density Function Laplace distribution is given by

$$f(x) = \frac{1}{2b}\exp\left(-\frac{|x-a|}{b}\right), -\infty < x < \infty$$

and distribution function of Laplace distribution is given by

$$F(x) = \begin{cases} \frac{1}{2}\exp\left(\frac{|x-a|}{b}\right), & x \le a \\ 1-\frac{1}{2}\exp\left(-\frac{|x-a|}{b}\right), & x > a \end{cases}$$

Where, $a \in R$ is called the location parameter and $b > 0$ is termed as scale parameter.

LX may be used to produce two offsprings $y^{(1)} = (y_1^{(1)}, y_2^{(1)}, ..., y_n^{(1)})$ and $y^{(2)} = (y_1^{(2)}, y_2^{(2)}, ..., y_n^{(2)})$ from a pair of parents $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, ..., x_n^{(1)})$ and $x^{(2)} = (x_1^{(2)}, x_2^{(2)}, ..., x_n^{(2)})$ in the following way.

- Generate uniformly distributed random number u_i and r_i in the interval $[0,1]$.
- Generate a number β is which follows the Laplace distribution using the following formula

$$\beta = \begin{cases} a - b\log_e(u_i), & r_i \le 0.5 \\ a + b\log_e(u_i), & r_i > 0.5 \end{cases}$$

- The offsprings are given by the equation

$$y_i^{(1)} = x_i^{(1)} + \beta\left|x_i^{(1)} - x_i^{(2)}\right|$$

$$y_i^{(2)} = x_i^{(2)} + \beta\left|x_i^{(1)} - x_i^{(2)}\right|$$

From the above two equations it is clear that

- Both the offsprings are placed symmetrically with respect to the position of the parents.
- Lower the value of b, higher is the probability of creating offsprings near the parents.
- For a fixed value of a and b, LX dispenses offsprings proportional to the spread of parents.

3.2 Power Mutation

The Power Mutation (PM) by Deep and Thakur [9] is based on Power distribution. Its distribution function is given by

$$f(x) = px^{p-1} \qquad 0 \le x \le 1$$

and the density function is given by

$$F(x) = x^p \qquad 0 \le x \le 1$$

where p is the index of the distribution. The PM is used to create a solution y in the vicinity of a parent solution $\bar{x}$ in the following manner.

- Find a uniform random number $t \in [0,1]$.
- Create a random number s which follows the above mentioned distribution using the following formula

  $s = r^{\frac{1}{p}}$, where r is a uniform random number in the interval $[0,1]$
- Use following formula is used to create the muted solution

$$y = \begin{cases} \bar{x} - s(\bar{x} - x^l) & if\ t < r \\ \bar{x} + s(x^u - \bar{x}) & if\ t \ge r \end{cases}$$

where $t = \dfrac{\bar{x} - x_i^l}{x_i^u - \bar{x}}$ and x^l and x^u are lower and upper bounds of the decision variable.

From the above two equations it is clear that

- The strength of mutation is governed by the index of the mutation (p).
- For small values of p less perturbance in the solution is expected and for large values of p more diversity is achieved.

- The probability of producing a mutated solution y on left (right) side of $\bar{x}$ is proportional to distance of $\bar{x}$ from x^l (x^u) and the muted solution is always feasible.

The pseudo code of the GA (LX-PM) is given below.

```
  begin
  Generation = 0
    Initial population randomly
    Evaluate fitness of entire     population
     While (termination criterion is not
     satisfied) do
      Generation = Generation + 1
      Apply Selection operator
      Apply Crossover operator
      Apply Mutation operator
     end do
end begin
```

5 Experimental Setup, Results and Discussions

The vertical RMSAR on the same road for the Bajaj and Vikram vehicles has been minimized at 45kmph which is normally the vehicle speed in city travel. Thirteen different vehicle and suspension parameters are used as designed variable. The upper and lower bounds for all the suspension parameters (stiffness and damping) are taken in the range of ± 60 % of the original values in addition the stiffness and damping characteristics of left and right rear suspensions are taken same for both the vehicles. The wheelbase and track width, which affect the ride behavior, are taken in the range of ± 20 % of the original values.

The population size is taken as ten times the number of variables (13*10 = 130 in this case). Since the algorithms used are heuristic in nature, 30 independent runs are taken with different initial populations. The crossover probability and mutation probability for LX-PM are fixed to be 0.6 and 1/(no. of decision variables) in all cases. The crossover and mutation index are fixed to be 0.20 and 0.25 respectively. Tournament selection operator with tournament size 2 is applied. Elitism is applied with size one. The termination criteria are fixed to be a maximum of 200 generations.

The bounds on decision variables for Bajaj and Vikram vehicles are given in Table 1. The optimal values of decision variables for Bajaj and Vikram vehicle using both Min-Max and summation approach are given in Table 2. The minimum, average and standard deviation of 30 independent runs of LX-PM and the best results obtained by RST2 algorithm are given in Table 3. It is observed that in all cases LX-PM performs better than RST2 algorithm also the best and average objective function objective function value obtained by LX-PM is less than the best

value obtained by RST2 algorithm. The standard deviation of optimal solution obtained by LX-PM is also very less indicating that LX-PM is able to locate the near optimal solution consistently in all runs.

Table 1. Upper and Lower Bounds on parameters of Bajaj Vehicle and Vikram Vehicle

Variable	Upper Bound	Lower Bound	Upper Bound	Lower Bound
Front stiffness k1	13080.000000	52320.000000	34680.000000	138720.000000
Rear right and left k2 = k3	19920.000000	79680.000000	31920.000000	127680.000000
Front damping c1	1400.000000	5600.000000	818.400000	3273.600000
Rear right and left c2 = c3	883.200000	3532.800000	788.400000	3153.600000
Seat suspension stiffness ks	1560.000000	6240.000000	1560.000000	6240.000000
Seat suspension damping force cs	180.000000	720.000000	180.000000	720.000000
Track width l1 = l2	0.460000	0.690000	0.467200	0.700800
Wheel base1 l3	1.149600	1.724400	1.159600	1.739400
Wheel base2 l4	0.450400	0.675600	0.331600	0.497400
Seat distance l5	0.377256	0.565884	0.396902	0.595354
Mass of sprung mass Ms	176.720000	706.880000	188.000000	752.000000
Moment of Inertia Ipy	68.000000	272.000000	80.000000	320.000000
Moment of Inertia Ipx	72.880000	291.520000	83.760000	335.040000

The vertical RMS acceleration responses obtained using LX-PM at the center of gravity of the sprung mass for Bajaj vehicle for both Summation and Min/Max the approaches are shown in Fig. 1 and Fig. 2. The results obtained by RST2 together with the ISO limits are also shown. Considerable improvements in vertical RMS acceleration responses are observed for Min-Max as well as Summation approaches. Fig. 3 and Fig. 4 consist of RMS acceleration responses of Vikram vehicle for Summation and Min/Max the approaches respectively. The results obtained by LX-PM and RST2 are shown in both the cases together with the ISO standards. The RMSAR value in the entire frequency range is found to be well below the ISO standards using both the algorithms. LX-PM particularly performs better in the mid frequency range where the reference values are violating the ISO limits. In all the cases LX-PM is found to clearly outperform RST2 in the mid frequency range as well as in on an average in the entire frequency range considered.

Table 2. Value of decision variables for Bajaj and Vikram Vehicle using Min-Max and Summation approach

Decision Variables	Bajaj Vehicle		Vikram Vehicle	
	Min-Max approach	Summation approach	Min-Max approach	Summation approach
Front stiffness k1	13080.000000	13080.000000	34680.000000	34680.000000
Rear right and left k2 = k3	29155.021550	19920.000000	42510.810125	31920.000000
Front damping c1	1400.000000	1400.000000	1163.113952	818.400000
Rear right and left c2 = c3	1044.961759	883.200000	788.400373	788.429998
Seat suspension stiffness ks	6239.999854	6240.000000	6240.000000	6240.000000
Seat suspension damping force cs	181.685929	180.000000	180.000000	180.647995
Track width l1 = l2	0.690000	0.690000	0.700505	0.700800
Wheel base1 l3	1.199306	1.149600	1.722438	1.739400
Wheel base2 l4	0.542516	0.616198	0.346520	0.331600
Seat distance l5	0.565884	0.565884	0.396902	0.396902
Mass of sprung mass Ms	616.961844	706.880000	188.000000	277.576726
Moment of Inertia Ipy	271.740612	272.000000	133.340939	85.810056
Moment of Inertia Ipx	165.779462	118.062082	299.968328	171.398767

Table 3. Optimal Results for Bajaj and Vikram Vehicle using Min-Max and Summation approach (30 runs each by LX-PM and RST)

Algorithm		Bajaj Vehicle		Vikram Vehicle	
		Min-Max approach	Summation approach	Min-Max approach	Summation approach
LX-PM	Min.	0.30107241	1.03589800	0.44791401	2.12527822
	Avg.	0.32144637	1.03589821	0.45523057	2.13255428
	S.D.	0.02441122	0.00000011	0.00516735	0.00369435
RST	Min.	0.43515105	1.20574315	0.60129576	2.36489765

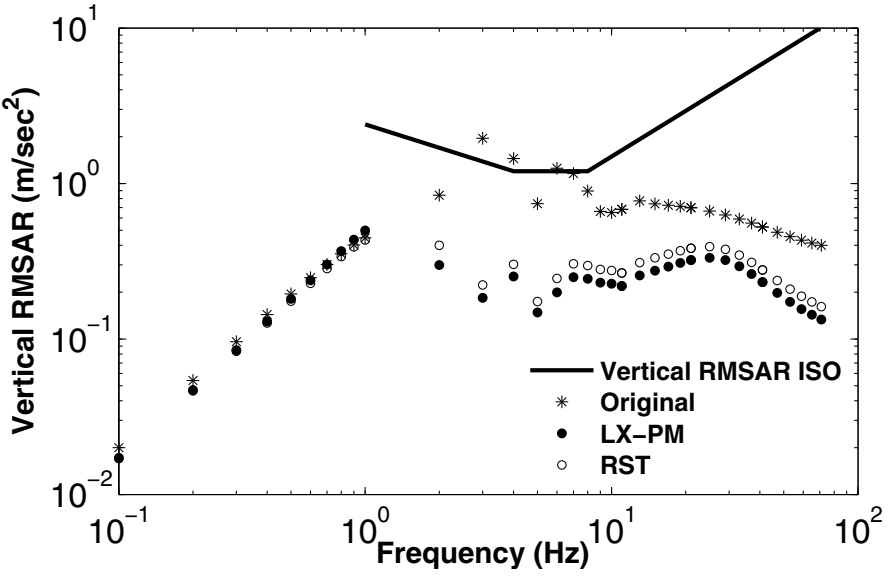


Fig. 1. Vertical RMSAR of Bajaj vehicle sprung mass center by Min/Max approach

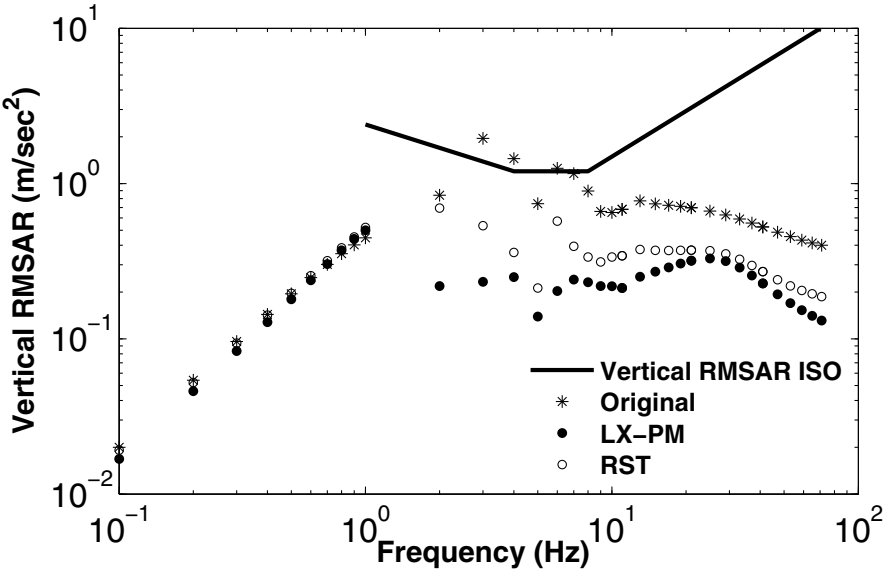


Fig. 2. Vertical RMSAR of Bajaj vehicle sprung mass center by Summation approach

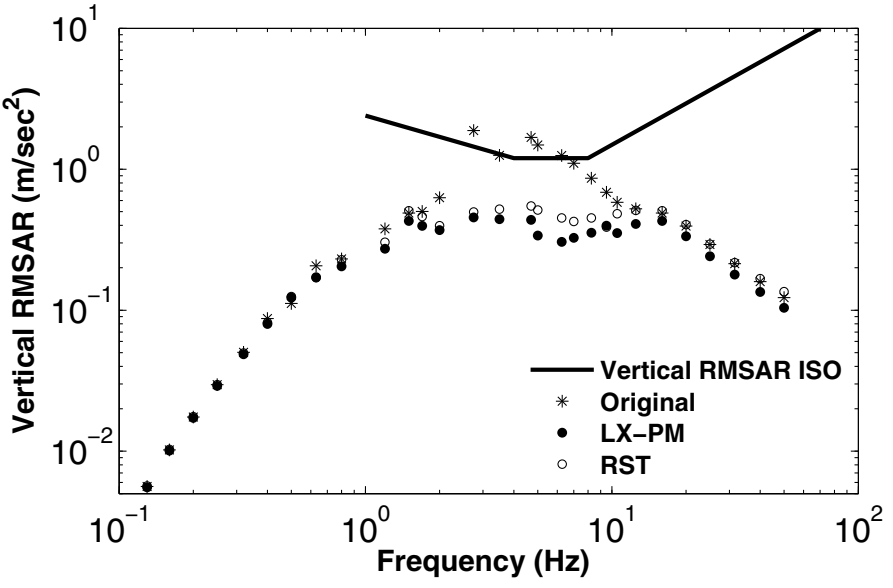


Fig. 3. Vertical RMSAR of Vikram vehicle sprung mass center by Min/Max approach

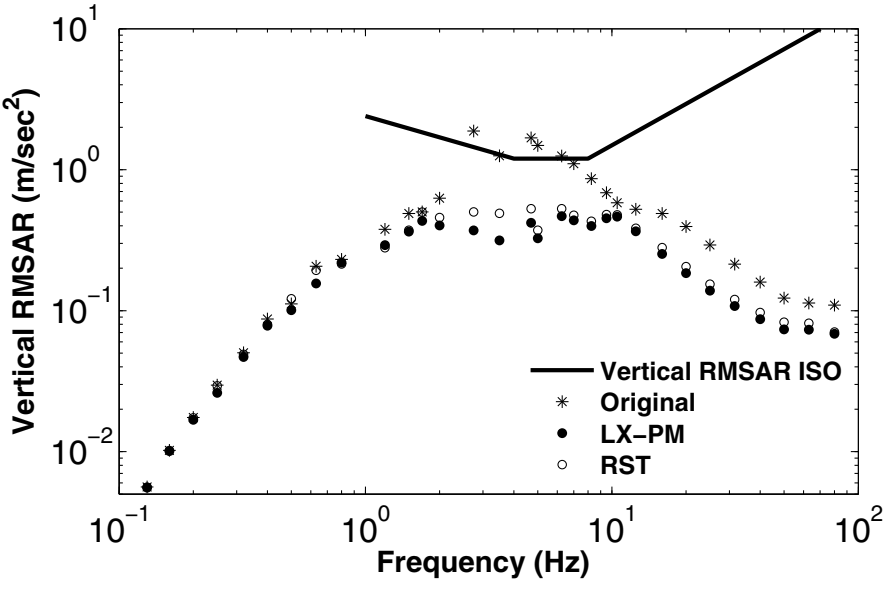


Fig. 4. Vertical RMSAR of Bajaj vehicle sprung mass center by Summation approach

6 Conclusion

The suspension system design problem for Bajaj and Vikram vehicle is considered as an optimization problem involving many decision variables influencing ride behavior. The problem turns out to be a nonlinear multi objective optimization problem. Although there are a number of ways in which a multi objective optimization problem can be solved, in this study we have considered two of the simplest approaches viz: Min-Max approach and Summation approach. The LX-PM RCGA is used to solve the optimization problems. In each case 30 runs are taken and the minimum, mean and standard deviation values of the objective functions are recorded. In order to compare the results with the existing algorithms all the models are solved using RST2 algorithm also. Tough the RMSAR corresponding to optimal parameter suggested by both the algorithms lie within the ISO limits, the results obtained by LX-PM are constantly found to outperform RST2 algorithm for both the Bajaj and Vikram vehicles using the Min-Max and Summation approaches.

References

[1] Duncan, A.E.: Application of modal modeling and mount system optimization to light duty truck ride analysis. SAE Transactions, 4075–4089 (1982)
[2] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, New York (1989)
[3] Mohan, C., Shanker, K.D.: A controlled random search technique for global optimization using quadratic approximation. Asia-Pacific Journal of Operations Research 11, 93–101 (1994)
[4] Tamboli, J.A., Joshi, S.G.: Optimum design of a passive suspension system of a vehicle subjected to actual random road excitations. Journal of Sound and Vibration 2(219), 193–205 (1999)
[5] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
[6] Castillo, J.M.D., Pintado, P., Bentiz, F.G.: Optimization for vehicle suspension 2: Frequency domain. Vehicle System Dynamics 19, 331–352 (1990)
[7] Afimiwala, K.A., Mayne, R.W.: Optimum design of an impact absorber. Journal of Engineering for Industry (73)
[8] Deep, K., Thakur, M.: A New Crossover Operator for Real Coded Genetic Algorithms. Applied Mathematics and Computations 188(1), 895–911 (2007a)
[9] Deep, K., Thakur, M.: A New Mutation Operator for Real Coded Genetic Algorithms. Applied Mathematics and Computations 193(1), 211–230 (2007b)
[10] Ramji, K.: Coupled Vertical - Lateral Dynamics of Three-Wheeled Motor Vehicles. Ph. D. Thesis. Department of Mechanical Engineering, Indian Institute of Technology Roorkee, Roorkee, India (2004)
[11] Ramji, K., Goel, V.K., Deep, K., Thakur, M.: Optimum Design of Suspension System of Three - Wheeled Motor Vehicles. World Journal of Modeling and Simulation 3(1), 36–44 (2007)
[12] Ramji, K., Goel, V.K.: Coupled vertical-lateral dynamics of three-wheeled motor vehicles, pp. 20–24. Technical University of Denmark, Denmark (2001)

[13] Ramji, K., Goel, V.K.: Accepted for publication in Journal of Institute of Engineers (India) and Presented at 17th Seventeenth Mechanical Engineering Paper Meeting, Indore (2001)
[14] Ramji, K., Goel, V.K.: Ride characteristics of three wheeled motor-vehicles, 26–27. Institute of Engineers (India) local center, Indore (2001)
[15] Demic, M.: Optimization of vehicles elasto-damping elements characteristics from the aspect of ride comfort. Vehicle System Dynamics 23, 351–377 (1994)
[16] Elmadary, M.M., Dokainish, M.A.: Optimum design of tractor semi trailers suspension systems. SAE Transactions, 4497–4505 (1980)
[17] Mechanical vibration and shock evaluation of human exposure to whole body vibrations, ISO 2631-Part 1 (1997)
[18] Pintado, P., Bentiz, F.G.: Optimization for vehicle suspension-1: Time domain. Vehicle System Dynamics 19, 273–288 (1990)
[19] Bhave, S.Y., Kaul, V.: Optimization of vehicle suspension parameters by minimizing RMS acceleration at a definite point. Journal of Institution of Engineers (India) 80, 111–116 (1999)
[20] Dahlberg, T.: Optimization criteria for vehicles traveling on a randomly profiled road- a survey. Vehicle System Dynamics 8, 239–252 (1979)
[21] Tan, T.E., Huston, I.C.: Three-wheeled atv-a no suspension rigid rider system, part 2: Applications- Handling and Ride. In: SAE International off- Highway and Power plant Congress & Exposition, Mecca, Milwaukee, Wisconsion (1984)
[22] Elmaraghy, W.H., Dokainish, M.A., Siddall, J.N.: Minimax optimization of railway vehicle suspensions. ASME
[23] Nack, W.V.: Optimization for vibration isolation. International Journal for Numerical Methods in Engineering 20, 89–100 (1984)
[24] Peilu, X., Li, H.L., Papalambros, P.: A design procedure for the optimization of vehicle suspensions. International Journal of Vehicle Design 5(1-2), 129–142 (1984)

In-situ Efficiency Determination of Induction Motor through Parameter Estimation

S. Anil Chandrakanth[1,*], Thanga Raj Chelliah[2], S.P. Srivastava[1], and Radha Thangaraj[2]

[1] Department of Electrical Engineering, Indian Institute of Technology, Roorkee
anilfren1012@gmail.com
[2] Water Resource Development and Management Department, Indian Institute of Technology, Roorkee

Abstract. Due to robustness, reliability, low price and maintenance free, induction motors (IMs) are used in most of the industrial applications. The influence of these motors (in terms of energy consumption) in energy intensive industries is significant in total input cost. The exact knowledge of some of the induction motor parameters is very important to implement efficient control schemes and its in-situ efficiency determination. These parameters can be obtained by no-load test that is not easily possible for the motors working in process industries where continuous operation is required. Here, Particle Swarm Optimization is used for in-situ efficiency determination of induction motor (5 hp) without performing no-load test. Results are compared with physical efficiency measurement method. The error between estimated and actual efficiencies is found for different objective functions and for different standards.

Keywords: Induction motor, particle swarm optimization, parameter estimation, efficiency determination, efficiency standards.

1 Introduction

There has been a growing global concern over energy consumption and the environment and high energy efficiency has become one of the most important factors in the development of the products that consume electrical energy [1]. Induction motor is a high efficiency electrical machine when working close to its rated torque and speed. However, at light loads, no balance in between copper and iron losses, results considerable reduction in the efficiency. The part load efficiency and power factor can be improved by making the motor excitation adjustment in accordance with load and speed. In this situation an accurate in situ efficiency determination of these motors is essential but requires the motor's electrical parameters.

* Corresponding author.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 689–700.
springerlink.com

Many nonlinear programming techniques like Newton-Raphson technique, cyclic method, Hook, Jeeves, and Rosenbrock methods have been applied to parameter estimation and hence efficiency determination of induction motors. The optimum determined by the Newton-Raphson technique depends heavily on the initial guess of the parameter, with the possibility of a slightly different initial value, causing the algorithm to converge to an entirely different solution [2]. Also, this algorithm needs derivative during the optimization process, which may be difficult to calculate. In [3], the authors proved that the Rosenbrock method is better than scatter search and Hook and Jeeves methods in terms of fast and efficient search. Apart from conventional methods, some of the evolutionary techniques like genetic algorithm (GA) [4], genetic programming [5], particle swarm optimization (PSO) [6], differential evolution (DE) [7] evolution strategy [8] and a PSO variant [8] have been successfully applied to induction motor parameter estimation. In the present study, PSO is used to calculate the in-situ efficiency of induction motor.

The efficiency can be defined from the individual loss terms as:

$$\eta = \frac{P_{out}}{P_{in}} = \frac{P_{in} - P_{loss}}{P_{in}} = 1 - \frac{P_{loss}}{P_{in}} \tag{1}$$

Therefore in principle three types of efficiency measurements may be used:

- Direct measurement of electrical input and mechanical output power
- Direct measurement of the overall losses and the input power
- Measurement of the individual loss components and the input power

The individual loss components are the following:

$$P_{loss} = P_{J1} + P_{J2} + P_{mech} + P_{iron} + P_{additiona} \tag{2}$$

Where,

P_{J1} : Stator joule losses obtained from stator resistance corrected for temperature

P_{J2} : Rotor joule losses obtained from the slip corrected for temperature

P_{iron} : Iron losses, obtained from a no-load test

P_{mech} : Mechanical friction and windage losses, obtained from the no-load test

$P_{additional}$: Additional load losses i.e. losses that are not covered by the other loss components, referred to as stray load losses or supplementary losses.

The last component, the additional load losses, is the critical component, which determines the variation of efficiency among the standards.

The article is organized as follows. Section 2 explains the standards for induction motor efficiency determination. Section 3 describes the particle swarm optimization used in this study. In section 4, mathematical model of the in-situ efficiency determination is given and section 5 gives the experimental setup and parameter settings of the algorithm. In section 6, results and discussions are given and section 7 is the conclusion.

2 Standards for Induction Motor Efficiency Determination

The methods for efficiency measurements can be divided into two categories: direct and indirect methods [4]. In the direct method, the torque has to be assessed in one way or another. Three different standards are discussed: IEEE standard 112, IEC standard 34-2 and JEC standard.

2.1 IEEE Method 112-1996

The IEEE 112-1996 consists of five basic methods to determine the efficiency: A, B, C, E and F [4]. In method A, the input and output power is measured and the efficiency is directly obtained as the ratio of output power to input power. This method is only used for very small machines. Method B employs a direct method to obtain the stray load losses. To reduce the influence of the measuring error, a linear regression is made of the stray load losses at different loads, versus the torque squared.

In method C, total stray load losses are obtained via a separation of losses for both motor and generator, which are then divided between the motor and generator proportional to the rotor currents. In method E, the stray load losses are directly measured using the reverse rotation test. In method E_1, the stray load losses are set to an assumed value. For a 3.74 KW induction motor, stray load losses are assumed to be 1.8% of output power [9]. Methods F and F_1 use equivalent circuit of the machine [4]. The stray load losses are again directly measured or in the case of F1 an assumed value is used. Motors with ratings >180 kW can be tested using methods B, C, E, E_1, F or F_1.

2.2 IEC 34-2 Method

For poly phase induction machines, the preferred method of determining efficiency is the summation-of-losses method. Because of the unavoidable measurement errors, direct determination by measuring the power input and output is generally not accurate enough at efficiency >90% and is therefore usually applied only in the range <100 kW [10]. According to the summation-of-losses method, efficiency is determined by calculating total losses which include the following:

- Constant losses: Losses in active iron (hysteresis and eddy-current losses including additional no- load stray losses), Friction losses, Windage losses
- Current (load) dependent losses: I^2R losses in the stator and rotor windings
- Additional load losses: additional losses occurring at load in active iron and other metal parts, eddy-current losses due to current-dependent leakage fields.

In accordance with IEC 34-2, the additional load losses are assumed to be equal to 0.5% of the power input P_i and to vary as the square of the stator current [4].

2.3 JEC Method

The efficiency evaluation through the Japanese standard can be considered as an indirect method. This method neglects the stray losses [11]. Hence, the obtained efficiency values are generally higher. Also, no thermal correction of the Joule losses is specified because it is very difficult to find the measurement procedures prescribed by the Japanese standard, it is reasonable to evaluate the machine efficiency using the results of the test which are required by the other Standards [11].

3 Particle Swarm Optimization

The concept of PSO was first suggested by Kennedy and Eberhart in 1995 [12].The mechanism of PSO is inspired from the complex social behaviour shown by the natural species. For a D-dimensional search space, the position of the i^{th} particle is represented as $Xi = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Each particle maintains a memory of its previous best position $P_{best\ i} = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The best one among all the particles in the population is represented as $P_{gbest} = (p_{g1}, p_{g2}, \ldots, p_{gD})$. The velocity of each particle is represented as $Vi = (v_{i1}, v_{i2}, \ldots, v_{iD})$. In each iteration, the P vector of the particle with best fitness in the local neighbourhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity [10]. The two basic equations that govern the working of standard PSO (PSO) are that of velocity vector and position vector given by

$$v_{id} = wv_{id} + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \tag{3}$$

$$x_{id} = x_{id} + v_{id} \tag{4}$$

The first part of Eq. (3) represents the inertia of the previous velocity, the second part is the cognition part that tells us about the personal experience of the particle and the third part represents the cooperation among particles. Acceleration constants c_1, c_2 and inertia weight w are the predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

4 In situ Efficiency Determination

The general block diagram of in situ efficiency determination of induction motor using optimization algorithm is shown in Fig.1 [13].

First, the stator line resistance is measured after shutting down the motor. 5hp, four-pole induction motor is considered as test motor. ‘‘Summation of losses’’ method for efficiency determination is used with the assumption of stray load

losses. The winding arrangement of a star connected motor is shown in Fig.2 and the resistance per phase is calculated as in Eq. (5).

$$r_1 = \frac{r_{1\,line}}{2} \tag{5}$$

Where r_{1line} is stator line resistance and r_1 is stator phase resistance.

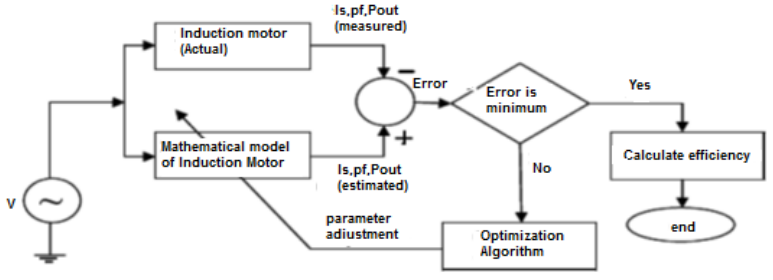


Fig. 1. Block diagram of induction motor in situ efficiency determination

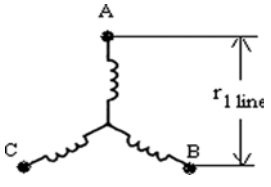


Fig. 2. Winding arrangement of star connected motor

Some measurements on the motor are required before running the optimization algorithm to estimate the motor parameters: stator line to line voltage V_1, stator current I_1, input power P_{inp} and speed at different load points [13].Then, power factor can be calculated as in Eq. (6). The measured and calculated values of the test motor for a wide range of loads and the equivalent circuit considered here is taken from Pillay et al. [14] and is shown in Fig.3.

$$pf = \frac{P_{inp}}{\sqrt{3}V_1 I_1} \tag{6}$$

In the present study we considered stray load loss at full load is 1.8% [IEEE Standard 2004] and its value at different load point is

$$P_{st} = P_{stfl}\frac{I_2^2}{I_{2\,fl}^2} \tag{7}$$

Where P_{st}, P_{stfl} are stray load losses at any point and full load respectively and I_2, I_{2fl} are rotor currents at these load points.

The stray load loss resistance r_{st} is

$$r_{st} = 0.018 r_2 \frac{(1 - s_{fl})}{s_{fl}} \tag{8}$$

Temperatures of stator and rotor windings are assumed to be the same and calculated as in Eq. (9) with the IEEE recommended reference temperature.

$$T_t = \frac{I_1 - I_0}{I_{fl} - I_0}(T_r - T_s) + T_s \tag{9}$$

Where I_1, I_{fl} are the measured and nameplate stator currents, I_0 is the stator current under no-load DC test, T_r=75°C is the reference temperature for the insulation system of class A and T_s=25°C is the ambient temperature.

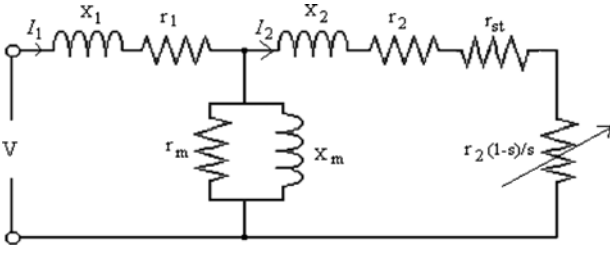


Fig. 3. Equivalent circuit of induction motor with stray loss resistance

The stator and rotor resistances are corrected to the test temperature as:

$$r_{1c} = \frac{r_1(T_t + k_c)}{T_s + k_c} \qquad r_{2c} = \frac{r_2(T_t + k_a)}{T_s + k_a} \tag{10}$$

Where r_1 is the stator resistance measured during DC test and r_2 is the assumed rotor resistance.

The complex admittances of the branches of the equivalent circuit of Fig.3 are given below [14].

$$\overline{Y}_2 = \frac{1}{r_{2c}/s + r_{st} + jx_2} \qquad \overline{Y}_m = \frac{-j}{x_m} + \frac{1}{r_m} \tag{11}$$

The stator current I_1, rotor current I_2, current through r_m (I_m), input power, output power and efficiency can be estimated as

$$I_{1est} = |\overline{I}_1| = \left| \frac{\overline{V}_1 \overline{Y}_1 (\overline{Y}_2 + \overline{Y}_m)}{\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_m} \right| \tag{12}$$

$$I_2 = \left| \frac{\overline{V}_1 \overline{Y}_1 \overline{Y}_2}{\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_m} \right| \tag{13}$$

$$I_m = \left| \frac{\overline{V}_1 \overline{Y}_1}{r_m(\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_m)} \right| \tag{14}$$

$$P_{inpest} = 3(I_1^2 r_{1c} + I_2^2 (r_{2c}/s + r_{st}) + I_m^2 r_m) \tag{15}$$

$$P_{outest} = 3I_2^2 r_{2c} \frac{1-s}{s} \tag{16}$$

$$\eta = \frac{P_{outest}}{P_{\text{inf } est}} 100\% \tag{17}$$

In the present study, two methods of objective functions are considered and are taken from [13]. In method 1, the objective function is

$$\text{Max } ff_1 = \frac{1}{f_1^2 + f_2^2} \tag{18}$$

Where $f_1 = (I_{1est} - I_1)100/I_1$ and $f_2 = (P_{inest} - P_{in})\ 100/P_{in}$

In method 2, the objective function includes stator current, input power, and power factor which is

$$\text{Max } ff_2 = \frac{1}{f_1^2 + f_2^2 + f_3^2} \tag{19}$$

where *f1, f2* are as same as in objection function *ff1* and $f_3 = (pf_{est} - pf)\ 100/pf$

The decision variables of the above objective functions are x_1, r_2, x_m and r_m. Optimization algorithm is used to determine the above said unknown variables.

5 Experimental Setup and Parameter Settings

5.1 Experimental Setup

The experimental setup for testing induction machines up to 100 kW is shown in Fig.4. The machine is loaded by a DC motor equipped with a four quadrant rectifier feeding the energy back to the supply. The input power, input voltages and currents are measured using voltage and non-contacting current probes. Although the measurement data are the same, the efficiency at different standards differs substantially due to different additional losses accounted by each method. Due to deviation in supply frequency, full load speed is affected.

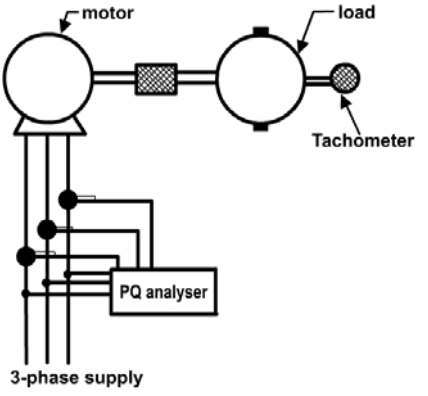


Fig. 4. Experimental setup for determining induction motor efficiency

5.2 Parameter Settings

For the PSO algorithm the inertial weight w is taken to be linearly decreasing from 0.9 to 0.4 and the acceleration constants c_1 and c_2 are taken as 1.49 each. The assigned parameters of the given motor are K_a=225, K_c=234.5, r_1=4.69 ohm, I_O (No-load current) = 1.9 A, Supply frequency=50 Hz, Poles=4.

Table 1. Voltage, current, power input, speed and power factor at various load points

Load Points	Voltage input(V)	Power input(W)	Line current(A)	Actual speed(rpm)	Power factor
1	425	1460	3.4	1432	0.58
2	425	2100	4	1428	0.7
3	425	3090	5.4	1402	0.8
4	425	3900	6.5	1392	0.84

Table 2. Efficiency calculation of induction motor by IEEE

Load Points	Constant loss(W)	Slip	St Cu Loss(W)	Ro Cu Loss(W)	Stray loss(W)	Power Out(W)	Efficiency (%)
1	223	0.0202	54.216	23.954	20.859	1137.971	77.943
2	223	0.0229	75.040	29.096	31.912	1740.953	82.902
3	223	0.0408	136.760	111.332	47.140	2571.768	83.229
4	223	0.0476	198.152	165.659	59.637	3253.551	83.424

Table 3. Efficiency calculation of induction motor by IEC

Load Points	Costant loss(W)	Slip	St Cu Loss (W)	Ro Cu Loss (W)	Addl loss (W)	Power Out(W)	Effi-ciency (%)
1	223	0.0202	54.216	23.954	7.3	1151.53	78.872
2	223	0.0229	75.040	29.096	10.5	1762.364	83.922
3	223	0.0408	136.760	111.332	15.45	2603.458	84.254
4	223	0.0476	198.152	165.659	19.5	3293.688	84.454

Table 4. Efficiency calculation of induction motor by JEC

Load Points	Constant loss(W)	Slip	St Cu Loss(W)	Ro Cu Loss(W)	Power Out(W)	Efficiency (%)
1	223	0.0202	54.216	23.954	1158.83	79.372
2	223	0.0229	75.040	29.096	1772.864	84.422
3	223	0.0408	136.760	111.331	2618.908	84.754
4	223	0.0476	198.152	165.659	3313.188	84.954

Table 5. Error calculation for different standards using objective functions ff_1, ff_2

Load Points	Estimated Eff. (%)		Error w.r.t. ff_1			Error w.r.t. ff_2		
	ff1	ff2	IEEE	IEC	JEC	IEEE	IEC	JEC
1	81.957	81.625	4.014	3.085	2.585	3.682	2.753	2.253
2	81.579	81.377	-1.323	-2.343	-2.843	-1.525	-2.545	-3.045
3	77.207	77.306	-6.022	-7.047	-7.547	-5.923	-6.948	-7.448
4	74.513	74.678	-8.911	-9.941	-10.441	-8.746	-9.776	-10.276

6 Results and Discussion

The efficiency calculation for different standards namely IEEE, IEC and JEC are shown in Tables 2 to 4. Table 5 shows the error calculation for different standards using objective functions ff_1 and ff_2. Figures 5(i) and 5(ii) give a comparison of measured and estimated line currents and power factor respectively. Measured efficiencies (IEEE, IEC, and JEC) and estimated efficiencies (ff_1, ff_2) are plotted for different load points in Figure 6. Figure 7(i) shows variation of error in efficiency with load for different standards using objective function ff_1. From the results, it can be observed that at load point 1, minimum error is obtained using JEC standard. Similarly, at load point 2, 3 and 4 minimum error is obtained using IEEE standard. Similar trend is followed in case of objective functions ff_2 shown in

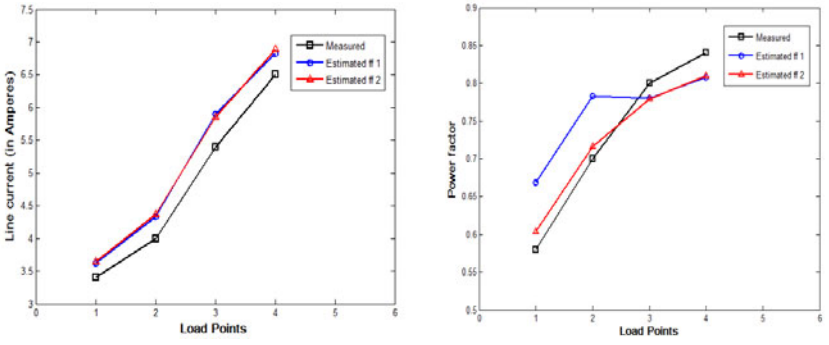


Fig. 5. Comparison of measured and estimated (i) line currents and (ii) power factor of 3.74 KW induction motor respectively

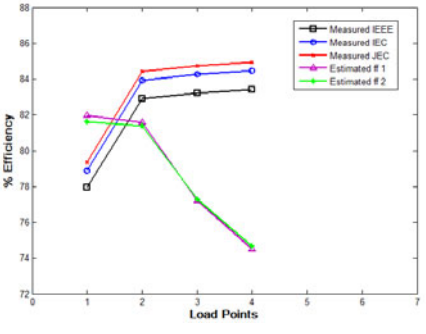


Fig. 6. Comparison of measured efficiency and estimated efficiencies of 3.74 KW induction motor

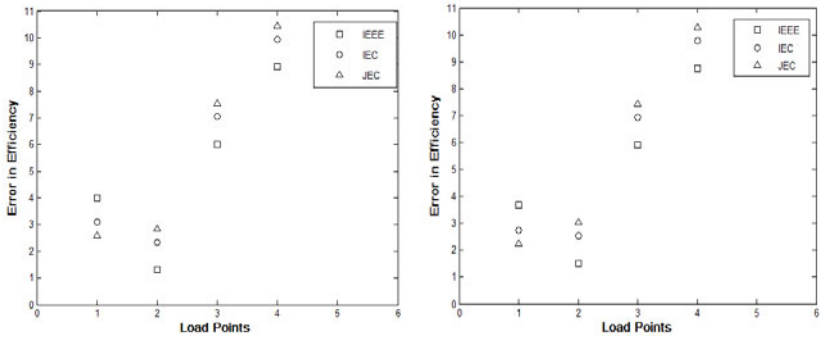


Fig. 7. Comparison of error in efficiency for different standards and at different load points for objective function (i) ff_1 and (ii) ff_2.

Figure 7(ii) i.e. at load point 1, error is minimum for JEC and at load points 2, 3 and 4 error is minimum for IEEE. From Table 5, it can be seen that for all the three standards, error is minimum in case of ff_2. So, only three input parameters

namely stator current, input power and power factor are sufficient to determine the motor parameters quickly.

7 Conclusion

In this paper, particle swarm optimization is used for in-situ efficiency determination of induction motor (5 hp) without performing no-load test. Results are compared with actual values. The error between estimated and actual efficiencies is found for different methods. The differences in methods were based on the number of input parameters used in the optimization algorithm. In-situ efficiency of induction motor can be accurately calculated by using input power, current and power factor as the input parameters of optimization algorithm. Also, IEEE standard leads to minimum error compared to other standards, namely IEC and JEC.

References

[1] Nasir Uddin, M., Nam, S.W.: Adaptive back stepping based online loss minimisation control of an IM drive. IEEE Trans. Power Electronics 23(2), 926 (2008)
[2] Liu, J., Sun, J., Xu, W.: Quantum-Behaved Particle Swarm Optimization with Adaptive Mutation Operator. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006, Part I. LNCS, vol. 4221, pp. 959–967. Springer, Heidelberg (2006)
[3] Bounekhla, M., Zaim, M.E., Rezzoug, A.: Comparative study of three minimization methods applied to the induction machine parameters identification using transient stator current. Electric Power Components and Systems 33, 913–930 (2005)
[4] Renier, B., Hameyer, K., Belmans, R.: Comparison of standards for determining efficiency of three phase induction motors. IEEE Journal (1999)
[5] Nangsue, P., Pillay, P., Conry, S.E.: Evolutionary algorithms for induction motor parameter determination. IEEE Trans. Energy Conversion 14(3), 447–453 (1999)
[6] Benaidja, N., Khenfer, N.: Identification of asynchronous machine parameters by evolutionary techniques. Electric Power Components and Systems 34, 1359–1376 (2006)
[7] Ursem, R.K., Vadstrup, P.: Parameter identification of induction motors using differential evolution. In: Proc. IEEE Congress on Evolutionary Computation, pp. 790–796. IEEE Press, New Jersey (2003)
[8] Ursem, R.K., Vadstrup, P.: Parameter identification of induction motors using stochastic optimization algorithms. Applied Soft Computing 4(1), 49–64 (2004)
[9] IEEE Standard. IEEE Power Engineering Society, IEEE standard test procedure for poly phase induction motors and generators, IEEE Std: 112-2004 (November 4, 2004)
[10] Auinger, H.: Determination and designation of the efficiency of electrical machines. Power Engineering Journal (1999)
[11] Boglietti, A., Cavagnino, A., Lazzari, M., Pastorelli, M.: Induction Motor Efficiency Measurements in accordance to IEEE 112-B, IEC 34-2 and JEC-37 International Standards. IEEE Journal (2003)

[12] Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
[13] Thangaraj, R., Thanga Raj, C., Pant, M., Nagar, A.K.: In-Situ Efficiency Determination of Induction Motor: A Comparative Study of Evolutionary Techniques. Application of Artificial Intelligence 25(2), 116–140 (2011)
[14] Pillay, P., Levin, V., Otaduy, P., Kueck, J.: In situ induction motor efficiency determination using genetic algorithm. IEEE Trans. Energy Conversion 13(4), 326–333 (1998)

Successive Feed-Forward Neural Network for Learning Fuzzy Decision Tree

Manu Pratap Singh and Rajesh Lavania

Department of Computer Science, IET, Dr. B. R. Ambedkar University, Khandari Campus, Agra, Uttar Pradesh, India
manu_p_singh@hotmail.com, rl1304@rediffmail.com

Abstract. Fuzzy decision trees have been substantiated to be a valuable tool and more efficient than neural networks for pattern recognition task due to some facts like computation in making decisions are simpler and important features can be selected automatically during the design process. Here we present a feed forward neural network which learns fuzzy decision trees during the descent along the branches for its classification. Every decision instances of decision tree are represented by a node in neural network. The neural network provides the degree of membership of each possible move to the fuzzy set $<< \ good \ \ move \ >>$ corresponding to each decision instance. These fuzzy values constitute the core of the probability of selecting the move out of the set of the children of the current node. This results in a natural way for driving the sharp discrete-state process running along the decision tree by means of incremental methods on the continuous-valued parameters of the neural network. A simulation program in C has been deliberated and developed for analyzing the consequences. The effectiveness of the learning process is tested through experiments with three real-world classification problems.

Keywords: Decision tree, pattern classification, fuzzy system, artificial neural networks, fuzzy logic.

1 Introduction

Today, in the mass storage era, knowledge acquisition represents a major knowledge engineering bottleneck. Computer programs extracting knowledge from data successfully attempt to alleviate this problem. Among such, systems inducing symbolic decision trees [1-4] for decision making, or classification [5] are very popular. The resulting knowledge, in the form of decision trees [2] and inference procedures [6], has been praised for comprehensibility. This appeal to a wide range of users who are interested in domain understanding, classification capabilities, or the symbolic rules that may be extracted from the tree and

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 701–714.
springerlink.com

subsequently used in a rule-based decision system [7-8]. The humans effortlessly recognize and distinguish objects right from the natural scenario. The human brain intercepts imprecise and incomplete sensory information provided by perceptive organs [5-6]. They can perceive the grossly distorted, ambiguous, noisy or fuzzy patterns [9]. They are also capable to distinguish overlapped patterns. Human performs the entire decision-processing task with biological neural networks [10], which consists of the interconnection of neurons in an immensely intricate fashion. Artificial neural network [6], simulated structure for this interconnection, is capable for pattern recognition [12] task after proper learning of stimuli's. But they are inadequate while classifying fuzzy information [13-14] consisting linguistic rules [15]. Artificial neural networks [6] so as fuzzy logic are dealing with important aspects of knowledge representation, reasoning and learning, but in different approaches with their advantages and weaknesses. Artificial neural networks [6] can learn and classify information from the examples, but it is nearly impossible to describe knowledge acquired in that way. On the other hand, fuzzy logic which enables approximate reasoning [11] has a structural knowledge representation in the form of fuzzy if-then rules [15] but lacks the adaptability to deal with changing external environments. Thus, we incorporate neural network learning concepts in fuzzy inference systems, resulting in fuzzy-neural networks (FNN) with their greatest application in implementation of classification of fuzzy information.

It is well known that decision trees (DT) are the operational support of non-deterministic computations and also very efficient for pattern recognition task [12]. Actually, as long as pattern recognition is considered, a DT can be considered as more efficient than a neural network (NN). There are mainly two reasons, first the computations in making decisions are simpler – only one feature is used in each non-terminal node, and the only computation can be a very simple comparison (say, $x_i < a$). Second, important features can be selected automatically during the design process. In using an NN, since we do not know which feature is important, the only thing we can do is to use all features. However, the DTs do not have the adaptive or learning ability, and thus they cannot be used in changing environment. This problem can be avoided if we map a DT to neural network. Actually there is a very simple mapping from DT to neural network. This mapping integrates the symbolic approach (DT) and sub-symbolic one (neural network). Specifically, this makes DTs adaptable, and at the same time, provides a systematic approach for structural learning of DTs. Learning decision trees is a key problem in computational learning since it is at the borderline between feasible and unfeasible tasks, depending on the assumed ancillary hypotheses. In the connectionist framework we distinguish two main approaches to exploit decision trees.

In the stream of the Hopfield and Tank [17] approach for the resolution of combinatorial problems, in Saylor and Stork [18] a special dynamic neural network is proposed which has the best decision path coded in its fixed point. In the stream of learning from examples to select the appropriate paths over a given decision tree, various architectural solutions and learning algorithms have been proposed [16]. Error back-propagation is the most used training algorithm, but

other algorithms like the Widrow-Hoff learning rule are employed [19]. The use of backpropagation learning rule in neuro-fuzzy system to train fuzzy rules for the classification problem has been already consider in many real world applications [23].

Our learning approach follows the third architectural solution and uses simple backpropagation, with the following distinguishing features: (1) the same feed forward neural network is deputed to make decisions on each node. (2) The feed forward neural network is not a plain external classifier for the children of a given node, but is coupled with the decision tree in a tight way and (3) the training algorithm is especially devised to properly back propagate a particular error function along the branches of the decision tree . The key thought of presented approach is to train a feed forward neural network to output the degree of membership [9] to the fuzzy set $<< \textit{good move} >>$ in correspondence to an input move. Here each neuron is represented as a decision node capable of making decisions based on inputs. One important aspect of the proposed approach is any neuron can be replaced with a full feed forward neural network if a single neuron is not capable of making decisions. We have used a feed forward neural network in place of each neuron represented. The learning procedure we propose offers a set of operational options concerning the objective function to be minimized and the decision tree visiting methods. Conceptually, a feed forward neural network is actually a static mapping between its input and output spaces; this mapping may be either a simple linear relationship of a highly nonlinear one, depending on the network structure. The output of a feed forward neural network plays a double role in incremental training methods: locally, it is the primer for subsequent states; globally, it is the input to the error function. This double role becomes extremely critical in the essentially discrete dynamics of our neural network, with the risk of getting the training process stuck in the meaningless fixed points. To avoid this drawback we give to the fuzzy values returned by the network the general meaning of conditional probability - after proper normalizations - of preferring one move among the available alternatives. There is one more issue pertaining related to generation of decision tree from real time information. For generation of decision trees, various algorithms have been proposed [20] and successfully worn.

The proposed learning procedure used for decision trees using simple backpropagation has been tested over well-known real time statistics i.e. IRIS [21], image segmentation [22], Postoperative Patient Data. Initially the classification for both the problems is performed with backpropagation algorithm. The results obtained exhibit the inadequacy of backpropagation algorithm for classification. Results exhibit that the simple artificial neural network is not able to generate adequate classification due to presence of fuzzy information. The adequate classification can be achieved using proposed procedure. The procedure envisages superior consequences in contrast with the feed forward neural network. The next section discusses the methodology and design of the problem. The experimental analysis and results have been shown in section 3. A brief discussion is presented in section 4. The Section 5 concludes this paper with a summary, the conclusions of this study.

2 Methodology and Simulation Design

The architecture of presented neural network system is extended from the multilayer feed forward neural network. At the lowest level, every decision instances of decision tree are represented by a node in neural network as shown in figure 1. The neural network whose inputs are the branches of the decision tree and whose outputs are the corresponding preference scores i.e. degree of membership. Since we presume that the branches of the decision tree do not look mutually independent to the neural network and we ask it sequentially node by node. The neural network system consists of the components of a conventional neural network system except that computation of degree of membership for each decision instance is performed by each neuron and the neural network's learning capacity is provided to enhance the system knowledge. Here we are discussing the mathematical foundation of both the systems i.e. simple backpropagation neural network and presented neural network system for the valuation.

2.1 Simple Neural Network Architecture

The input-output stimuli's for a particular data set are trained with the neural network has three layers: one input layer, one output layer and a combination of hidden layer(s). Classification of fuzzy information can't be accomplished precisely with the help of conventional artificial neural network architecture.

In backpropagation training algorithm, an input pattern vector ***P*** having ***n*** features as $P_1, P_2, P_3, ...P_n$. Classification of these patterns will be in M classes having the output pattern respectively $C_1, C_2, C_3, ..., C_M$.

Output layer neuron's activation A_K^O and output function O_K^O could be specified as follows,

$$A_K^O = \sum_{i=0}^{H} w_{io_k} O_i^h \tag{2.1.1}$$

$$O_k^o = f\left(A_K^O\right) = f\left(\sum_{i=0}^{H} w_{io_k} O_i^h\right) \tag{2.1.2}$$

Now similarly, the activation and output value for the neurons of hidden layers can be written as,

$$A_K^h = \sum_{i=0}^{N} w_{ik} O_i^I \tag{2.1.3}$$

and output value of input layer neurons

$$O_k^i = f\left(A_k^i\right) \tag{2.1.4}$$

In the backpropagation learning algorithm the change in weight vector is being done according to the calculated error in the network, after iterative training. The error and change in weights in the network can be calculated as,

$$\Delta w_{ho}(n+1) = -\eta \sum_{i=1}^{H} \frac{\partial E}{\partial w_{ho}} + \alpha \Delta w_{ho}(n) \tag{2.1.5}$$

$$\Delta w_{ih}(n+1) = -\eta \sum_{i=1}^{N} \frac{\partial E}{\partial w_{ih}} + \alpha \Delta w_{ih}(n) \tag{2.1.6}$$

$$E^p = \frac{1}{2} \sum_{m=1}^{M} \left(C_m^p - O_m^{p^o} \right)^2 \tag{2.1.7}$$

for $m = 1$ to M output pattern features and $p = 1$ to P presented input patterns and $\left(C_m^p - O_m^{p^o}\right)^2$ is the squared difference between the actual output value of output layer for pattern P and the target output value.

2.2 Representation of Decision Tree over Neural Network

A decision tree is constructed from a training set, which consists of objects. Each object is completely described by a set of attributes and a class label. Attributes can have ordered or unordered values. The concept underlying a data set is the true mapping between the attributes and class. A noise-free training set is one in which all the objects are generated using the underlying concept. A decision tree contains zero or more internal nodes and one or more leaf nodes. All internal

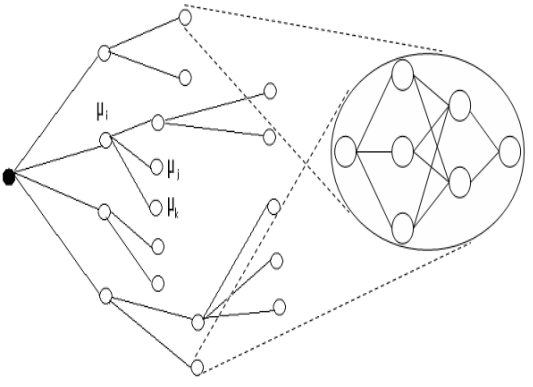


Fig. 1. Descent of the neural advisor for the decision tree. At each node the network outputs a value μ_{ab} . μ_{ab} is the degree of membership for deciding the output of each node. Node is represented by a neural network that learns a decision instance from a decision tree.

nodes have two or more child nodes. All internal nodes contain splits, which test the value of an expression of the attributes. Arcs from an internal node t to its children are labeled with distinct outcomes of the test at t. each leaf node has a class label associated with it. The task of constructing a tree from the training set has been called tree induction, tree building and tree growing. Most existing tree induction systems proceed in a greedy top-down fashion. Starting with an empty tree and the entire training set, some variant of the following algorithm is applied until no more splits are possible.

- If all the training examples at the current node t belong to category c, create a leaf node with the class c.
- Otherwise, score each one of the set of possible splits S, using a goodness measure.
- Choose the best split s* as the test at the current node.
- Create as many child nodes as there are distinct outcomes of s*. Label edges between the parent and child nodes with outcomes of s*, and partition the training data using s* into the child nodes.
- A child node t is said to be pure if all the training samples at t belong to the same class. Repeat all above steps on all impure child nodes.

Discrimination is the process of deriving classification rules from samples of classified objects, and classification is applying the rules to new objects of unknown class. Decision trees have been used for discrimination as well as classification.

In our study, we adopted this representation of fuzzy decision tree as a list of 5-tuples. Each 5-tuples corresponds to a node. There are two kinds of nodes non-terminal & terminal node. Specifically a node is defined by

$$node = \{t, label, P, \mu, size\} \tag{2.2.1}$$

where

- t is the node number. The node ($t = 0$) is called the root.
- Label is the class label of a terminal node, and it is meaningful only for terminal nodes.
- P is pointer to the parent. For root it is NULL.
- μ is degree of membership for suggesting a decision for the next move. Suppose we have ***p*** input-output stimuli, each having ***n*** features and each stimulus belongs to one of ***M*** classes. In fuzzy artificial neural network, the degree of membership for ith pattern (i = *1* to *P* patterns) with the jth class (j = *1* to *M* classes) can be generated as follows,

$$\mu_i^{C_j} = e^{-\frac{1}{2}\left[\frac{P_{ik} - c_k}{\sigma_k}\right]} \tag{2.2.2}$$

where k = *1* to *L* pattern feature. $\mathbf{c_k}$ and $\boldsymbol{\sigma_k}$ are the center and width corresponding to the whole set of patterns for that particular feature across M classes. This method of generating degree of membership is taken from the standard Gaussian membership function (MF). A Gaussian MF is determined completely by **c** and **σ**; **c** represents the MFs center and **σ** determines the MFs width. The figure 2 plots Gaussian membership function *Gaussian (x; 50, 20)*.

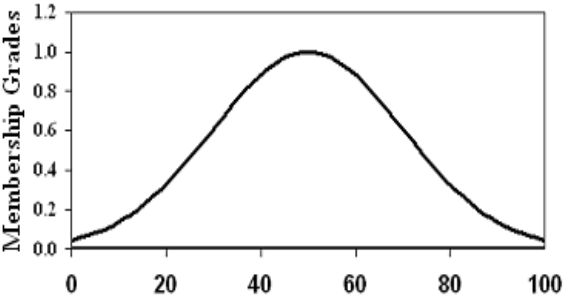


Fig. 2. Gaussian MF

- Size is the size of the node when it is considered a sub-tree. The size of the root is the size of whole tree and of terminal node is 1.

We adopted backpropagation learning algorithm as the learning for our presented approach. Since the neural network has to learn to follow branching decision paths, the whole process must split into two stages:

- On the single tree node, the usual process feeds the inner dynamics of the network.
- Along the branches of the decision tree we have a process which obeys the syntactic rules of the decision process.

Now, in correspondence to each step of the second stage, the error to be minimized must depend in a non-linear way on the switching variables (network outputs at each layer) local to the current node of the decision tree. The method we used for computing the gradient of the error function is forward in both the stages. As discussed, the task of our neural network is to computer the degree of membership for each decision instance. The generated degree of membership will decide the next move. Indeed, among the possible modalities of carrying the decision process out, we followed the approach: at each decision node we select the moves that receive the highest score and continue from there our exploration of the decision tree. Intermediate strategies may plan to follow at each step a limited number of favorite next moves and to take a final decision from the collected paths.

Every node (i^{th} instance of decision tree) will generate degree of membership based on training data. This process will produce a vector $\mathbf{V}(\boldsymbol{p, m})$ of degree of membership corresponding to the relationship between various input-output stimuli's as follows;

$$V = \begin{matrix} \mu_1^{C_1} & \mu_1^{C_2} & \mu_1^{C_3} \ldots & \mu_1^{C_m} \\ \mu_2^{C_1} & \mu_2^{C_2} & \mu_2^{C_3} \ldots & \mu_2^{C_m} \\ \mu_i^{C_1} & \mu_i^{C_2} & \mu_i^{C_3} \ldots & \mu_i^{C_m} \\ \ldots & \ldots & \ldots \ldots & \ldots \\ \mu_p^{C_1} & \mu_p^{C_2} & \mu_p^{C_3} \ldots & \mu_p^{C_m} \end{matrix} \quad (2.2.3)$$

Here $\mu_i^{C_1}, \mu_i^{C_2}, \mu_i^{C_3} \ldots \mu_i^{C_m}$ represents the input pattern vector for training. For classification of this input pattern vector of degree of membership with the fuzzy-neural network system a target output corresponding each input pattern in the form of degree of membership may be defined as follows,

$$\mu_{\max}^{i} = \max \left(\mu_i^{C_1}, \mu_i^{C_2}, \mu_i^{C_3} \ldots \mu_i^{C_m} \right) \quad (2.2.4)$$

This vector of degree of membership will be used as input-output stimuli's for training to the fuzzy-neural network system in support of generating the appropriate classification using the backpropagation algorithm. The error in the network can be calculated as,

$$E^p = \frac{1}{2} \sum_{m=1}^{M} \left(\mu_{\max}^{p} - S_m^{po} \right)^2 \quad (2.2.5)$$

where $\left(\mu_{\max}^{p} - S_m^{po} \right)^2$ is the squared difference between the actual output value of output layer and the target output value in the form of degree of membership.

To minimize the error signal, coupling-strength is updated by an amount proportional to the partial derivative of E^p with respect to w_{ik} (weight between hidden and output layer units)

$$\begin{aligned} \frac{\partial E^p}{\partial w_{ik}} &= \frac{1}{2} \frac{\partial}{\partial w_{ik}} \left[\mu_{\max}^{p} - f\left(\sum_m w_{mk} S_m^{ph} \right) \right]^2 \\ &= \left\{ S_m^{po} - \mu_{\max}^{p} \right\} \dot{f} \left[\sum_m w_{mk} S_m^{ph} \right] S_m^{ph} \\ &= \left(S_m^{po} - \mu_{\max}^{p} \right) S_m^{ph} \left(1 - S_m^{po} \right) S_m^{ph} \end{aligned}$$

$$\frac{\partial E^p}{\partial w_{ik}} = \partial_m^{po} S_m^{ph} \tag{2.2.6}$$

where $\partial_m^{po} = \left(S_m^{po} - \mu_{\max}^p\right) S_m^{ph}\left(1 - S_m^{po}\right)$ and S_m^{ph} is the output from hidden layer. Here ∂_m^{po} is the error term from output layer.

Similarly the partial derivative of E^p with respect to w_{ho} (weight between input and hidden layer units) can be derived as follows,

$$\frac{\partial E^p}{\partial w_{ho}} = \sum_m \left(S_o^i - \mu_{\max}^p\right)\left[S_o^i\left(1 - S_o^i\right)\left\{w_{io} S_o^h\left(1 - S_o^h\right)\right\}\right]\mu_h^{C_i}$$

$$\frac{\partial E}{\partial w_{ho}} = \partial_o^h S_o^i \tag{2.2.7}$$

where $\partial_o^h = \sum_m \left(S_o^i - \mu_{\max}^p\right)\left[\left(1 - S_o^i\right)\left\{w_{io} S_o^h\left(1 - S_o^h\right)\right\}\right]\mu_h^{C_i}$ and this is the error removed from hidden layer.

Change in weights in the network for optimization of weights can be calculated as,

$$\Delta w_{ho}(p+1) = -\eta \sum_{i=1}^{H} \frac{\partial E}{\partial w_{ho}} + \alpha \Delta w_{ho}(p) \tag{2.2.8}$$

$$\Delta w_{ih}(p+1) = -\eta \sum_{i=1}^{N} \frac{\partial E}{\partial w_{ih}} + \alpha \Delta w_{ih}(p) \tag{2.2.9}$$

3 Experiments and Results

In order to consistently validate our method, we performed two experiments for three different sets of data i.e. IRIS, image segmentation and Postoperative Patient Data. First we are attempting to classify both the real-world data with the conventional artificial neural network, later the classification is carried out using the proposed approach of learning decision trees using neural network. First experiment was executed with the varying neural network architectures for generating the possible appropriate classification. Different combinations of hidden layers for artificial neural network have been used for investigating the adequacy of simple neural network. We have chosen three combinations of hidden layers i.e. one, two and three hidden layers. In the second approach, initially we used to generate the decision tree. These decision trees are now mapped to feed forward neural network. A decision instance is represented by a node in the feed forward neural network. A node can be a simple neuron or again a simple

feed forward neural network depending upon the capability to produce degree of membership based on decision tree instance. Here we have used a simple feed forward neural network for a node. This node gives the degree of membership based on the inputs applied to decision trees. Tolerance of neural network has taken for error i.e. ($MAXE \leq 0.001$ or 0.1%). IRIS data contains four fuzzy input constraints to decide classification in three different classes named IRIS Setosa, IRIS Versicolor, and IRIS Virginica. Here we are considering only necessary 30 out of 150 rules for classification. Image segmentation contains 19 input constraints to decide classification in seven different classes named brickface, sky, foliage, cement, window, path and grass. Here we have all 210 training samples for learning of feed forward neural network based on backpropagation. Postoperative Patient data contains 8 input constraints to decide classification in three classes different named I (patient must be sent to Intensive Care Unit), S (patient prepared to go home) and A (patient is out of danger and ready to send to general hospital floor).

3.1 Feed-Forward Neural Network

Here we have performed the experiment for classification of IRIS, image segmentation and Postoperative Patient data with varying hidden layers. We have trained all the sample data for around 100 times. Figure 3 depicts the results of backpropagation learning algorithm. The results clearly show the difficulties of feed-forward neural network for learning and classifying various data.

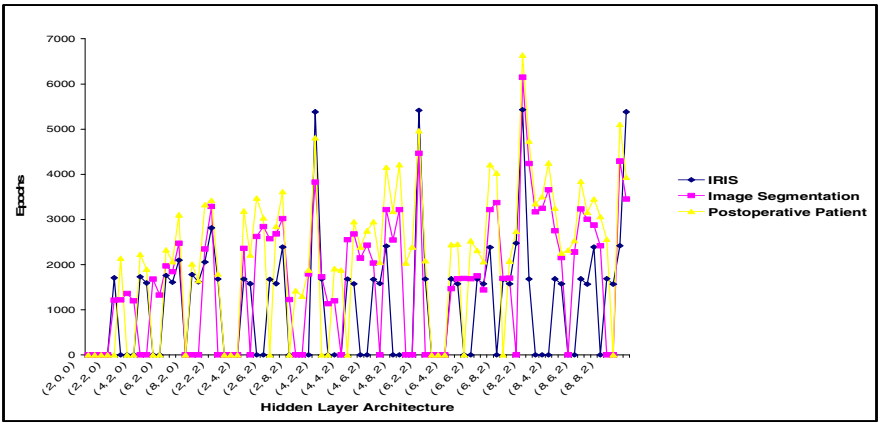


Fig. 3. Comparison of Classification of various data using Backpropagation algorithm

3.2 Fuzzy Decision Tree over Feed Forward Neural Network

These decision trees are mapped to feed forward neural network as in figure-1. A decision instance is represented by a node in the feed forward neural network. A node can be a simple neuron or again a simple feed forward neural network

depending upon the capability to produce degree of membership based on decision tree instance and input parameters. We have used a simple feed forward neural network for a node. Here we have performed the experiment for classification of IRIS, image segmentation and Postoperative Patient data with varying hidden layers inside a node. We have trained all the sample data for around 100 times. Figure 4 depicts the result for the proposed method of neuro fuzzy system. The results clearly show the superiority of presented approach for learning and classifying various data over artificial neural network.

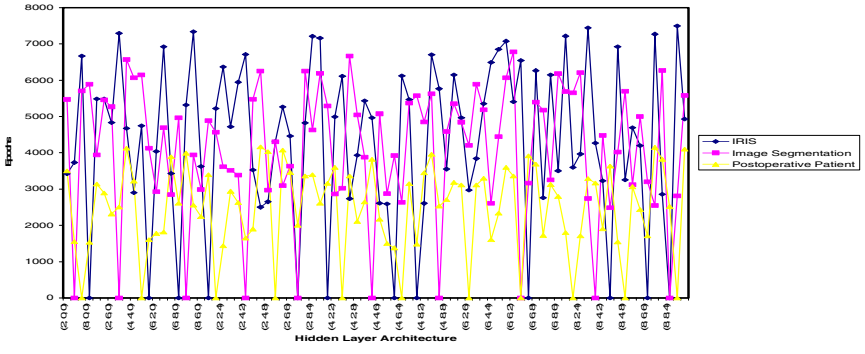


Fig. 4. Comparison of Classification of various data using presented approach

4 Discussion

To test the performance of the presented approach, it has been employed to the real-world problem of IRIS, image segmentation and Post-Operative Patient Data. The proceedings described in this work have some advantages over the standard backpropagation algorithm. The decision tree generated from the real world problems and onwards mapped to primary artificial neural network. This artificial neural network is itself composed of different secondary artificial neural networks in order to train various decision instances rather than standard real features values. Secondary artificial neural network has been used to generate degree of membership based on input parameters provided. The weights of the primary artificial neural network have been allocated based on relationship between various decision instances. Weights of secondary artificial neural network have been assigned randomly initially. After the training procedure weights have been normalized.

In the first experiment we are trying to classify various real world problems using simple backpropagation algorithm of artificial neural networks. In the second experiment, we have generated decision trees based on the real world data. The role of presented approach is to classify the various testing inputs into corresponding approximate output classes based on degree of membership of each decision instance.

In result the term "*Error*" represents the misclassification of input stimuli's. This misclassification representation shows that the conventional artificial neural network is not able to classify with the particular combination of hidden layers. The epochs are representing the convergence of the network. The epochs in decimal number is representing the network convergence up to 20000 iterations with the error representing backpropagation algorithm of artificial neural network has been stuck into local minima. Results indicate the superiority of presented approach over conventional neural network. Figure 3 and 4 shows the comparison study of standard backpropagation algorithm and presented approach. Results indicate that only 52% convergence for IRIS, 66 % convergence for image segmentation and 71% convergence for post-operative patient data, achieved with conventional artificial neural network trained with simple Backpropagation algorithm, while with the presented approach we are able to achieve 86% convergence for IRIS, 88 % convergence for image segmentation and 87% convergence for post-operative patient data. These results can be improved if we increase the level of current iteration i.e. 20000 and maximum error i.e. $MAXE \leq 0.001$.

The results demonstrated that, within the simulation framework presented above, large significant differences exist between the performances of backpropagation feed forward neural network and presented approach for the classification problem of IRIS, image segmentation and Post-Operative Patient data. Results show classification of real world problems performed with the methodology up to having the maximum limit of 20000 iterations (Decimal number indicates error exists after 20000 epochs).

The simulation program, which we have developed in VC++ 6.0 IDE, for testing the adequacy of presented methodology over the data set of IRIS, image segmentation and Post-Operative Patient data, generates the initial weights randomly through its random generator. So the epochs for the algorithms will be different every time with the same network structure and the same training data set. We have chosen the best suitable epochs for designing our results by testing the same training set on the same network structure repeatedly.

5 Conclusion

Taking good decisions in everyday life requires us to exploit formal knowledge seasoned with pieces of personal experience, which makes up for the lack of some symbolic wedge and is affected by the decision maker's style. To follow this strategy, in this paper:

(i) We mapped a decision tree generated from data related real world problems to neural network. The different decision instances are mapped to various nodes in neural network.

(ii) We propose a higher order system, where a single neural network is the switcher of a decision tree. This single neural network consists of various nodes mapped to decision tree instances.

(iii) We have used the backpropagation learning algorithm for generating the degree of membership based on parameters i.e. decision tree instances, input-output stimuli's, where the assessment of a discrete goal variable is driven by an incremental learning process.

Fuzzy-neural networks are based on the integration of two complementary theories i.e. fuzzy logic and artificial neural networks. Purpose of their integration is to compensate weaknesses of one theory with advantages of the other. Based on the analysis of several fuzzy-neural network models, we tried to introduce uniform representation model for classification of fuzzy information. Fuzzy logic facilities reasoning based on incomplete and imprecise information, known as approximate reasoning. On the other hand, artificial neural networks are developed to ape biological neural systems in performing functions such as adaptation, pattern classification, pattern recalling, pattern association and lots of other functionalities. While fuzzy logic enables mechanism for reasoning based on incomplete and imprecise information, artificial neural networks provide some remarkable abilities such as learning, adaptation and generalization.

The results demonstrated that, large significant differences exist between the performances of backpropagation feed forward neural network and presented approach for the classification problem of IRIS, image segmentation and Post-Operative Patient data in the terms of accuracy, convergence and epochs. These results recommend the adequacy of approach for classification. In first experiment i.e. using feed forward neural network, success percentage is quite lower for IRIS, image segmentation and Post-Operative Patient data.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth, Inc., Belmont (1984)
[2] Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
[3] Quinlan, J.R.: Simplifying decision trees. International Journal of Man-Machine Studies 27, 221–234 (1987)
[4] Quinlan, J.R.: Decision trees and decision making. IEEE Transactions on Systems, Man and Cybernetics 20(2), 339–346 (1990)
[5] Abdi, H.: A Neural Network Primer. Journal of Biological Systems 2(3), 247–283 (1994)
[6] Haykin, S.: Neural Networks, 2nd edn. Prentice Hall (1999)
[7] Yoon, Y.O., Brobst, R.W., Bergstresser, P.R., Peterson, L.L.: A desktop neural network for dermatology diagnosis. Journal of Neural Network Computing, 43–52 (Summer 1989)
[8] Lindley, D.V.: The probability Approach to the treatment of Uncertainty in Artificial Intelligence and Expert Systems. Statistical Science 2(1), 17–24 (1987)
[9] Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
[10] Muller, B., Reinhardt, J.: Neural Networks: An Introduction, Physics of Neural Networks. Springer, Heidelberg (1991)
[11] Poggio, T., Girosi, F.: Networks for approximation and learning. The Proceedings of the IEEE 78(9), 1485–1497 (1990)

[12] Schalkoff, R.: Pattern Recognition- Statistical, Structural and Neural Appraoches. John Wiley & Sons, New York (1992)
[13] Bezdek, J.C.: A review of probabilistic, fuzzy and neural models for pattern recognition. In: Fuzzy Logic and Neural Network Handbook. McGraw Hill, USA (1996)
[14] Zadeh, L.A.: Fuzzy logic, neural networks and soft computing. One-page course announcement of CS 294-4, Spring 1993, University of California at Berkeley (1992)
[15] Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Parts 1, 2 & 3, Information Sciences (1975)
[16] Valiant, L.: A theory of the learnable. Communication of ACM 27, 1134–1142 (1984)
[17] Hopfield, J., Tank, D.: Neural computations of decisions in optimization problems. Biological Cybernetics 52(3), 141–152 (1985)
[18] Saylor, J., Stork, D.: Parallel analog neural networks for tree searching. In: Proc. Neural Networks for Computing, pp. 392–397 (1986)
[19] Sethi, I.: Entropy nets- from decision trees to neural networks. Proceedings of the IEEE 78, 1605–1613 (1990)
[20] Abe, S., Lan, M.S.: A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification. IEEE Transactions of Fuzzy Systems 3(1), 18–28 (1995)
[21] Anderson, E.: The Irises of the Gaspe Peninsula. Bulletin America, IRIS Soc. (1935)
[22] Budihardjo, A., Grzymala-Busse, J., Woolery, L.: Program LERS_LB 2.5 as a tool for knowledge acquisition in nursing. In: Proceedings of the 4th Int. Conference on Industrial & Engineering Applications of AI & Expert Systems, pp. 735–740 (1991)
[23] Jain, M., Butey, P.K., Singh, M.P.: Classification of Fuzzy-Based Information using Improved backpropagation algorithm of Artificial Neural Networks. International Journal of Computational Intelligence Research 3(3), 265–273 (2007)

Enhancing Different Phases of Artificial Bee Colony for Continuous Global Optimization Problems

Tarun Kumar Sharma and Millie Pant

Indian Institute of Technology, Roorkee, India
taruniitr1@gmail.com, millifpt@iitr.ernet.in

Abstract. The working of basic ABC algorithm depends on the functioning of three categories of bees; the employed bees, the onlooker bees and the scout bees. Although, employed and onlooker bees have different functionality, they follow the same equation for exploration and exploitation. Obviously, the performance of ABC greatly depends on single equation. In order to provide a variation in the working of ABC, we propose the use of different equations in the employed bee and onlooker bee phase. The new mechanism proposed by us for the movement of the bees depends on the convex linear combination of three candidate solutions. This scheme is initially embedded in the employed bees phase while the original equation is maintained for the onlooker bees. In the second variation the basic equation for employed bees is retained while for onlooker bees, different equation is used. The simulation results demonstrate that the modification increases efficiency and capability in terms of balancing exploration and exploitation as well as the accelerating the convergence rate of the ABC.

Keywords: Artificial Bee Colony, Solution vector, Mutation, Exploitation, Exploration, Convergence.

1 Introduction

Artificial bee colony algorithm (ABC) is a computation technique developed by Karaboga [1] based on simulating the foraging behavior of honey bee swarm. The performance of ABC is competitive to other population-based algorithms with an advantage of employing fewer control parameters [2]–[4]. Due to its simplicity and ease of implementation, ABC has captured much attention and has been applied to solve many practical optimization problems [5]–[7].

In this paper a scheme proposed by [8] is embedded in various phases of ABC to accelerate and balance the exploration and exploitation factors. The proposed modification is incorporated in the structure of ABC, initially for employed bees, and then for onlooker bees. Further, when we observed that the proposed scheme

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 715–724.
springerlink.com

works well for the employed and the onlooker phase we embedded it in the final phase to guide the movement of scout bees. The proposed variants are termed as EABC1, EABC2 and E-Scout respectively.

The rest of the paper is organized as follows: section 2 gives an overview of ABC algorithm. The proposed variants are discussed in section 4. Numerical results, parameter settings and result analysis are given in section 5. Finally the paper concludes with section 6.

2 Artificial Bee Colony

2.1 Initialization of Food Sources (Population)

The initial population of solutions is filled with SN number of randomly generated n-dimensional real-valued vectors (i.e., food sources). Let $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ represent the ith food source in the population, and then each food source is generated by equation (1).

$$x_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{1}$$

Where i = 1,2,..,SN. j = 1,2,…,n. $x_{\max,j}$ and $x_{\min,j}$ are upper and lower bounds of parameter j, respectively. These food sources are randomly assigned to SN number of employed bees and their fitness is evaluated.

2.2 Employed Bee Initialization

In this phase each employed bee X_i generates a new food source V_i in the neighborhood of its present position by using solution search equation (2). Once V_i is obtained, it is evaluated and compared to X_i. If the fitness of V_i is equal to or better than that of X_i, V_i will replace X_i and become a new member of the population; otherwise X_i is retained. In other words, a greedy selection mechanism is employed between the old and candidate solutions

$$v_{ij} = x_{ij} + \phi_{ij}(x_{i,j} - x_{k,j}) \tag{2}$$

where $i, k = 1,...,SN, j = 1,...,n,$ and v_i is the new food source generated by using both, the current food source x_i and a randomly chosen food source x_k from the population and $-1 \le \phi_{ij} \le 1$ (generated at random every time it is used) determines the step size of the movement. Both, i and j are generated at random but $k \neq i$.

2.3 Probabilistic Selection

An important stage of the ABC algorithm, from which in fact the collective intelligence arises, is the sharing of information. This is achieved by influencing the behavior of onlookers which will select their food source according to probability equation (3).

$$P_i = f_i / \sum_{k=1}^{SN} f_k \tag{3}$$

where f_i is the fitness value of a i, food source (position in parameter space). In other words onlookers will explore promising locations with higher probability than others.

2.4 Onlooker Bee Phase

An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source X_i depending on its probability value P_i. Once the onlooker has selected her food source X_i, she produces a modification on X_i by using Equation (2). As in the case of the employed bees, if the modified food source has a better or equal nectar amount than X_i, the modified food source will replace X_i and become a new member in the population.

2.5 Scout Bee Phase

If a food source X_i cannot be further improved through a prespecified number of trials limit, the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. The scout produces a food source randomly using equation (1).

3 Proposed Schemes

The objective of this study is to enhance the working of basic ABC by producing efficient food sources which in terms of ABC implies to the solution vectors having better fitness value. Also, we aim to develop a scheme which is able to exploit the local information efficiently and is easy to apply.

One of the simplest ways to search the solution space can be done by linear exploitation. In the proposed study, two schemes are considered. In the first scheme the new solution vector (food source) is generated by taking arithmetic mean of the convex linear combination of three mutually exclusive solution vectors.
The convex linear combination of three vectors is given as:

$$V_0 = \sum_{i=1}^{3} \lambda_i X_{r1,G}$$

where the weights, $\lambda_i \geq 0$ and $\sum_{i=1}^{3} \lambda_i = 1$. From this general solution vector definition the following special case was derived by [8].

Arithmetic Mean of solution V_0

$$V_1 = \frac{1}{3}\sum_{i=1}^{3} \lambda_i X_{r1,G} \tag{4}$$

The aim of this modification is to accelerate the convergence rate and to determine a solution vector that can comprise the local information of all members in the triplet. In the search space, the triplet of individuals selected can form a trigonometric hyper plane having the three member individuals as its vertices. The solution vector determined by the scheme will lie (or lie with a high probability) within this trigonometric hyper plane. A two-dimensional optimization problem is taken by [8] as an example for providing an easy-to-understand description. In Figure 1, it is assumed that the triplet selected for mutation consists of the individuals $X_{r1,G}$, $X_{r1,G}$, $X_{r1,G}$. From the formulation (4), it can easily be deduced that the larger the weight that a member of the triplet has, the closer it "attracts" the solution vector to itself. For the arithmetic mean case, the new solution vector is simply situated at the geometric center of the trigonometric plane.

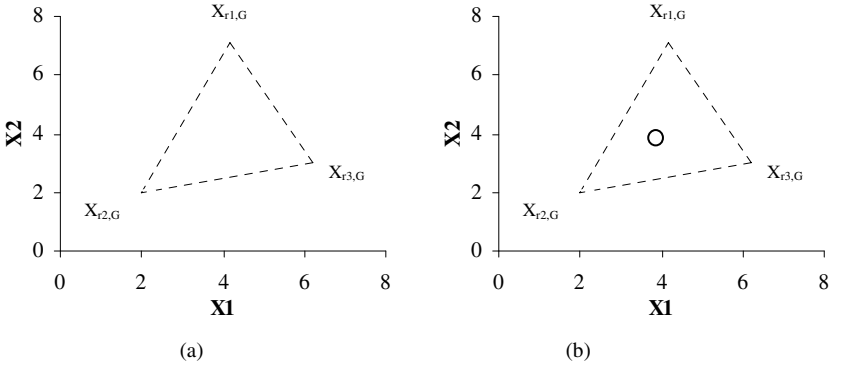


Fig. 1. A geometric presentation of the mutation donor scheme with a two-dimensional optimization problem (a) donor '0' (b) donor '1'

4 Proposed Variants

In ABC, the employed bees and onlooker bees carry out exploration and exploitation using the same equation. Obviously, the performance of ABC greatly depends on single equation. Also, in the scout bee phase the new location is generated randomly. In the present study we embed the scheme discussed in the previous section one-by-one on different phases of ABC

- Modification in Employed Bee Phase - EABC1
- Modification in Onlooker Bee Phase -EABC2 and
- Modification in Scout Bee Phase - E-Scout

The equation used in different phases of ABC to generate the new food locations is given as:

$$v_{ij} = (x_{r1,j} + x_{r2,j} + x_{r3,j})/3 + \phi_{ij}(x_{i,j} - x_{k,j}) \quad (5)$$

where r1≠r2≠ r3≠i∈ [1, SN] are selected randomly.

After initialization the variants starts like the usual ABC as discussed in previous section. In EABC1 the employed bee phase uses Equation (5) to generate new food locations and rest of the structure of ABC remain same. Similarly in EABC2, Onlooker uses Equation (5) and rest the structure of ABC. In case of E-Scout, the food sources become abandoned and new food sources are generated using Equation (5), keeping the basic ABC structure intact.

The description of the proposed variants are demonstrated in Figure 2 as a highlighted text.

1. **Begin**
2. Initialize the set of food sources (Eq.(1)) x_i, i=1,...,SN
3. Evaluate each x_i , i = 1,…,SN
4. g = 1
5. **Repeat**
6. **For** i = 1 **to** SN
7. Generate $v_{i,g}$ with $x_{i,g-1}$ using Eq(2) *for ABC, EABC2, E-Scout*
   Eq(5) *for EABC1*
8. Evaluate $v_{i,g}$
9. **If** $v_{i,g} < x_{i,g-1}$
10. $x_{i,g} = v_{i,g}$
11. **Else**
12. $x_{i,g} = x_{i,g-1}$
13. **End If**
14. **End For**
15. **For** i = 1 **to** SN
16. Select, based on fitness proportional selection food source $x_{i,g}$
17. Generate $v_{i,g}$ with $x_{i,g}$ by Eq(2) *for ABC, EABC1, E-Scout*
    Eq(5) *for EABC2*
18. Evaluate $v_{i,g}$
19. **If** $v_{i,g} < x_{i,g}$
20. $x_{i,g} = v_{i,g}$
21. **End If**
22. **End For**
23. Generate new food sources at random for those whose limit to be improved has been reached using Eq(1) *for ABC, EABC1, and EABC2*
    Eq(5) *for E-Scout*
24. Keep the best solution so far
25. $g = g + 1$
26. **Until** $g = MCN$
27. **End**

Fig. 2. Pseudocode of the proposed variants

5 Numerical Simulations and Comparisons

5.1 Experimental Setting

- The main parameters of ABC are: the colony size (SN), MCN (Maximum Cycle Numbers) and "*limit*" which have been carefully fine tuned after conducting a series of experiments.
- All algorithms are implemented in Dev-C++ and the experiments are conducted on a computer with 2.00 GHz Intel (R) core (TM) 2 duo CPU and 2-GB of RAM.
- In order to make a fair comparison of ABC and the proposed algorithm, we fixed the same seed for random number generation so that the initial population is same for all the algorithms.
- For each problem, the ABC and EABC variants are independently run 30 times.
- The parameter setting is taken as follows:

Population size	*100 (i.e SN=50)*
limit	*100*
Value to Reach (VTR)	10^{-15}
Maximum MCN	*10000*

5.2 Performance Criteria

The average of function fitness value that an algorithm can find, using predefined MCN, is recorded in each run and then average of the function fitness values is calculated. Also the average, standard deviation, best and worst of the fitness values are calculated.

We set the termination criteria as $\left|f_{optimal} - f_{global}\right| \leq VTR$ and recorded MCN over 30 runs (only for engineering problems).

The proposed variants are evaluated on 5 well-known benchmark problems [9] and three engineering design problems [10]-[12] given in the appendix.

5.3 Simulation Results

In Table-1 we have taken the results on the basis of average error. In this case MCN is fixed at 10000, to estimate the average of minimum fitness function value in 30 runs.

From the Table-1 it can be clearly observed that for 3 benchmark functions EABC2 gives better results than ABC, EABC1 and E-Scout algorithms and in case of Ackley ABC emerges as winner where as in case of Rosenbrock E-Scout is winner. From Table 2, it can be observed that the EABC2 algorithm costs less computational effort than the basic ABC, EABC1 and E-Scout algorithms However, all the algorithms suffered from premature convergence, since they did not achieve the absolute accuracy level in all runs with Rosenbrock function. In case

of engineering design problems again EABC2 outperformed ABC and its other variants, which can be clearly observed from Tables 3 & 4 respectively. The graphical representation in terms of MCN for both the type of problems considered in the paper i.e. benchmark and engineering problems are shown in Figure 3 (a) & (b) respectively.

Table 1. Comparison of the EABC1, EABC2, E-SCOUT and ABC Algorithms *D* =30 & Population Size=100

F	Stat.	ABC	EABC1	EABC2	E-Scout
Sphere	mean	2.27e-017	5.23e-017	**2.64e-017**	3.59e-017
	std.d.	6.46e-017	1.49e-016	**7.51e-017**	1.02e-016
Ackley	mean	**1.99e+000**	3.57e-015	2.50e-015	3.50e-015
	std.d.	**5.68e+000**	1.01e-014	7.11e-015	1.00e-014
Rosenbrock	mean	4.81e-001	2.04e-001	1.35e-001	**1.93e-002**
	std.d.	1.37e+000	5.80e-001	3.86e-001	**5.50e-002**
Griewank	mean	2.66e-017	2.72e-017	**2.52e-017**	2.75e-017
	std.d.	7.58e-017	7.74e-017	**7.19e-017**	7.83e-017
Rastrigin	mean	2.428e-018	1.82e-018	**1.64e-018**	1.99e-018
	std.d.	6.912e-018	5.18e-018	**4.69e-018**	5.67e-018

Table 2. Comparison of the, EABC2, E-SCOUT and ABC Algorithms D =30 & Population Size=100 in terms of MCN to VTR

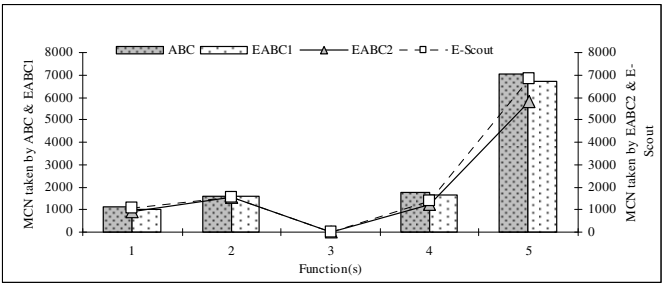
F	ABC	EABC1	EABC2	E-Scout
Sphere	1107	997	917	1082
Ackley	1598	1604	1521	1552
Rosenbrock	NC	NC	NC	NC
Griewank	1739	1676	1209	1371
Rastrigin	7017	6719	5809	6814

Table 3. Mean of Fitness Function Value and Standard Deviation (STD) of Engineering Problems F_1-F_3

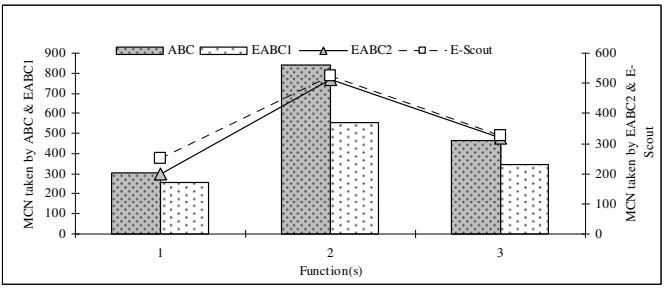
F	D	ABC	EABC1	EABC2	E-Scout
F_1	2	169.849 ±(0.00801)	169.857 ± (1.94.e-012)	169.602 ± (0.00021)	169.842 ± (7.703e-011)
F_2	3	4.21319 ± (3.19e-016)	4.21315 ± (.000031)	4.21201 ± (7.09e-016)	4.21311 ± (2.09e-016)
F_3	4	4.7457e-008 ± (1.87e-008)	2.9014e-008 ± (2.944e-024)	9.6910e-009 ± (9.25e-09)	1.8536e-008 ± (4.08e-010)

Table 4. Comparison of Proposed Algorithms in terms of MCN For Engineering Problems F_1-F_3

F	ABC	EABC1	EABC2	E-Scout
F1	306	256	197	249
F2	838	556	513	524
F3	463	348	317	324



(a)



(b)

Fig. 3. MCN taken by five benchmark functions (a) and engineering problems (b)

6 Conclusion and Future Work

In ABC, the employed bees and onlooker bees carry out exploration and exploitation using the same equation. Obviously, the performance of ABC greatly depends on single equation. This study presents some modification in the original structure of ABC. In order to verify the feasibility and the performance of the proposed algorithms, five benchmark problems and three engineering design problems are tested. From the simulation results it is concluded that EABC2 is better in term of searching quality, and efficiency than ABC and its variant. However, making a concrete judgment at this stage is not correct as the test bed considered is rather narrow. In future we will implement it on a larger test bed with constrained optimization problems.

References

[1] Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06. Erciyes University, Kayseri, Turkey (2005)

[2] Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39, 171–459 (2007)

[3] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)

[4] Karaboga, D., Basturk, B.: A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 214, 108–132 (2009)
[5] Singh, A.: An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied Soft Computing 9, 625–631 (2009)
[6] Kang, F., et al.: Structural inverse analysis by hybrid simplex artificial bee colony algorithms. Computers & Structures 87, 861–870 (2009)
[7] Samrat, L., et al.: Artificial bee colony algorithm for small signal model parameter extraction of MESFET. Engineering Applications of Artificial Intelligence 11, 1573–2916 (2010)
[8] Fan, H.-Y., Lampinen, J., Dulikravich, G.S.: Improvements to the Mutation Donor Formulation of Differential Evolution. In: International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems EUROGEN, pp. 1–12 (2003)
[9] Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local Search. IEEE Transactions on Evolutionary Computation 12(1), 107–125 (2008)
[10] Beightler, C., Phillips, D.: Applied geometric programming. John Wiley and Sons, New York (1976)
[11] Prasad, B., Saini, J.: Optimal thermo hydraulic performance of artificially roughened solar air heater. Journal Solar Energy 47, 91–96 (1991)
[12] Babu, B.: New optimization techniques in engineering. Springer, Heidelberg (2004)

Appendix

List of benchmark problems

Sphere

$$f_1(x) = \sum_{i=1}^{n} x_i^2$$

with -5.12 $\leq x_i \leq$ 5.12, min $f_1(0,\ldots,0) = 0$

Ackley's

$$f_2(x) = -20 * \exp\left(-.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$$

With -32 $\leq x_i \leq$ 32, min $f_2(0,\ldots,0) = 0$

Rosenbrock's

$$f_3(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1}^2 - x_i^2\right) + \left(1 - x_i^2\right)\right]$$

with -30 $\leq x_i \leq$ 30, min $f_3(1,\ldots,1) = 0$

Griekwank

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

with -600 $\leq x_i \leq$ 600, min $f_4(0,\ldots,0) = 0$

Restrigin's

$$f_5 = 10\ n + \sum_{i=1}^{n} \left(x_i^2 - 10\ \cos\ (2\pi x_i) \right)$$

with -5.12 $\leq x_i \leq$ 5.12, min f_5 (0…, 0) = 0

List of engineering design problems

F1: Optimal Capacity of Gas Production Facilities [10]
Minimize:

$$f(x) = 61.8 + 5.72x_1 + 0.2623[(40 - x_1)\ln(\frac{x_2}{200})]^{-0.85} + 0.087(40 - x_1)\ln(\frac{x_2}{200}) + 700.23x_2^{-0.75}$$

Subject to:

$x_1 \geq 17.5$, $x_2 \geq 200$; $17.5 \leq x_1 \leq 40$, $300 \leq x_2 \leq 600$.

F2: Optimal Thermohydraulic Performance of an Artificially Roughened Air Heater [11]
Maximize:

$$L = 2.51 * \ln e^+ + 5.5 - 0.1R_M - G_H$$

where $R_M = 0.95x_2^{0.53}$; $G_H = 4.5(e^+)^{0.28}(0.7)^{0.57}$

$e^+ = x_1 x_3 (\bar{f}/2)^{1/2}$; $\bar{f} = (f_s + f_r)/2$; $f_s = 0.079x_3^{-0.25}$; $f_r = 2(0.95x_3^{0.53} + 2.5 * \ln(1/2x_1)^2 - 3.75)^{-2}$

Subject to:

$0.02 \leq x_1 \leq 0.8$, $10 \leq x_2 \leq 40$, $3000 \leq x_3 \leq 20000$

F3: Design of Gear Train [12]
Minimize:

$$f = \left\{ \frac{1}{6.931} - \frac{T_d T_b}{T_a T_f} \right\}^2 = \left\{ \frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right\}^2$$

Subject to:

$12 \leq x_i \leq 60 \quad i = 1,2,3,4$

$[x_1, x_2, x_3, x_4] = [T_d, T_b, T_a, T_f]$, x_i's should be integers. Ta, Tb, Td, and Tf are the number of teeth on gears A, B, D and F respectively. The design of a compound gear train is shown in Figure 4.

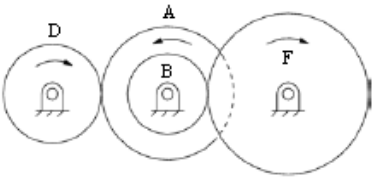


Fig. 4. Compound Gear Train

Analyzing Fuzzy System Reliability Using Arithmetic Operations on Different Types of Intuitionistic Fuzzy Numbers

Mohit Kumar and Shiv Prasad Yadav

Department of Mathematics, I.I.T Roorkee, Roorkee, India
e-mail: msharmadma.iitr@gmail.com, yadavfma@iitr.ernet.in

Abstract. This paper presents a new method for fuzzy system reliability analysis based on arithmetic operations of different types of intuitionistic fuzzy numbers. In this paper, the reliabilities of the components of a system are represented by different types of intuitionistic fuzzy numbers. Because in the proposed method, we allow the reliabilities of the components of a system to have different types of intuitionistic fuzzy numbers, it is more flexible than the ones presented in [12, 13]. Furthermore, because the proposed method uses the simple arithmetic operations of intuitionistic fuzzy numbers based on α-cuts and β-cuts rather than the complicated nonlinear programming techniques mentioned in [17], it is simpler in calculating fuzzy system reliability than the one presented in that paper. Numerical example is also given to illustrate the proposed method.

Keywords: Fuzzy system reliability, intuitionistic fuzzy number, (α, β) -cuts.

1 Introduction

One of the important engineering tasks in design and development of any technical system is reliable engineering. The conventional reliability of a system is defined as the probability that the system performs its assigned function properly during a predefined period under the condition that the system behavior can be fully characterized in the context of probability measures. However, in the real world problems, the collected data or system parameters are often imprecise because of incomplete or non-obtainable information, and the probabilistic approach to the conventional reliability analysis is inadequate to account for such built-in uncertainties in data. To overcome this problem, fuzzy set theory [1] has been used in the evaluation of the reliability of a system. In the work of Cai et al. [2], the fuzzy system reliability was established based on the binary state assumption and possibility assumption. However in the work of Cai et al. [3], the fuzzy system reliability was established based on the three-state assumption and possibility

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 725–736.
springerlink.com

assumption. In the work of Cai et al. [4], the fuzzy system reliability was developed based on the basis of fuzzy state assumption and probability assumption. They presented a fuzzy set based approach to failure rate and reliability analysis, where profust failure rate is defined in the context of statistics. Cai et al. [5] also discussed the system reliability for coherent system based on the fuzzy state assumption and probability assumption.

Singer [6] used a fuzzy set approach for fault tree and reliability analysis in which the relative frequencies of the basic events are considered as fuzzy numbers. Cheng and Mon [7] analyzed fuzzy system reliability analysis by use of interval of confidence. Through theoretical analysis and computational results, they have shown that their proposed approach is more general and straightforward compared to Singer [6].

Chen [8] presented a new method for fuzzy system reliability analysis using fuzzy number arithmetic operations in which reliability of each component is considered as fuzzy number and used simplified fuzzy arithmetic operations rather than complicated interval fuzzy arithmetic operations of fuzzy numbers [7] or the complicated extended algebraic fuzzy numbers [6].

Fuzzy set theory has been shown to be a useful tool to handle such situations by attributing a degree to which a certain object belongs to a set. In real life, a person may assume that an object belongs to a set to a certain degree, but it is possible that he is not so sure about it. In other words, there may be a hesitation or uncertainty about the membership degree of x in A. In fuzzy set theory, there is no means to incorporate that hesitation in the membership degrees. A possible solution was to use intuitionistic fuzzy sets (IFS), defined by Atanassov [9].

The concepts of IFS can be viewed as an appropriate/alternative approach to define a fuzzy set in case where available information is not sufficient for the definition of an imprecise concept by means of a conventional fuzzy set. In fuzzy sets the degree of acceptance is considered only but IFS is characterized by a membership function and a non-membership function so that the sum of both values is less than or equal to one [10]. Presently intuitionistic fuzzy sets are being studied and used in different fields of Science and Engineering. With best of our knowledge, Burillo [11] proposed definition of intuitionistic fuzzy number (IFN) and studied perturbations of IFN and the first properties of the correlation between these numbers. Mahapatra and Roy [12] presented a method for reliability evaluation using triangular IFNs arithmetic operations. Kumar et al. [13] developed a new approach for analyzing the fuzzy system reliability of series and parallel systems using intuitionistic fuzzy set theory.

In this paper, a new method is developed for fuzzy system reliability analysis based on arithmetic operations of intuitionistic fuzzy numbers using α-cuts and β-cuts, where the reliabilities of the components of a system are represented by different types of intuitionistic fuzzy numbers.

2 Intuitionistic Fuzzy Set Theory

Atanassov [10] introduced the concept of an IFS which is characterized by a membership function $\mu_{\tilde{A}^i}(x)$ indicating evidence for $x \in X$ and a non-membership function $\nu_{\tilde{A}^i}(x)$ indicating the evidence against $x \in X$. This idea which is a natural generalization of usual fuzzy set seems to be useful in modeling many real life situations.

2.1 Intuitionistic Fuzzy Set

Let X be a universe of discourse. Then an IFS $\tilde{A}^i$ in X is a set of ordered triples given by

$$\tilde{A}^i = \left\{ < x, \mu_{\tilde{A}^i}(x), \nu_{\tilde{A}^i}(x) > : x \in X \right\},$$

where $\mu_{\tilde{A}^i} : X \to [0,1]$ and $\nu_{\tilde{A}^i} : X \to [0,1]$ are functions such that

$$0 \le \mu_{\tilde{A}^i}(x) + \nu_{\tilde{A}^i}(x) \le 1 \qquad \forall x \in X.$$

For each $x \in X$, the numbers $\mu_{\tilde{A}^i}(x)$ and $\nu_{\tilde{A}^i}(x)$ represent respectively the degree of membership and degree of non-membership of the element $x \in X$ to $\tilde{A}^i \subseteq X$. For each $x \in X$, the intuitionistic fuzzy index of x in $\tilde{A}^i$ is defined as $\pi_{\tilde{A}^i}(x) = 1 - \mu_{\tilde{A}^i}(x) - \nu_{\tilde{A}^i}(x)$.

2.2 Convex Intuitionistic Fuzzy Set

An IFS $\tilde{A}^i$ in the universe of discourse X is convex [14, 15] if and only if

(i) Membership function $\mu_{\tilde{A}^i}(x)$ of $\tilde{A}^i$ is fuzzy-convex i.e.

$$\mu_{\tilde{A}^i}\left(\lambda x_1 + (1-\lambda) x_2\right) \ge \min\left(\mu_{\tilde{A}^i}(x_1), \mu_{\tilde{A}^i}(x_2)\right) \quad \forall x_1, x_2 \in X,\ \lambda \in [0,1],$$

(ii) Non-membership function $\nu_{\tilde{A}^i}(x)$ of $\tilde{A}^i$ is fuzzy-concave i.e.

$$\nu_{\tilde{A}^i}\left(\lambda x_1 + (1-\lambda) x_2\right) \le \max\left(\nu_{\tilde{A}^i}(x_1), \nu_{\tilde{A}^i}(x_2)\right) \quad \forall x_1, x_2 \in X,\ \lambda \in [0,1].$$

2.3 Normal Intuitionistic Fuzzy Set

An IFS $\tilde{A}^i$ in the universe of discourse X is normal [14, 15] if there exits at least two points $x_1, x_2 \in X$ such that $\mu_{\tilde{A}^i}(x_1) = 1$ and $\nu_{\tilde{A}^i}(x_2) = 1$.

2.4 Intuitionistic Fuzzy Number

An IFS $\tilde{A}^i = \{< x, \mu_{\tilde{A}^i}(x), \nu_{\tilde{A}^i}(x) >: x \in R\}$ in the real line R is called an IFN if

(i) $\tilde{A}^i$ is convex and normal.
(ii) $\mu_{\tilde{A}^i}(x)$ is upper semi continuous and $\nu_{\tilde{A}^i}(x)$ is lower semi continuous.
(iii) Supp $\tilde{A}^i = \{x \in X : \nu_{\tilde{A}^i}(x) < 1\}$ is bounded.

2.5 (α, β)-Level Set or (α, β)-Cut of an Intuitionistic Fuzzy Set

(α, β)-cut of IFS $\tilde{A}^i$ is defined as

$$\tilde{A}^i_{\alpha,\beta} = \{x \in X : \mu_{\tilde{A}^i}(x) \geq \alpha, \nu_{\tilde{A}^i}(x) \leq \beta\} \quad ; 0 \leq \alpha, \beta \leq 1,\ \alpha + \beta \leq 1,$$

3 Arithmetic Operations on Intuitionistic Fuzzy Numbers

In this section, arithmetic operations on two IFNs are introduced using $\alpha-$cuts and $\beta-$cuts. Atanassov [10] has defined two kinds of cuts for IFS as given bellow.

For membership function, α - cut of $\tilde{A}^i$ is defined as

$$\tilde{A}^i_\alpha = \{x : \mu_{\tilde{A}^i}(x) \geq \alpha\} \ ; \alpha \in [0,1] \tag{1}$$

For non-membership function, β -cut of $\tilde{A}^i$ is defined as

$$\tilde{A}^i_\beta = \{x : \nu_{\tilde{A}^i}(x) \leq \beta\} \ ; \beta \in [0,1] \tag{2}$$

If $\tilde{A}^i$ is an IFN, then cuts will be closed intervals due to the property of convexity of IFN.

$$\tilde{A}^i_\alpha = \{x : \mu_{\tilde{A}^i}(x) \geq \alpha\} = \left[a^L(\alpha), a^R(\alpha)\right], \quad \forall \alpha \in (0,1] \tag{3}$$

$$\tilde{A}^i_\beta = \{x : \nu_{\tilde{A}^i}(x) \leq \beta\} = \left[a^L(\beta), a^R(\beta)\right] \quad \forall \beta \in [0,1), \tag{4}$$

where $a^L(\alpha), a^R(\beta)$ will be increasing functions of α and β respectively and $a^R(\alpha), a^L(\beta)$ will be decreasing functions of α and β respectively.

According to Kandel [16], the fuzzy number $\tilde{A}$ may be decomposed into their cuts. So, generalizing this concept for IFN $\tilde{A}^i$ with membership function $\mu_{\tilde{A}^i}$ and non-membership function $\nu_{\tilde{A}^i}$ which are defined as follows

$$\mu_{\tilde{A}^i} = \sum_{\alpha} \alpha \tilde{A}^i_{\alpha}, \qquad \nu_{\tilde{A}^i} = \sum_{\beta} \beta \tilde{A}^i_{\beta} \tag{5}$$

Let us consider two IFNs $\tilde{A}^i$ and $\tilde{B}^i$ with α-cuts $\tilde{A}^i_{\alpha} = \left[a^L(\alpha), a^R(\alpha)\right]$ and $\tilde{B}^i_{\alpha} = \left[b^L(\alpha), b^R(\alpha)\right]$ respectively and β-cuts $\tilde{A}^i_{\beta} = \left[a^L(\beta), a^R(\beta)\right]$ and $\tilde{B}^i_{\beta} = \left[b^L(\beta), b^R(\beta)\right]$ respectively. The arithmetic operations based on $\alpha-$ cuts and $\beta-$ cuts are presented as follows:

Intuitionistic Fuzzy number addition $\oplus$

$\tilde{A}^i \oplus \tilde{B}^i$ is an IFN with membership function $\mu_{\tilde{A}^i \oplus \tilde{B}^i}$ and non-membership function $\nu_{\tilde{A}^i \oplus \tilde{B}^i}$ which are defined as follows

$$\begin{aligned}\mu_{\tilde{A}^i \oplus \tilde{B}^i} &= \sum_{\alpha} \alpha\left(\tilde{A}^i_{\alpha} \oplus \tilde{B}^i_{\alpha}\right) = \sum_{\alpha} \alpha\left(\left[a^L(\alpha), a^R(\alpha)\right] \oplus \left[b^L(\alpha), b^R(\alpha)\right]\right) \\ &= \sum_{\alpha} \alpha\left[a^L(\alpha) \oplus b^L(\alpha), a^R(\alpha) \oplus b^R(\alpha)\right]\end{aligned}$$

$$\begin{aligned}\nu_{\tilde{A}^i \oplus \tilde{B}^i} &= \sum_{\beta} \beta\left(\tilde{A}^i_{\beta} \oplus \tilde{B}^i_{\beta}\right) = \sum_{\beta} \beta\left(\left[a^L(\beta), a^R(\beta)\right] \oplus \left[b^L(\beta), b^R(\beta)\right]\right) \\ &= \sum_{\beta} \beta\left[a^L(\beta) \oplus b^L(\beta), a^R(\beta) \oplus b^R(\beta)\right]\end{aligned}$$

Intuitionistic Fuzzy number minus Θ

$\tilde{A}^i \Theta \tilde{B}^i$ is an IFN with membership function $\mu_{\tilde{A}^i \Theta \tilde{B}^i}$ and non-membership function $\nu_{\tilde{A}^i \Theta \tilde{B}^i}$ which are defined as follows

$$\begin{aligned}\mu_{\tilde{A}^i \Theta \tilde{B}^i} &= \sum_{\alpha} \alpha\left(\tilde{A}^i_{\alpha} \Theta \tilde{B}^i_{\alpha}\right) = \sum_{\alpha} \alpha\left(\left[a^L(\alpha), a^R(\alpha)\right] \Theta \left[b^L(\alpha), b^R(\alpha)\right]\right) \\ &= \sum_{\alpha} \alpha\left[a^L(\alpha) \Theta b^R(\alpha), a^R(\alpha) \Theta b^L(\alpha)\right]\end{aligned}$$

$$\begin{aligned}\nu_{\tilde{A}^i \Theta \tilde{B}^i} &= \sum_{\beta} \beta\left(\tilde{A}^i_{\beta} \Theta \tilde{B}^i_{\beta}\right) = \sum_{\beta} \beta\left(\left[a^L(\beta), a^R(\beta)\right] \Theta \left[b^L(\beta), b^R(\beta)\right]\right) \\ &= \sum_{\beta} \beta\left[a^L(\beta) \Theta b^R(\beta), a^R(\beta) \Theta b^L(\beta)\right]\end{aligned}$$

Intuitionistic Fuzzy number multiplication $\otimes$

$\tilde{A}^i \otimes \tilde{B}^i$ is an IFN with membership function $\mu_{\tilde{A}^i \otimes \tilde{B}^i}$ and non-membership function $\nu_{\tilde{A}^i \otimes \tilde{B}^i}$ which are defined as follows

$$\begin{aligned}\mu_{\tilde{A}^i \otimes \tilde{B}^i} &= \sum_{\alpha} \alpha\left(\tilde{A}^i_{\alpha} \otimes \tilde{B}^i_{\alpha}\right) = \sum_{\alpha} \alpha\left(\left[a^L(\alpha), a^R(\alpha)\right] \otimes \left[b^L(\alpha), b^R(\alpha)\right]\right) \\ &= \sum_{\alpha} \alpha\left[a^L(\alpha) \otimes b^L(\alpha), a^R(\alpha) \otimes b^R(\alpha)\right]\end{aligned}$$

$$\nu_{\tilde{A}^i \otimes \tilde{B}^i} = \sum_{\beta} \beta\left(\tilde{A}^i_{\beta} \otimes \tilde{B}^i_{\beta}\right) = \sum_{\beta} \beta\left(\left[a^L(\beta), a^R(\beta)\right] \otimes \left[b^L(\beta), b^R(\beta)\right]\right)$$
$$= \sum_{\beta} \beta\left[a^L(\beta) \otimes b^L(\beta), a^R(\beta) \otimes b^R(\beta)\right]$$

Intuitionistic Fuzzy number division $\oslash$

$\tilde{A}^i \oslash \tilde{B}^i$ is an IFN with membership function $\mu_{\tilde{A}^i \oslash \tilde{B}^i}$ and non-membership function $\nu_{\tilde{A}^i \oslash \tilde{B}^i}$ which are defined as follows

$$\mu_{\tilde{A}^i \oslash \tilde{B}^i} = \sum_{\alpha} \alpha\left(\tilde{A}^i_{\alpha} \oslash \tilde{B}^i_{\alpha}\right) = \sum_{\alpha} \alpha\left(\left[a^L(\alpha), a^R(\alpha)\right] \oslash \left[b^L(\alpha), b^R(\alpha)\right]\right)$$
$$= \sum_{\alpha} \alpha\left[a^L(\alpha) / b^R(\alpha), a^R(\alpha) / b^L(\alpha)\right]$$
$$\nu_{\tilde{A}^i \oslash \tilde{B}^i} = \sum_{\beta} \beta\left(\tilde{A}^i_{\beta} \oslash \tilde{B}^i_{\beta}\right) = \sum_{\beta} \beta\left(\left[a^L(\beta), a^R(\beta)\right] \oslash \left[b^L(\beta), b^R(\beta)\right]\right)$$
$$= \sum_{\beta} \beta\left[a^L(\beta) / b^R(\beta), a^R(\beta) / b^L(\beta)\right]$$

4 Fuzzy System Reliability Analysis of Series and Parallel System

In this section, we present a method for analyzing fuzzy system reliability based on arithmetic operations of IFNs using $\alpha-$cuts and $\beta-$cuts.

Consider a series system having 'n' components shown in Fig.1, where the reliability of j^{th} component is represented by IFN $\tilde{R}^i_j, \quad j = 1, 2, ..., n.$ and assuming that reliabilities of the components may be represented by different types of IFNs. Then the reliability $\tilde{R}^i_S$ of serial system can be evaluated as follows:

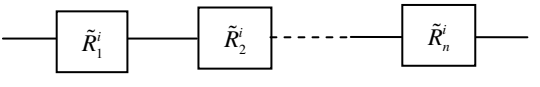


Fig. 1. Series system

Reliability of series system is

$$\tilde{R}^i_S = \tilde{R}^i_1 \otimes \tilde{R}^i_2 \otimes ... \otimes \tilde{R}^i_n \tag{6}$$

For membership function, α - cut of $\tilde{R}^i_j$ is

$$\tilde{R}^i_{j\alpha} = \left[r^L_{j\alpha}, r^R_{j\alpha}\right], \quad \forall \alpha \in (0,1], \quad j = 1, 2, ..., n.$$

For non-membership function, β - cut of $\tilde{R}_j^i$ is

$$\tilde{R}_{j\beta}^i = \left[r_{j\beta}^L, r_{j\beta}^R \right], \quad \forall \beta \in [0,1), \quad j = 1,2,...,n.$$

Then based on previous discussion,

For membership function, α - cut of $\tilde{R}_S^i$ is

$$\begin{aligned} \tilde{R}_{S\alpha}^i &= \tilde{R}_{1\alpha}^i \otimes \tilde{R}_{2\alpha}^i \otimes ... \otimes \tilde{R}_{n\alpha}^i \\ &= \left[r_{1\alpha}^L, r_{1\alpha}^R \right] \otimes \left[r_{2\alpha}^L, r_{2\alpha}^R \right] \otimes ... \otimes \left[r_{n\alpha}^L, r_{n\alpha}^R \right] \\ &= \left[r_{1\alpha}^L r_{2\alpha}^L ... r_{n\alpha}^L \ , \ r_{1\alpha}^R r_{2\alpha}^R ... r_{n\alpha}^R \right] \end{aligned} \tag{7}$$

For non-membership function, β - cut of $\tilde{R}_S^i$ is

$$\begin{aligned} \tilde{R}_{S\beta}^i &= \tilde{R}_{1\beta}^i \otimes \tilde{R}_{2\beta}^i \otimes ... \otimes \tilde{R}_{n\beta}^i \\ &= \left[r_{1\beta}^L, r_{1\beta}^R \right] \otimes \left[r_{2\beta}^L, r_{2\beta}^R \right] \otimes ... \otimes \left[r_{n\beta}^L, r_{n\beta}^R \right] \\ &= \left[r_{1\beta}^L r_{2\beta}^L ... r_{n\beta}^L \ , \ r_{1\beta}^R r_{2\beta}^R ... r_{n\beta}^R \right] \end{aligned} \tag{8}$$

Thus system reliability $\tilde{R}_S^i$ is an IFN with membership function $\mu_{\tilde{R}_S^i}$ and non-membership function $\nu_{\tilde{R}_S^i}$ which are defined as follows

$$\mu_{\tilde{R}_S^i} = \sum_{\alpha} \alpha \tilde{R}_{S\alpha}^i = \sum_{\alpha} \alpha \left[r_{1\alpha}^L r_{2\alpha}^L ... r_{n\alpha}^L \ , \ r_{1\alpha}^R r_{2\alpha}^R ... r_{n\alpha}^R \right] \tag{9}$$

$$\nu_{\tilde{R}_S^i} = \sum_{\beta} \beta \tilde{R}_{S\beta}^i = \sum_{\beta} \beta \left[r_{1\beta}^L r_{2\beta}^L ... r_{n\beta}^L \ , \ r_{1\beta}^R r_{2\beta}^R ... r_{n\beta}^R \right] \tag{10}$$

Consider a parallel system having 'n' components shown in Fig. 2, where the reliability of j^{th} component is represented by IFN $\tilde{R}_j^i, \quad j = 1,2,...,n.$ and assuming that reliabilities of the components may be represented by different types of IFNs. Then the reliability $\tilde{R}_P^i$ of the parallel system can be evaluated as follows:

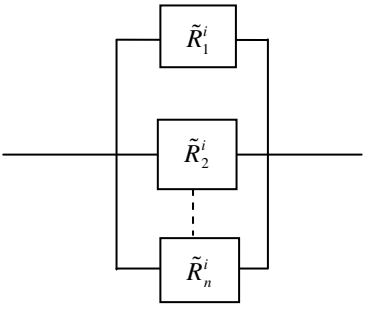


Fig. 2. Parallel system

Reliability of parallel system is

$$\tilde{R}_P^i = 1 \Theta \prod_{j=1}^{n} \left(1 \Theta \tilde{R}_j^i\right) \tag{11}$$

Then based on previous discussion,

For membership function, α - cut of $\tilde{R}_P^i$ is

$$\tilde{R}_{P\alpha}^i = 1 \Theta \prod_{j=1}^{n} \left(1 \Theta \tilde{R}_{j\alpha}^i\right) \tag{12}$$

For non-membership function, β - cut of $\tilde{R}_P^i$ is

$$\tilde{R}_{P\beta}^i = 1 \Theta \prod_{j=1}^{n} \left(1 \Theta \tilde{R}_{j\beta}^i\right) \tag{13}$$

Thus, system reliability $\tilde{R}_P^i$ is an IFN with membership function $\mu_{\tilde{R}_P^i}$ and non-membership function $\nu_{\tilde{R}_P^i}$ which are defined as follows

$$\mu_{\tilde{R}_P^i} = \sum_{\alpha} \alpha \tilde{R}_{P\alpha}^i, \qquad \nu_{\tilde{R}_P^i} = \sum_{\beta} \beta \tilde{R}_{P\beta}^i \tag{14}$$

5 Numerical Example

Consider the system shown in Fig. 3, where the reliability of the components P_1, P_2, P_3, and P_4 are represented by different type IFNs $\tilde{R}_1^i$, $\tilde{R}_2^i$, $\tilde{R}_3^i$ and $\tilde{R}_4^i$ respectively. Assumed values of $\tilde{R}_1^i$, $\tilde{R}_2^i$, $\tilde{R}_3^i$ and $\tilde{R}_4^i$ are presented in Table 1.

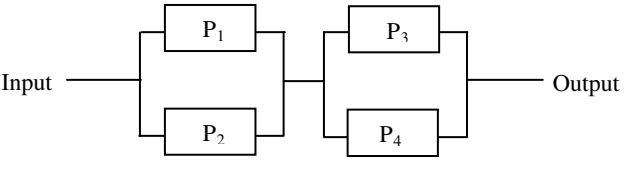


Fig. 3. Series- parallel system

Based on the previous discussions, the reliability $\tilde{R}$ of the system can be represented by

$$\tilde{R}=\left[1\ominus\left(1\ominus\tilde{R}_1\right)\otimes\left(1\ominus\tilde{R}_2\right)\right]\otimes\left[1\ominus\left(1\ominus\tilde{R}_3\right)\otimes\left(1\ominus\tilde{R}_4\right)\right]$$
$$=\left[\tilde{R}_1\oplus\tilde{R}_2\ominus\tilde{R}_1\otimes\tilde{R}_2\right]\otimes\left[\tilde{R}_3\oplus\tilde{R}_4\ominus\tilde{R}_3\otimes\tilde{R}_4\right]$$

For membership function, α - cut of $\tilde{R}^i$ is

$$\tilde{R}^i_\alpha = \left[\tilde{\mathrm{R}}_{1\alpha}\oplus\tilde{\mathrm{R}}_{2\alpha} \ominus \tilde{\mathrm{R}}_{1\alpha}\otimes\tilde{\mathrm{R}}_{2\alpha}\right]\otimes\left[\tilde{\mathrm{R}}_{3\alpha}\oplus\tilde{\mathrm{R}}_{4\alpha} \ominus \tilde{\mathrm{R}}_{3\alpha}\otimes\tilde{\mathrm{R}}_{4\alpha}\right] \quad (15)$$

For non-membership function, β - cut of $\tilde{R}^i$ is

$$\tilde{R}^i_\beta = \left[\tilde{\mathrm{R}}_{1\beta}\oplus\tilde{\mathrm{R}}_{2\beta} \ominus \tilde{\mathrm{R}}_{1\beta}\otimes\tilde{\mathrm{R}}_{2\beta}\right]\otimes\left[\tilde{\mathrm{R}}_{3\beta}\oplus\tilde{\mathrm{R}}_{4\beta} \ominus \tilde{\mathrm{R}}_{3\beta}\otimes\tilde{\mathrm{R}}_{4\beta}\right] \quad (16)$$

Table 1.

Reliability $\tilde{R}^i_j$ of j^{th} component of	Type of intuitionistic fuzzy number
$\tilde{R}^i_1 = (0.04, 0.05, 0.06; 0.03, 0.05, 0.08)$	Triangular intuitionistic fuzzy number [12]
$\tilde{R}^i_2 = (0.03, 0.04, 0.05, 0.07; 0.02, 0.04, 0.05, 0.08)$	Trapezoidal intuitionistic fuzzy number [15]
$\tilde{R}^i_3 = (0.04, 0.07, 0.09, 0.12; 0.03, 0.05, 0.1, 0.15)$	Trapezoidal intuitionistic fuzzy number [13]
$\tilde{R}^i_4 = (0.03, 0.05, 0.07, ; 0.02, 0.05, 0.09)$	Triangular intuitionistic fuzzy number [12]

Firstly, $\alpha-$cuts and $\beta-$ cuts of IFNs $\tilde{R}^i_1$, $\tilde{R}^i_2$, $\tilde{R}^i_3$ and $\tilde{R}^i_4$ are computed as listed in Table 2 and Table 3 respectively, where $\alpha, \beta \in [0,1]$. Then based on Tables (2 and 3) and formulas (15 and 16), $\alpha-$cuts and $\beta-$ cuts of $\tilde{R}^i$ are computed for different values of α and β as shown in Table 4.

Table 2. For membership, $\alpha-$ cuts of $\tilde{R}^i_1$, $\tilde{R}^i_2$, $\tilde{R}^i_3$ and $\tilde{R}^i_4$

α	$\tilde{R}^i_{1\alpha}$	$\tilde{R}^i_{2\alpha}$	$\tilde{R}^i_{3\alpha}$	$\tilde{R}^i_{4\alpha}$
0	[0.04, 0.06]	[0.03, 0.07]	[0.04, 0.12]	[0.03, 0.07]
0.1	[0.041, 0.059]	[0.031, 0.068]	[0.043, 0.117]	[0.032, 0.068]
0.2	[0.042, 0.058]	[0.032, 0.066]	[0.046, 0.114]	[0.034, 0.066]
0.3	[0.043, 0.057]	[0.033, 0.064]	[0.049, 0.111]	[0.036, 0.064]
0.4	[0.044, 0.056]	[0.034, 0.062]	[0.052, 0.108]	[0.038, 0.062]
0.5	[0.045, 0.055]	[0.035, 0.060]	[0.055, 0.105]	[0.040, 0.060]
0.6	[0.046, 0.054]	[0.036, 0.058]	[0.058, 0.102]	[0.042, 0.058]
0.7	[0.047, 0.053]	[0.037, 0.056]	[0.061, 0.099]	[0.044, 0.056]
0.8	[0.048, 0.052]	[0.038, 0.054]	[0.064, 0.096]	[0.046, 0.054]
0.9	[0.049, 0.051]	[0.039, 0.052]	[0.067, 0.093]	[0.048, 0.052]
1	[0.05, 0.05]	[0.04, 0.05]	[0.07, 0.09]	[0.05, 0.05]

Table 3. For non-membership, β – cuts of $\tilde{R}_1^i$, $\tilde{R}_2^i$, $\tilde{R}_3^i$ and $\tilde{R}_4^i$

β	$\tilde{R}_{1\beta}^i$	$\tilde{R}_{2\beta}^i$	$\tilde{R}_{3\beta}^i$	$\tilde{R}_{4\beta}^i$
0	[0.05, 0.05]	[0.04, 0.05]	[0.05, 0.1]	[0.05, 0.05]
0.1	[0.048, 0.053]	[0.038, 0.053]	[0.048, 0.105]	[0.047, 0.054]
0.2	[0.046, 0.056]	[0.036, 0.056]	[0.046, 0.110]	[0.044, 0.058]
0.3	[0.044, 0.059]	[0.034, 0.059]	[0.044, 0.115]	[0.041, 0.062]
0.4	[0.042, 0.062]	[0.032, 0.062]	[0.042, 0.12]	[0.038, 0.066]
0.5	[0.04, 0.065]	[0.03, 0.065]	[0.04, 0.125]	[0.035, 0.07]
0.6	[0.038, 0.068]	[0.028, 0.068]	[0.038, 0.13]	[0.032, 0.074]
0.7	[0.036, 0.071]	[0.026, 0.071]	[0.036, 0.135]	[0.029, 0.078]
0.8	[0.034, 0.074]	[0.024, 0.074]	[0.034, 0.14]	[0.026, 0.082]
0.9	[0.032, 0.077]	[0.022, 0.077]	[0.032, 0.145]	[0.023, 0.086]
1	[0.03, 0.08]	[0.02, 0.08]	[0.03, 0.15]	[0.02, 0.09]

Table 4. $\alpha-$cuts and $\beta-$ cuts of $\tilde{R}^i(t)$

α,β	$\tilde{R}_\alpha^i(t)$	$\tilde{R}_\beta^i(t)$
0	[0.004053, 0.024317]	[0.008313, 0.014455]
0.1	[0.004558, 0.023087]	[0.007431, 0.016329]
0.2	[0.005086, 0.021886]	[0.006595, 0.018314]
0.3	[0.005636, 0.020716]	[0.005803, 0.020411]
0.4	[0.006208, 0.019575]	[0.005057, 0.022618]
0.5	[0.006803, 0.018466]	[0.004358, 0.024936]
0.6	[0.007420, 0.017386]	[0.003706, 0.027363]
0.7	[0.008059, 0.016338]	[0.003103, 0.029899]
0.8	[0.008720, 0.015320]	[0.002548, 0.032545]
0.9	[0.009402, 0.014333]	[0.002044, 0.035299]
1	[0.010106, 0.013377]	[0.001591, 0.038160]

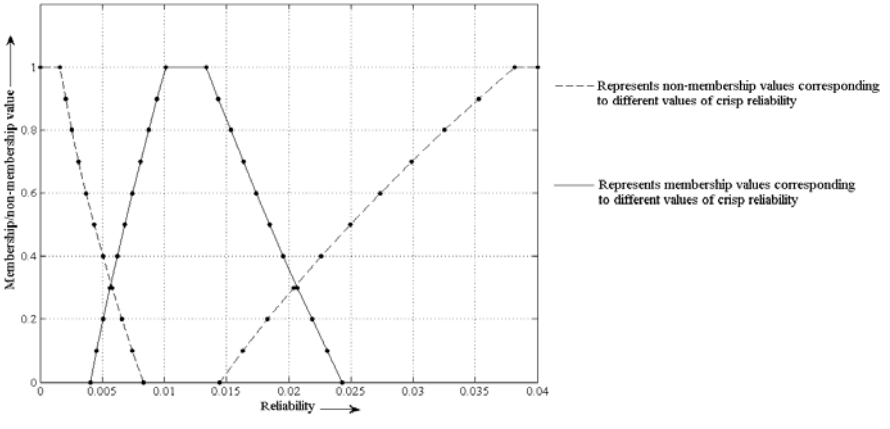


Fig. 4. Membership function and non-membership function of system reliability

Finally, based on the formula (5) and using results shown in Table 4, membership function and non-membership function of the fuzzy system reliability can be obtained as shown in Fig. 4.

5 Conclusion

This paper addresseṣa new approach for fuzzy system reliability analysis based on arithmetic operations of intuitionistic fuzzy numbers using $\alpha-$cuts and $\beta-$cuts, where the reliabilities of the components of a system are represented by different types of intuitionistic fuzzy numbers. The proposed method allows the reliabilities of the components of a system to have different types of intuitionistic fuzzy numbers, so it is more flexible than the ones presented in [12, 13]. Furthermore, because the proposed method uses the simple arithmetic operations of intuitionistic fuzzy numbers based on α-cuts and β-cuts rather than the complicated nonlinear programming techniques mentioned in [17], it is simpler in calculating fuzzy system reliability than the one presented in that paper. Moreover, our study generalizes the various works [7, 8, 12-13, 17, 18] of the literature. One can also use the proposed approach to analyze the fuzzy reliability of parallel-system.

Acknowledgement. The first author gratefully acknowledges the financial support given by the Council of Scientific and Industrial Research, Govt. of India, New Delhi, India.

References

[1] Onisawa, T., Kacprzyk, J.: Reliability and safety under fuzziness. Physica Verlag (1995)
[2] Cai, K.Y., Wen, C.Y., Zhang, M.L.: Fuzzy variables as a basis for a theory of fuzzy reliability in the possibility context. Fuzzy Sets and Systems 42, 145–172 (1991)
[3] Cai, K.Y., Wen, C.Y., Zhang, M.L.: Posbist reliability behavior of typical systems with two types of failures. Fuzzy Sets and Systems 43, 17–32 (1991)
[4] Cai, K.Y., Wen, C.Y., Zhang, M.L.: Fuzzy states as a basis for a theory of fuzzy reliability. Microelectronic Reliability 33, 2253–2263 (1993)
[5] Cai, K.Y., Wen, C.Y., Zhang, M.L.: Coherent systems in profust reliability theory. In: Onisawa, T., Kacprzyk, J. (eds.) Reliability and Safety Analyzes Under Fuzziness, pp. 81–94. Physica-Verlag, Heidelberg (1995)
[6] Singer, D.: A fuzzy set approach to fault tree and reliability analysis. Fuzzy Sets and Systems 34, 145–155 (1990)
[7] Cheng, C.H., Mon, D.L.: Fuzzy system reliability analysis by interval of confidence. Fuzzy Sets and Systems 56, 29–35 (1993)
[8] Chen, S.M.: Fuzzy system reliability analysis using fuzzy number arithmetic operations. Fuzzy Sets and Systems 64, 31–38 (1994)
[9] Atanassov, K.T.: Intuitionistic fuzzy sets. Central Tech Library, Bulgarian Academy Science, Sofia, Bulgaria, Rep. No. 1697/84 (1983)
[10] Atanassov, K.T.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20, 87–96 (1986)

[11] Burillo, P., Bustince, H., Mohedano, V.: Some definition of intuitionistic fuzzy number. first properties. In: Proc. of the First Workshop on Fuzzy Based Expert Systems, Sofia, Bulgaria, September 28-30, pp. 53–55 (1994)
[12] Mahapatra, G.S., Roy, T.K.: Reliability evaluation using triangular intuitionistic fuzzy number. Int. J. Mathematical and Statistical Sciences 1(1), 31–38 (2009)
[13] Kumar, M., Yadav, S.P., Kumar, S.: A new approach for analysing the fuzzy system reliability using intuitionistic fuzzy number. Int. J. Industrial and Systems Engineering 8(2), 135–156 (2011)
[14] Grzegorzewski, P.: Distances and orderings in a family of intuitionistic fuzzy numbers. In: EUSFLAT Conf., pp. 223–227 (2003)
[15] Xin, L.Z., Cheng, Z., Yang, W.L., Min, F.E.: The Order of Intuitionistic Fuzzy Number. Intelligent Information Management Systems and Technologies 2, 490–496 (2006)
[16] Kandel, A.: Fuzzy Mathematical Techniques with Applications. Addison-Wesley, Reading (1986)
[17] Mon, D.L., Cheng, C.H.: Fuzzy system reliability analysis for components with different membership functions. Fuzzy Sets and Systems 64, 145 (1994)
[18] Kumar, M., Yadav, S.P., Kumar, S.: Fuzzy system reliability evaluation using time-dependent intuitionistic fuzzy set. International Journal of Systems Science iFirst, 1–17 (2011)

Cognitive Radio Parameter Adaptation Using Multi-objective Evolutionary Algorithm

Deepak K. Tosh[1], Siba K. Udgata[1], and Samrat L. Sabat[2]

[1] Department of Computer and Information Sciences, University of Hyderabad, Hyderabad-500046, India
dktosh@gmail.com, udgatacs@uohyd.ernet.in
[2] School of Physics, University of Hyderabad, Hyderabad-500046, India
slssp@uohyd.ernet.in

Abstract. Cognitive Radio (CR) is an intelligent Software Defined Radio (SDR) that can alter its transmission parameters according to predefined objectives by sensing the dynamic wireless environment. In this paper, we propose a method to determine the necessary transmission parameters for a multicarrier system based on multiple scenarios using a multi-objective evolutionary algorithm like Non-dominated Sorting based Genetic Algorithm (NSGA-II). Each scenario is represented by a fitness function and represented as a composite function of one or more radio parameters. We model the CR parameter adaptation problem as an unconstrained multi-objective optimization problem and then propose an algorithm to optimize the CR transmission parameters based on NSGA-II. We compute the fitness score by considering multiple scenarios at a time and then evolving the solution until optimal value is reached. The final results are represented as a set of optimal solutions referred as *pareto-front* for the given scenarios. We performed multi-objective optimization considering two objectives and the best individual fitness values which are obtained after final iteration are reported here as the *pareto-front*.

1 Introduction

Ideally a Cognitive Radio (CR) possesses the capability of sensing, perceiving, decision making, planning, and learning[1] in a wireless environment. Thus CR can alter its transmission parameters to ensure optimal usage of currently available resources, and achieve required Quality of Service (QoS). It alleviates spectrum scarcity problem[2] [4] identified by Federal Corporation of Communication (FCC). The cognitive radio must sense the environment periodically and appropriately alter the transmission parameters according to objectives and requirements of the user. This learning characteristic of CR makes it intelligent that adapts itself in dynamic situations.

Though CR has a main goal to improve spectrum utilization efficiency[3], it can also have other goals to achieve; like Bit Error Rate (BER) minimization,

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 737–748.
springerlink.com

transmit power minimization, interference minimization, throughput maximization, and spectral efficiency maximization. Therefore, the transmission parameters have to be optimized according to the specified objectives of a user. The tunable transmission parameters of CR are transmission power (P), Modulation Type, Modulation Index (m), Symbol Rate (R_s), Bandwidth (B), Frame Length (L), Time Division Duplex (TDD) in percentage, and Coding Rate (R_c) etc. Many research works have been done on spectrum sensing, allocation as well as parameter adaptation of CR. Dynamic power adjustment schemes have been proposed in [7] [8] for wireless systems. For attaining high throughput, adaptive modulation scheme is proposed in [10] [11] [12]. A set of single carrier and multicarrier fitness functions have been derived and GA is used to search the optimal set of transmission parameters in [5]. Adaptive transmission in the context of cognitive radio networks is addressed in [17] where GA is used to optimize the CR parameters by considering several QoS and channel coding schemes. A Binary Ant Colony Optimization (BACO) algorithm is used to solve the problem by formulating it into single objective function [13]. Particle Swarm Optimization (PSO) is also being used in [14] to determine the CR parameters for a given set of objectives. It was observed that PSO outperforms GA in terms of convergence speed and consistency. A multi-objective hybrid genetic algorithm (HGA) is used to determine the optimal set of radio transmission parameters for a single carrier CR engine in [9].

All the above works have attempted to solve the parameter adaptation problem in CR by formulating it into single objective functions using weighted sum approach. If a multi-objective optimization problem is solved by weight vector aggregate sum approach the result will find a single value for each parameter and thus the behavior of solution space cannot be deduced with respect to multiple objective functions. For the real-time scenario, the performance of CR system depends on the choice of weights to specific objective functions which is hard and scenario dependent.

A Non-dominated Neighbor Distribution multi-objective genetic algorithm (NNDA) is proposed in [15] which ensures superior individuals are always selected to next generation. The same problem is solved by using multi-objective immune genetic algorithm (MIGA) based on non-dominated sort in [16] and a comparative study with NSGA-II[18] is performed. These works considered two scenarios only; minimization of power and maximization of bit rate. The fitness functions representing two scenarios do not consider parameters like modulation index (m), bandwidth (B), coding rate (R_c) and symbol rate (R_s) which play significant role in achieving QoS. Additionally, there can be more scenarios that must be considered for finding all reconfigurable CR parameters. In our work, we focused to perform a simultaneous optimization of more than one conflicting objectives at a time using Non-dominated Sorting-based Genetic Algorithm (NSGA-II) instead of weighted sum approach. This helps to find a set of optimal solutions from the large range of solution space defined in terms of pareto optimal set.

For experimentation, we have performed multi-objective optimization by taking two objectives at a time and best individual fitness values which are obtained after final iteration are reported here as the *pareto-front*.

The rest of the paper is organized as follows. Different radio parameters along with single and multicarrier objective functions are described in section 2. In section 3, the multi-objective approach to the parameter adaptation problem is briefly discussed. The experimental setup for CR adaptation, its analysis and results are reported in section 4. Finally, conclusions are drawn in section 5.

2 Problem Definition and Cognitive Radio Objectives

Assuming a multicarrier dynamic wireless environment with N_c subcarriers, the basic characteristic of CR is to sense the environmental parameters, and it learns itself to adjust the value of transmission parameters to achieve the predefined quality of service (QoS). In CR system, the environmental parameters act as input to the problem and the transmission parameters act as decision variables. Hence the problem can be defined as to find the set of transmission parameters by modeling the scenarios as multi-objective functions. The proposed algorithm is based on NSGA-II to solve the formulated multi-objective function to obtain the required solutions. The below subsection briefs about the radio parameters involved in CR engine [5].

2.1 Radio Parameters

Radio parameters of CR are categorized into environmental parameters and transmission parameters. The former gives knowledge of environmental characteristics of multicarrier wireless environment which are used as inputs to the CR system, where different environmental parameters[6] include path loss, noise power, signal-to-noise ratio (SNR) in decibel, spectrum information etc.

Transmission parameters are the tunable parameters of CR system. The CR tunes its transmission knobs to corresponding values from the optimal parameter set. The transmission parameters [6] are listed as: transmit power (P), type of modulation (mod_type) scheme used for the communication, modulation index (m), bandwidth (B), channel coding rate (R_c), size of transmission frame in byte (L), time division duplex in percentage (TDD), and symbol rate (R_s). Modulation Index (m) is defined as the total number of symbols in a modulation scheme and TDD represent the percentage of transmit time.

2.2 Objective Functions

The objective functions guide the CR towards finding a set of optimal solutions. Based on the type of scenario and QoS demand, the CR autonomously

chooses a parameter set from the *pareto-front* to work on. In this work five objective functions: Bit-Error-Rate minimization, system throughput maximization, power consumption minimization, spectral interference minimization, and spectral efficiency maximization are considered [6]. In the next subsections, we describe the objective functions in detail.

Minimizing Bit-Error-Rate (BER): The ideal goal of communication system is to maintain received power same as transmit power or in other words to minimize the bit error rate in the communication process. Generally bit error rate depends on several parameters like transmit power (P), modulation type, modulation index (m), bandwidth (B), symbol rate (R_s) and noise power (N). Mathematically, the normalized objective function for minimizing BER [6] is defined as

$$f_{min\_ber} = 1 - \frac{log_{10}(0.5)}{log_{10}\bar{P}_{be}} \tag{1}$$

Where $\bar{P}_{be}$ is the mean BER probability for N_c subcarriers. The probability of bit error rate (P_{be}) [19] which is dependent on type of modulation scheme and modulation index (m), can be computed as
In case of BPSK modulation,

$$P_{be} = Q(\sqrt{\gamma}) \tag{2}$$

For m-ary PSK modulation,

$$P_{be} = \frac{2}{log_2(m)} Q(\sqrt{2 * log_2(m) * \gamma} * sin\frac{\pi}{m}) \tag{3}$$

For m-ary QAM modulation,

$$P_{be} = \frac{4}{log_2(m)}(1 - \frac{1}{\sqrt{m}})Q(\sqrt{\frac{3 * log_2(m)}{m-1} * \gamma}) \tag{4}$$

Where $Q(x)$ represents the Q-function in terms of error function. The energy per bit (γ) is expressed as

$$\gamma = \frac{E_b}{N_0} = 10log_{10}\left[\frac{P}{N}\right] + 10log_{10}\left[\frac{B}{R_s * m}\right](dB) \tag{5}$$

Maximizing Throughput: By definition, throughput is defined as the amount of correct information received at the receiver. Maximizing throughput is a common aim in the multimedia environment. Determining the theoretical throughput depends on the parameters such as bandwidth (B) in use, coding rate (R_c), modulation index (m), frame size (L), probability of BER (P_{be})and percentage of transmit time (TDD). The normalized objective function for maximizing throughput [6] for N_c subcarriers is defined as

$$f_{max\_tp} = \frac{\sum_{i=1}^{N_c} \frac{L_i}{L_i+O+H} * (1 - P_{be_i})^{L_i+O} * R_{c_i} * TDD_i}{N_c} \tag{6}$$

Where, O is physical layer overhead and H is MAC and IP layer overhead. Subscript i refers to i^{th} subcarrier of the multicarrier system.

Minimizing Power Consumption: Energy is the most important factor which must be spent optimally for the communication process. Hence minimum power should be consumed for all the tasks while communicating or computing. The parameters contribute to the fitness for power minimization are bandwidth (B), modulation type, coding rate (R_c), time division duplexing (TDD), symbol rate (R_s) and transmit power (P). The normalized objective function for minimizing power [6] is expressed as

$$f_{min\_power} = 1 - \left[\alpha * \frac{\sum_{i=1}^{N_c}(P_{max} + B_{max}) - (P_i + B_i)}{N_c * (P_{max} + B_{max})} + \beta * \frac{\sum_{i=1}^{N_c} log_2(m_{max}) - log_2(m_i)}{N_c * log_2(m_{max})} + \lambda * \frac{\sum_{i=1}^{N_c} R_{s_{max}} - R_{s_i}}{N_c * R_{s_{max}}} \right] \tag{7}$$

Where, α, β, and λ represents weighting factors of the objective function. Subscript i refers to i^{th} subcarrier of the multicarrier system.

Minimizing Spectral Interference: Interference is caused due to simultaneous transmissions by multiple users in the same spectrum band. So it affects throughput as well as bit error rate of the communication system. To achieve high throughput with less error rate, minimization of spectral interference is necessary. Transmission parameters such as transmit power (P), bandwidth (B) and time division duplexing (TDD) are used to determine the approximate amount of spectral interference fitness value. The normalized objective function for minimizing interference [6] is represented as:

$$f_{min\_si} = 1 - \frac{\sum_{i=1}^{N_c}((P_i * B_i * TDD_i) - (P_{min} * B_{min} * 1))}{N_c * (P_{max} * B_{max} * 100)} \tag{8}$$

Maximizing Spectral Efficiency: Spectral efficiency can be defined as the amount of information that can be transmitted over a given bandwidth. The symbol rate (R_s) and modulation index (m) are used to determine the total amount of information being transmitted. To maximize the spectral efficiency, high amount of information is needed in small bandwidth. The normalized objective for maximizing spectral efficiency [6] is expressed as

$$f_{max\_se} = \frac{\sum_{i=1}^{N_c} \frac{m_i * R_{s_i} * B_{min}}{B_i * m_{max} * R_{s_{max}}}}{N_c} \tag{9}$$

In the above expressions, parameter with subscript max and min represent the maximum and minimum value of that parameter respectively defined in table 1.

Algorithm 1. CR_PARAMETER_ADAPTATION

Input: Environmental sensing information
Output: Set of optimal transmission parameters

Define all the objective functions and search range for the radio parameters
for all distinct combinations of two objective functions **do**
Extract the environment parameters from sensed information of wireless environment.
Initialize the population P (Transmission Parameters) with uniform random numbers in a given search range.
while Stopping criteria is not met or Required accuracy is not reached **do**
Find out the optimal solution set by performing multi-objective optimization using NSGA-II.
end while
Plot the obtained optimal solution as *pareto-front* and record the transmission parameter values for CR adaptation.
end for

3 Multi-objective Approach for Parameter Adaptation

In general, a multi-objective optimization problem is to determine the optimal value of a set of solutions ($\overrightarrow{x}$) while optimizing a set of k conflicting objectives simultaneously. Mathematically, the multi-objective unconstrained optimization problem is defined as

$$\begin{aligned}&\text{Find } \overrightarrow{x} = (x_1, x_2, x_3, \cdots, x_D)\\&\text{that optimizes } \langle f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), \cdots, f_k(\overrightarrow{x})\rangle\\&\text{with } l_i \leq x_i \leq u_i,\ i = 1, 2, \cdots, D\end{aligned}$$

In the above formulation, $\overrightarrow{x}$ is the decision variable with X as parameter space and $\overrightarrow{y}$ is the set of objectives with Y as objective space; k is the number of objectives, D is the number of decision variables in the problem. l_i and u_i defines the lower and upper bounds of defining x_i, the search space of D-distinct radio parameters. The CR adaptation problem can be modeled as a multi-objective optimization problem which is defined in following

$$\begin{aligned}&\text{Find } \overrightarrow{x} = \langle P, mod\_type, m, B, R_c, L, TDD, R_s\rangle\\&\text{that minimizes,}\\&\left\langle f_{min\_ber}(\overrightarrow{x}), f'_{min\_tp}(\overrightarrow{x}), f_{min\_power}(\overrightarrow{x}), f_{min\_si}(\overrightarrow{x}), f'_{min\_se}(\overrightarrow{x})\right\rangle\\&\text{where } f'_{min\_tp}(\overrightarrow{x}) = -f_{max\_tp}(\overrightarrow{x})\\&\text{and } f'_{min\_se}(\overrightarrow{x}) = -f_{max\_se}(\overrightarrow{x})\end{aligned}$$

The proposed algorithm 1 finds out the optimal transmission parameters and pareto optimal front, considering two objectives at a time. Initially, all the objective functions have to be defined which return the fitness score based on

a single parameter set and the feasible search range must be specified. For all distinct combinations of two objective functions, we compute the fitness score for population size P and performed NSGA-II that evolves the solution until optimal value is reached. The final results obtained are referred as *pareto-front* for the two scenarios taken.

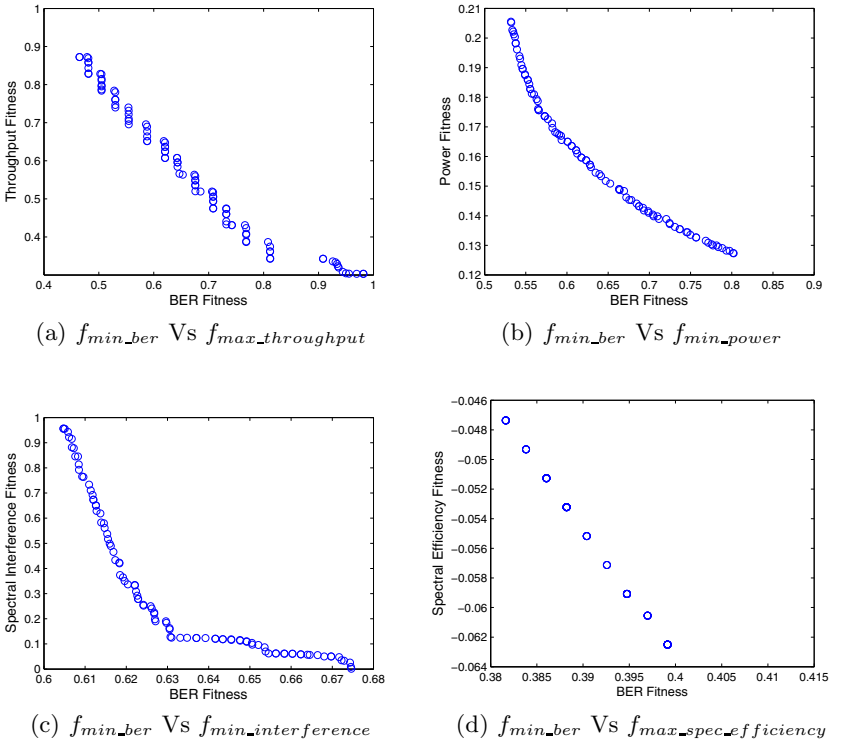
4 Simulation Setup and Results

We simulated the above mentioned multi-objective problem using Non-dominated Sorting based Genetic Algorithm (NSGA-II) in Matlab. We tried to address the problem by taking two conflicting objectives at a time and perform the optimization by assuming 16 subcarriers in the CR network. The parameter list with their labels and range of values are shown in the table 1. A population size of 100 and iterations of 1500 is considered during simulation. To obtain an optimal front, both objective functions need to be conflicting with each other, that means there should be some common parameters that guide both fitness functions towards optimality. As all the fitness functions are not linear, the nature of front can be nonlinear as well as discrete. We have taken all possible distinct combinations by taking two objectives at a time. The best individual fitness values which are obtained after final iteration are reported as the pareto set best described using *pareto-front* as in figures 1–3.

Table 1. Transmission Parameters with their Ranges

$Parameter(x)$	x_{min}	x_{max}
Transmit Power (P)	1 dBm	18 dBm
Modulation Index (m)	2	128
Bandwidth (B)	2 MHz	32 MHz
Coding Rate (R_c)	1/2	3/4
Frame Length (L)	94 Bytes	1504 bytes
Time Division Duplex (TDD)	1%	100%
Symbol Rate (R_s)	125 Ksps	1 Msps
Noise Power (N)	-30 dBm	-5 dBm

The pareto optimal fronts obtained by taking BER minimization fitness and one from all other fitness functions is shown in figure 1. In the figure 1(a) represents the *pareto-front* between fitness for minimizing BER and maximizing throughput. The probability of BER (P_{be}) is a common parameter in Eq. 1 and 6 . It can be inferred from the equations that if P_{ber} increases then f_{min_ber} and f'_{min_tp} increases. Hence f_{min_ber} and f_{max_tp} exhibit an inverse relationship which is depicted in the figure 1(a). In figure 1(b) the typical nature of minimization of BER and power fitness is depicted. In practice,



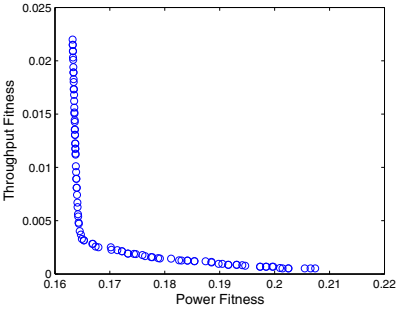
(a) f_{min_ber} Vs $f_{max\_throughput}$ (b) f_{min_ber} Vs f_{min_power}

(c) f_{min_ber} Vs $f_{min\_interference}$ (d) f_{min_ber} Vs $f_{max\_spec\_efficiency}$

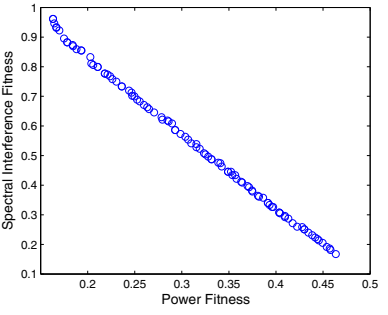
Fig. 1. *Pareto-front* obtained with respect to fitness for BER minimization

if transmit power (P) decreases, then probability of BER increases. Hence the f_{min_ber} in Eq. 1 increases where as the f_{min_power} in Eq. 7 decreases. Figure 1(c) shows optimal pareto-front between fitness for BER minimization (Eq. 1) and fitness for interference minimization (Eq. 8). In both fitness functions, power (P), symbol rate (R_s), modulation index (m) and bandwidth (B) are the common parameters which affect bit error rate and interference fitness. The f_{min_si} decreases with increase in f_{min_ber}. The optimal *pareto-front* for fitness of BER minimization (Eq. 1) and spectral efficiency maximization (Eq. 9) is shown in figure 1(d). As f'_{min_si} depends on modulation index (m), symbol rate (R_s) and bandwidth (B), so P_{be} is the common factor for f_{min_ber} and f'_{min_se}. Thus, both fitness functions are inversely related to each other due to the parameter P_{be}.

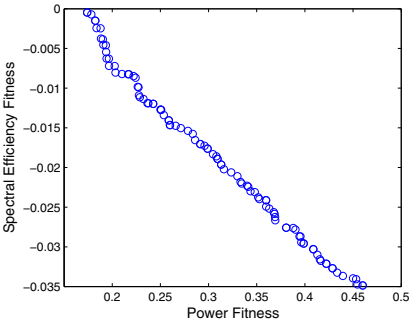
The pareto optimal fronts obtained by taking fitness for interference minimization and one from all other fitness functions without any repetition, is shown in figure 2.The *pareto-front* between minimizing power fitness (Eq. 7) and maximizing throughput fitness (Eq. 6) is shown in figure 2(a). The nature



(a) f_{min_power} Vs $f_{max\_throughput}$



(b) f_{min_power} Vs $f_{min\_interference}$

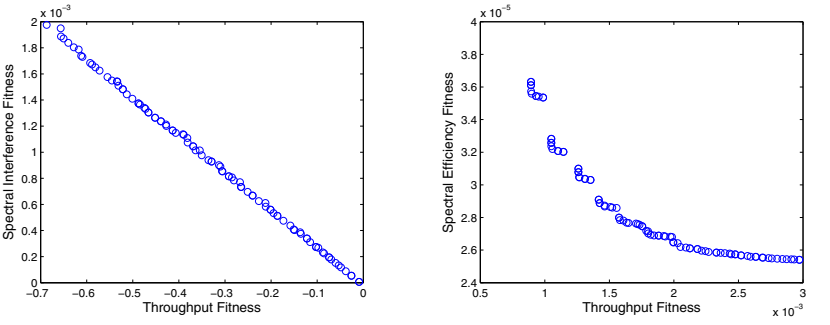


(c) f_{min_power} Vs $f_{max\_spec\_efficiency}$

Fig. 2. *Pareto-front* obtained with respect to fitness for power minimization

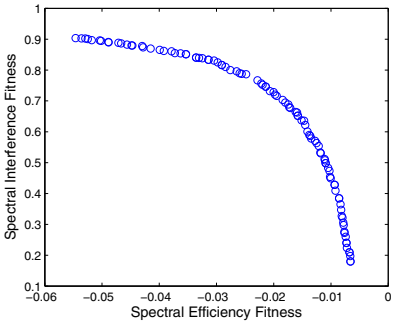
of the plot shows through f_{max_tp} is decreasing with increase in f_{min_power}. Then we have plotted the front between fitness for power minimization (Eq. 7) and fitness for interference minimization (Eq. 8) in figure 2(b). The common parameters between these two functions are power (P), bandwidth (B). If values of both parameters increase, then f_{min_si} decreases and f_{min_power} increases. The front is likely to be linear as both functions are linear. In figure 2(c), we have plotted the front between fitness of power minimization (Eq. 7) and fitness for spectral efficiency maximization (Eq. 9). In this case symbol rate (R_s), modulation index (m) and bandwidth (B) play important role to find out the *pareto-front*.

The pareto optimal fronts obtained by taking fitness for throughput maximization and one from all other fitness functions without any repetition is shown in figure 3. Figure 3(a) represents the optimal front between fitness of interference minimization (Eq. 8) and fitness for throughput maximization (Eq. 6). With decrease in power value, f_{min_si} and probability of bit error rate (P_{be}) increases. f'_{min_tp} value decreases with decrease in P_{be}. This indicates



(a) $f_{max\_throughput}$ Vs $f_{min\_interference}$ (b) $f_{max\_throughput}$ Vs $f_{max\_spec\_efficiency}$

Fig. 3. *Pareto-front* obtained with respect to fitness for throughput maximization



(a) $f_{max\_spec\_efficiency}$ Vs $f_{min\_interference}$

Fig. 4. *Pareto-front* obtained with respect to fitness for interference minimization

the inverse relationship between the two functions. Figure 3(b) represents the front by taking fitness for throughput maximization (Eq. 6) and spectral efficiency maximization (Eq. 9). As P_{be} and energy per bit (γ) depends on m, B, and R_s; so f_{max_se} and f_{max_tp} are dependent on these parameters. With increase in f_{max_tp}, P_{be} decreases and γ increases keeping other parameters constant. Thus the value of f_{max_se} decreases with increase in f_{max_tp} which is depicted in figure 3(b).

At last, in figure 4(a), we have plotted the front between fitness for maximizing spectral efficiency (Eq. 9) and minimizing interference fitness (Eq. 8). As f_{min_si} value increases with decrease in transmission power (P), it results decrease in γ value. Hence f'_{min_se} gradual decreases with increase in f_{min_si}.

5 Conclusions and Future Work

In this work, we modeled the Cognitive Radio parameter adaptation in dynamic spectrum allocation as a multi-objective optimization problem. NSGA-II algorithm is used to solve the problem. The different solution set obtained from the *pareto-front* can be used in the decision model for parameter tuning. To the best of our knowledge this is the first attempt to approach the decision model as a multi-objective optimization problem. The set of solutions obtained through the multi-objective solution approach helps us to get an insight to the problem and to obtain a set of solutions. This probably helps the cognitive radio engine to take the best decision based on requirement, availability and trade-off. In this phase of work, we tried to find optimal *pareto-front* by considering only two scenarios. But in practice, there can be more constraints in real field. For example, some user may want a high throughput and error-free communication with minimum power consumption. Hence for this case, we should consider more than two objectives and optimize them to find optimal solution set. The resulting solution set will help to analyze the nature and behavior of the solution space.

References

1. Mitola III, J.: An integrated agent architecture for software defined radio. Ph.D. dissertation, Royal Institute of Technology (KTH) (May 2000)
2. Etkin, R., Parekh, A., Tse, D.: Spectrum sharing for unlicensed bands. In: Proc. of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access, vol. 25(3), pp. 517–528 (2007)
3. Akyildiz, I.F., Lee, W.Y., Vuran, M.C., Mohanty, S.: NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. Computer Networks Journal 50, 2127–2159 (2006)
4. Spectrum Policy Task Force: Report of the Spectrum Efficiency Workgroup (November 2002), http://transition.fcc.gov/sftp/files/SEWGFinalReport_1.pdf
5. Newman, T.R., Barker, B.A., Wyglinski, A.M., Agah, A., Evans, J.B., Minden, G.J.: Cognitive engine implementation for wireless multicarrier transceivers. Wireless Communication and Mobile Computing 7(9), 1129–1142 (2007)
6. Newmann, T.R.: Ph.D Dissertation on Multiple Objective Fitness Functions for Cognitive Radio Adaptation, pp. 14–69 (2008)
7. Kim, D.I., Hossain, E., Bhargava, V.: Dynamic rate and power adaptation for provisioning class-based QoS in Cellular Multi-rate WCDMA systems. IEEE Transactions on Wireless Communications 3(4), 1590–1601 (2004)
8. Kose, C., Goeckel, D.: On power adaptation in adaptive signaling systems. IEEE Transactions on Communications 48(11), 1769–1773 (2000)
9. El-Saleh, A.A., Ismail, M., Ali, M.A.M., Ng, J.: Development of a cognitive radio decision engine using multi-objective hybrid genetic algorithm. In: Proc. of IEEE 9th Malaysia International Conference on Communications (MICC), pp. 343–347 (December 2009)

10. Paris, J., del Carmen Aguayo-Torres, M., Entrambasaguas, J.: Optimum discrete-power adaptive QAM scheme for Rayleigh fading channels. IEEE Transactions on Communications 5(7), 281–283 (2001)
11. Chung, S.T., Goldsmith, A.: Degrees of freedom in adaptive modulation: a unified view. IEEE Transactions on Communications 49(9), 1561–1571 (2001)
12. Falahati, S., Svensson, A., Ekman, T., Sternad, M.: Adaptive modulation systems for predicted wireless channels. IEEE Transactions on Communications 52(2), 307–316 (2004)
13. Waheed, M., Cai, A.: Cognitive Radio Parameter Adaptation in Multicarrier Environment. In: Proc. of Fifth International Conference on Wireless and Mobile Communications (ICWMC), pp. 391–395 (2009)
14. Zhao, Z., Xu, S., Zheng, S., Shang, J.: Cognitive radio adaptation using particle swarm optimization. Wireless Communication and Mobile Computing 9(7), 875–881 (2009)
15. Ma, J., Jiang, H.: Optimal Design of Cognitive Radio Wireless Parameter on Non-dominated Neighbor Distribution Genetic Algorithm. In: Proc. of Eighth IEEE/ACIS International Conference on Computer and Information Science (ICIS 2009), pp. 97–101 (2009)
16. Yong, L., Hong, J., Qing, H.Y.: Design of Cognitive Radio Wiereless Parameters Based on Multi-objective Immune Genetic Algorithm. In: Proc. of WRI International Conference on Communications and Mobile Computing (CMC 2009), vol. 1, pp. 92–96 (2009)
17. Zhang, Y., Zhang, F., He, W.: Adaptive transmission in cognitive radio networks. In: Proc. of Chinese Control and Decision Conference (CCDC 2009), pp. 1951–1953 (2009)
18. Pratap, A., Deb, K., Agarwal, S., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, Springer, Heidelberg (2000)
19. Proakis, J.G.: Digital Communications, 4th edn. McGraw-Hill, New York (2000)

Fault-Tolerant Relay Placement in Wireless Sensor Networks Using Particle Swarm Optimization

Deepak R. Dandekar[1] and P.R. Deshmukh[2]

[1] B.D. College of Engineering, Sevagram, Wardha, M.S., India
[2] Sipna Engineering College, Amravati M.S., India
d.dandekar@rediffmail.com, prdeshmukh@ieee.org

Abstract. In this paper, we study the problem of deploying relay nodes to provide desired fault tolerance through multi-path connectivity (*k*-connectivity) between all sensor nodes in wireless sensor network. Fault-tolerant relay node placement problem in multi-hop wireless sensor networks, specifically given a disconnected wireless network, few additional nodes is to be deployed so that the augmented network will have higher degree of connectivity for fault tolerance. We propose an algorithm based on evolutionary scheme that places optimum number of energy constrains relay nodes to achieve desired connectivity between homogeneous wireless sensor nodes in which the communication range of each sensor node is same. In this work, simulation results of two algorithms are compared for number of relay nodes required to obtain desired k-connectivity and the average value degree of nodes in resulting network.

Keywords: Wireless sensor networks, relay nodes, *k*-connectivity, fault tolerance, particle swarm optimization.

1 Introduction

Wireless Sensor Networks that consist of a large number of small low cost sensor nodes capable of wireless communication have been widely studied for their applications in many areas like environmental monitoring, field survey, target tracking, disaster relief and so on. Sensor nodes are small in dimensions, usually powered by low-power batteries.

In real applications, unpredictable events such as energy depletion and harsh environmental factors may cause these wireless devices to fail, partitioning the network, and disrupting normal network functions. Therefore, fault tolerance is a critical issue in wireless sensor networks. Usually fault tolerance is achieve by deploying a small number of additional relay nodes in wireless sensor network which provide k vertex-disjoint paths between every pair of node so that the

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 749–757.
springerlink.com

network can survive the failure of fewer than k nodes. This problem is addressed in in the literature [1], [2], [3], [4], [13] as relay node placement.

Since deploying relay nodes can both preserve network connectivity and prolong network lifetime, extensive researches have been focused on it recently. Wireless sensor network including relay nodes is assumed to be single-tiered network and treated the relay node placement problem by deploying a minimum number of relay nodes while making the network connected. In such a way, sensor nodes still need to route data, so the routing load unbalance among sensor nodes is be resolved. Hao et al. [5] took into account the failure of relay nodes, and explored the 2-connected relay node double cover (2*CRNDC*) problem correspondingly. They tried to deploy a minimum number of relay nodes so that, every sensor node can directly communicate with at least two relay nodes and the backbone network consisting only of relay nodes that has 2-connectivity.

To provide fault-tolerance and capacity guarantees by establishing k-connectivity in the network for desired value of k. Approximation and greedy algorithms based on the minimum k-CONNECTEDSUBGRAPH were proposed to achieve *full* k-connectivity.[8]Hisham M. Almasaeid et.al proposed a hybrid algorithm which considers the communication ranges by exploiting intersection regions needed to repair *k*-connectivity to reduce the number of additional nodes in a wireless sensor network. In [7] Juhua Pu and et. al. explored how to add as few as possible nodes (called Steiner nodes) to a sensor network such that the resulting network is *k*-connected (*Full k-conneced)* or partially *k*-connected. *K-connectivity* means that each pair of the nodes, whether Steiner or original, is connected by at least *k* node-disjoint paths, while partial *k*-connectivity only requires such connectivity among original nodes only. S. Misra and et al. [9] have studied a constrained version in which relay nodes can only be placed at a set of candidate locations in a heterogeneous and hybrid network that consists of three types of nodes: *sensors*, *relays*, and *base stations*. The objective was to guarantee *k*-connectivity to a base station with the least number of relays. Partial and full k-connectivity in heterogeneous wireless sensor networks was studied in [1], where sensor nodes possess different transmission radii. For a more detailed and comprehensive report on the current state of research on the minimum relay node placement problem we refer the reader to the survey in [10].

Aziz *et al* in [14] has proposed off-line *PSO-Voronoi* algorithm to minimize the area of coverage holes. The algorithm is for the region-of-interest (ROI) covered by a sensor, then the total coverage. Another application of PSO in stationary node deployment is explained by Hu *et al.*in [15], in which the author proposed centralized *PSO-Traffic* algorithm for topological planning in traffic surveillance application. *PSO-Traffic is* a binary PSO, which is used to minimize a multi-objective fitness parameter. This algorithm improves network performance with saving a system cost by symmetric distribution of high power transmitters.

In this paper, we proposed PSO based algorithm for the minimum relay node placement problem in wireless sensor network. We discuss two algorithms. In the first algorithm, concept of steinerization of edges between two degree deficient nodes is used. Result of this algorithm is compared with proposed PSO based algorithm.

2 Network Model and Problem Specification

We consider stationary symmetric homogeneous multi-hop wireless sensor networks with given a set of sensor nodes that are randomly distributed in a region. In a static network, the devices are stationary. Homogeneous means all the sensors possess the same communication and computing capability with equal initial energy levels. We make the further assumptions that our network is symmetric and that all communication links are bidirectional.

In this paper we suppose that the network is multi-hop, meaning that the mobile devices cooperate to route each others' messages. Thus we are interested in multi-node communication paths between the source and destination of a message. In anticipation of node failures resulting e.g. from power failure or power depletion of a sensor node, we are also interested in finding multiple disjoint communication paths between any sensor to sink.

In order to gather data packets from sensor nodes to the sink, we need to place some relay nodes to forward the data packets, such that there is a connecting path consisting of relay and/or sensor nodes so each hop of the path is no longer than the transmission range of the sensor nodes and the relay nodes. We assume that all sensor nodes and all relay nodes have the same communication radius.

In this work we assume that the network be given as set of vertices V. We place the relay nodes in the network so that the desired level of connectivity between terminal nodes is achieved. The problem can be to find the minimum number of relay nodes needed (and their locations) such that the network (including sensor nodes and relay nodes) is connected and set of nodes V' is k-edge (vertex) connected in the resulting graph G'.

3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedy [11]. This is global gradient-less stochastic evolutionary computation technique which is based on social behavior and movement dynamics of swarm. Each individual in the swarm is referred to as a particle or agent. All the particles in the swarm fly through N-dimensional space and act individually under the same governing principle: accelerate toward the best personal location - personal best(pbest) and their neighborhood's best location - global best (gbest)while constantly checking the value of its current location. Position of the particle is influenced by velocity .All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles.

In the beginning PSO is initialized with a group of random particles (solutions). In PSO space particle can be expressed by N dimensional vector, its position and velocity is expressed as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$, $V_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$ respectively and each i^{th} particle maintain a current position, X_i, current velocity V_i and personal best position pbest. The first gbest is selected from among these initial positions. The fitness function calculates fitness value using the coordinates of the

particle in solution space and returns this value to be assigned to the current location. If that value is greater than the value at the respective pbest for that particle, or the global gbest, then the appropriate locations are replaced with the current location. The velocity of the particle is changed according to the relative locations of pbest and gbest. It is accelerated in the directions of these locations of greatest fitness according to the following equation:

$$v[\ ] = v[\ ] + c1 * rand() * (pbest[\ ] - present[\ ]) + c2 * rand() * (gbest[\ ] - present[\ ]) \quad (1)$$

v[] is the particle velocity
rand() is a random number between (0,1).
c1, c2 are learning factors.

Once the velocity has been determined it is simple to move the particle to its next location. The velocity is applied for a given time-step and new coordinate is computed for each of the dimensions according the following equation:

$$present[\ ] = present[\ ] + v[\ ] \quad (2)$$

The particle is then moved to the location calculated by (2). For each particle in the swarm, the process is repeated and this way the particles move for discrete time intervals before being evaluated. At every time step the positions of all the particles are evaluated, and corrections are made to the positions of pbest, and gbest before letting the particles fly around for another step. Repetition of this cycle is continued until the fixed number of iterations *k*max is reached.

4 Fault Tolerant Relay Placement Algorithms

In this Section, we briefly describe the two algorithms Steiner_K-Connect and PSO_K-Connect for the minimum relay nodes placement problem for desired k-connectivity in the wireless sensor network for fault tolerance.

4.1 Steiner_K-Connect Algorithm

The objective of algorithm is to deploy a minimum number of relay nodes to form minimum k-vertex connected network such that in the resulting network, there exist minimum k vertex-disjoint paths from any target node to any other target node or relay nodes previously deployed. In brief, given a set of target nodes V, for degree deficient target nodes u and v, algorithm create a path between u and v, placing as few relay nodes as possible, while ignoring all the other target nodes and any previously deployed relay nodes. To facilitate creating such paths the algorithm first computes weight $W(u,v)$, such that $W(u,v) = 0$ if transmission range of u, $T(u) = T(v) = d(u,v)$ and $W(u,v) = ceil(\ d(u,v)/\ T(u))$ and then sort weight vector in ascending order for all target nodes in V, then we calculate degree of a node as it equal to number of zeros in weight vector. If degree of u and v found less than minimum degree specified then for the first $W(u,v) > 1$ which is minimum in sorted result, place one relay node on the straight line between u and

v such that $d(u,v)=T(u)$ and then, evenly place $W(u,v) - 1$ relay nodes along the straight line between u and v. In this way, create a path from u to v. This procedure described in algorithm 1, is near similar to steinerization of edges described in One-Way one-way partial fault-tolerant relay node placement algorithm for Minimum k-Vertex Connected Spanning Graph [1]. Iteratively increase the degree of each node in V to form defined minimum k-vertex connected network.

$$W(u,v) = \begin{cases} 0 & if\ d(u,v) \leq T(u) \\ ceil(\ d(u,v) / T(u)) & if\ d(u,v) > T(u) \end{cases} \quad (3)$$

Algorithm 1: ***Steiner_K-Connect***

1: INPUT: Integer k and a set of sensor nodes *V*.
2: OUTPUT: A set of relay nodes *R*.
3: $R \leftarrow \phi$ (empty set);
4: $d(u,v) \leftarrow \{\ |uv|\ ; u,v \in V\ \}$;
5: Define the weight of each edge $uv \in W$ according to Equation 3;
6: $G \leftarrow (V \cup R, W)$;
7: Draw a spanning graph *G*;
8. Compute degree of each Vertex and derive *G*' such that it consist of nodes having degree less than k defined;
9: Steinerize edge between first node *u* and its nearest node *v* in *G'* by placing relay nodes and add relay nodes into R;
10. Repeat step 4 to 9 till $G' \leftarrow \phi$ (empty set);
11: Output *R*;

4.2 PSO_K-Connect Algorithm

Algorithm2 mentioned is to find optimum number of relay nodes placed to form minimum k-vertex connected network using particle swarm optimization. This algorithm work for a set of target nodes V which is not k-vertex connected network but in the resulting network, there exist minimum k vertex-disjoint paths from any target node to any other target node or relay nodes. The initial steps of this algorithm is identical to steps 1 to 8 of Steiner_K-Connect algorithm, in which we first computes sorted weight *W(u,v)* vectors for all target nodes in V and its degree which is equal to the number of zeros in weight vector. The list of degree deficient nodes is created is then passes to PSO algorithm. The PSO works for each degree deficient node in the list to place a relay node with the objective of particle swarm population to minimize sum of weights all target node in the list with newly placed relay and to maximize connectivity of target node in the list. This procedure repeated until the list is vacant and all target nodes attains minimum desired k-connectivity.

Algorithm 2: ***PSO_K-Connect***

1: INPUT: Integer k and a set of sensor nodes *V*.
2: OUTPUT: A set of relay nodes *R*.
3: $R \leftarrow \phi$ (empty set);
4: $d(u,v) \leftarrow \{ \mid uv \mid ; u,v \in V \}$;
5: Define the weight of each edge $uv \in W$ according to Equation 3;
6: $G \leftarrow (V \cup R, W)$;
7: Draw a spanning graph *G*;
8. Compute degree of each Vertex and derive *G*' such that it consist of nodes having degree less than k defined;
9: Run PSO algorithm on G' to find position of relay node to be placed Between first node *u* and its neighbor nodes in *G'*;
10. Repeat step 4 to 9 till $G' \leftarrow \phi$ (empty set);
11: Output *R*;

5 Simulation Results

This section evaluates performance of proposed algorithm described in section 4. MATLAB is used for implementation and simulation of algorithms. We randomly deployed homogeneous type of sensor nodes in a 500m x 500m, 2D terrain. Transmission radius of all sensor nodes sets to 100 m.

The simulation is performed for two different values of k. In first set of simulation we fixed k=2 and then increase number of sensor nodes from 10 to 100. After each simulation the number of relay nodes and average degree of nodes are computed. Then we repeated the procedure for k=3.

Here we are defining a new parameter," Fault Tolerance Capacity" of algorithm for fault tolerance relay placement problem. We have calculated fault tolerance capacity as a ratio of average degree value provided by resultant minimum k-vertex graph to number of relay nodes placed by the algorithm. This parameter is useful for comparing effectiveness of algorithms in placing relay nodes to provide better fault tolerance.

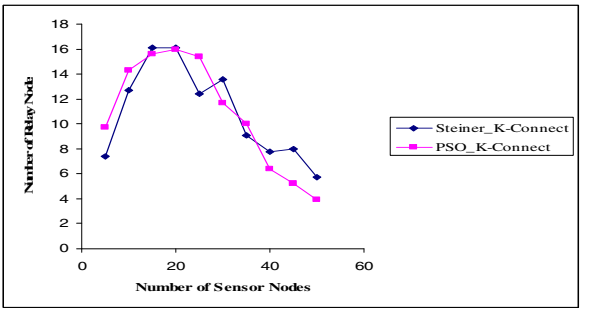


Fig. 1. Performance Comparison of algorithm For K=2

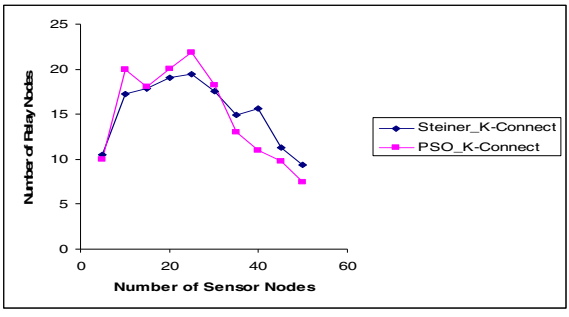


Fig. 2. Performance Comparison of algorithm For K=3

The comparisons of the performances of both the algorithms are shown in Fig 1, 2 and 3. We have the fallowing observations.

1. Number of relays placed for desired K-vertex connectivity is initially increased and then decreased with increasing number of sensor nodes in all the values of k for both the algorithms.
2. Initially for smaller number of sensor nodes, the relay nodes placed by PSO_K-Connect algorithm are in greater number than that of Steiner_K-Connect.
3. Later as number of sensor nodes increases, the relay nodes placed by PSO_K-Connect algorithm are goes on decreasing sharply as compared to Steiner_K-Connect algorithm.
4. Fault tolerance capacity comparison shown in fig. 3 clearly indicates that for any value of k, PSO_K-Connect algorithm performs better than Steiner_K-Connect.

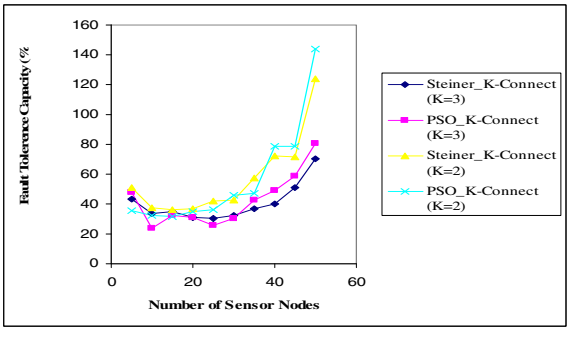


Fig. 3. Fault Tolerance Capacity Comparison

6 Conclusion and Future Work

In this work we proposed a fault tolerant relay placement algorithm for desired K-connectivity which is based on evolutionary optimization scheme which works on swarm intelligence principle. Proposed PSO based algorithm is to solve the problems of deploying minimum number of relay nodes to achieve desired fault tolerance in homogeneous wireless sensor network, where sensor nodes are homogeneous in sense of their transmission power. In performance comparison of two algorithms, the proposed PSO based algorithm, PSO_K-Connect proved better than Steiner_K-Connect, the steinerization based algorithm. Proposed algorithm comparatively uses lower number of relay nodes deployed to attain desired k-connectivity in wireless sensor network but in case of average degree value of the resulted network, steinerization based algorithm is found better. In the future, we would like to conduct similar experiments to address the optimization of relay node placement for fault tolerance in heterogeneous wireless sensor network wherein transmission range of each node is different.

References

[1] Han, X., Cao, X., Lloyd, E.L., Shen, C.-C.: Fault-Tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks. IEEE Transactions on Mobile Computing 9(5) (May 2010)

[2] Kashyap, A., Khuller, S., Shayman, M.: Relay Placement for Higher Order Connectivity in Wireless Sensor Networks. In: INFOCOM, April 1-12 (2006)

[3] Lloyd, E.L., Xue, G.: Relay Node Placement in Wireless Sensor Networks. IEEE Trans. Computers 56(1), 134–138 (2007)

[4] Cheng, X., Du, D., Wang, L., Xu, B.: Relay Sensor Placement in Wireless Sensor Networks. Journal of Wireless Networks 14(3), 347–355 (2008)

[5] Hao, B., Tang, H., Xue, G.: Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation. In: Proc. IEEE, Workshop on High Performance Switching and Routing, pp. 246–250 (2004)

[6] Cheng, X., Du, D.-Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. Journal on Wireless Networks 14(3) (2008)

[7] Pu, J., Xiong, Z., Lu, X.: Fault-tolerant deployment with k-connectivity and partial k-connectivity in sensor networks. Journal Wireless Communications and Mobile Computing 9(7) (July 2009)

[8] Almasaeid, H.M., Kamal, A.E.: On the Minimum k-Connectivity Repair in Wireless Sensor Networks. In: IEEE International Conference on Communication, pp. 1–5 (2009)

[9] Misra, S., Hong, S.D., Xue, G., Tang, J.: Constrained Relay Node Placement in Wireless Sensor Networks: Formulation and Approximations. ACM/IEEE Transaction on Networking 18(2) (2010)

[10] Degener, B., Fekete, S.P., Kempkes, B., Heide, F.M.A.D.: A survey on relay placement with runtime and approximation guarantees. Computer Science-Review, 57–68 (2011)

[11] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)

[12] Robinson, J., Rahmat-Samii, Y.: Particle Swarm Optimization in Electromagnetics. IEEE Transactions on Antennas and Propagation 52(2) (February 2004)
[13] Kulkarni, R.V., Venayagamoorthy, G.K.: Particle Swarm Optimization inWireless Sensor Networks: A Brief Survey. IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews 41(2), 262–267 (2011)
[14] Aziz, N.A.B.A., Mohemmed, A.W., Sagar, B.S.D.: Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization. In: Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS), pp. 961–965 (2007)
[15] Hu, J., Song, J., Zhang, M., Kang, X.: Topology optimization for urban traffic sensor network. Tsinghua Science & Technology (April 13, 2008)

A Novel Approach for Web Services Discovery Using Rough Sets

Ami Choksi[1] and Devesh Jinwala[2]

[1] C.K. Pithawalla College of Engg. & Tech., Surat, Gujarat State
[2] S.V. National Institute of Tech., Surat, Gujarat State
amichoksi.phd@gmail.com, dcjinwala@acm.org

Abstract. The Service Oriented Computing has gained vital significance today. Web services are aimed to provide platform independence, loose coupling, self-description, and efficient discovery. However, discovering the appropriate service efficiently, from amongst the proliferation of numerous ones in computational grid is a challenging task. There have been numerous attempts at devising suitable protocols for doing so, and many more are being developed still. In this paper, we present an approach based on using rough sets for web service discovery that handles uncertainty of reducible properties. We use semantic approach for functionality and input/output matching for irreducible properties. To the best of our knowledge, ours is an improved and a novel approach for doing so.

Keywords: Functionality matching, grid computing, rough set theory, semantic web, service oriented computing.

1 Introduction

Web services are *self-contained*, *self-describing, interoperable* software components that find the predominant use in application integration and component based application development. With the growing penetration of the Internet, the usage of web services for application integration has grown significantly. However, *continually changing* and *evolving* services, *perennial heterogeneity*, and *lack of constant availability* make discovery of the appropriate service non-trivial and a challenging task. This has grown severer with the proliferation of numerous service providers providing multitude of services.

Obviously, the discovery of appropriate service has to be based on a match making between the advertisement of the available service by the service provider and the description of the desired service by the service solicitor. Advertisement of services is typically based on registration of the properties of the services in a globally visible service registry. Service providers independently publish services using their predefined properties.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 759–771.
springerlink.com

The operational semantics of the match making lead to two different approaches in service descriptions: *non-semnatic web service descriptions* (e.g. those based on a typical Web Service Description Language [1]) based on syntactical matching of the service key words OR *semantic web services* based on the semantic matching using ontological description of the defined services.

Irrespective of the syntactic or semantic matching, the defined attributes of the services play a vital role in implementing match making and eventual discovery. The service attributes may be based on their functional and non-functional properties at the coarser level[2]. The example of functional properties are inputs, outputs, precondition, effects and that of the non-functional properties are cost, efficiency, reliability, security [3][4][5].

Services may be described using reducible and irreducible properties. *Inputs, outputs, preconditions, service name etc.* functional properties of services are considered as irreducible properties. Such properties are the mandatory properties of services. *Cost, efficiency, reliability, security, availability* etc. non-functional properties are considered as reducible properties.

There can be different ways of handling uncertainty e.g. probability theory, fuzzy logic, rough set theory. However, we observe that reducible and irreducible property descriptions using the rough set theory has been gaining significant importance [6][7]. Especially, the uncertainity of the defined service property due to the independent publication of the services by the service advertisers, can be handled using rough set theory. Irreducible properties can be matched using functionality match [8] at higher priority using RiWordnet [9] to get the synonyms of functionality words. Rough set theory is used for reducing irrelevant and dependent reducible properties. I/O match is done using semantic matching of irreducible properties.

We observe that application of rough set theory to all types of properties may give good number of hits, but will create problem while executing web services. Missing or wrong mandatory functional properties lead to fail the service execution. We believe web services properties should be considered as reducible or irreducible while applying rough set. In this context, we propose here a novel approach for web service discovery exploiting reducibility based on Rough Set theory. Our approach involves functionality matching with RiWordnet, QoS matching with rough set and I/O match using semantic matching. Functionality matching using Riwordnet reduces the services which are irrelevant. Rough Set approach is used to reduce irrelevant and dependent service properties. Lower approximation of rough set outputs a list of services that are having exact match to the requested service.

In section 2, we describe the theoretical background of the approaches. In section 3, we describe our proposed algorithm. Tools, languages, and test application used to implement our algorithm are described in section 4. Performance and analysis are available in section 5.

2 Theoretical Background and Related Work

In this section, we present the theoretical background and the existing related work in the literature. We use the following conventions in further description: advertised and requested services are depicted by *A* and *R* respectively. P1, P2… Pm depict the properties for advertised services whereas P1, P2…Pn depict the properties for the requested service.

Service match making for discovery can be either exact, subsume, plug-in and no-match as described below [10][11][12] :

- **Exact match:** If all the properties of the advertised service match with that of requested one is an exact match i.e. A (PROPERTIES) ≡R (PROPERTIES) → {A (Pi) ≡ R (Pi)}.
- **Subsume:** If all the properties of the advertised service are less than that of the requested service i.e. A (PROPERTIES) <= R (PROPERTIES) → {A (Pi) ⊆ R (Pi)}.
- **Plug-in:** If all the properties of the advertised service are greater than that of the requested service i.e. A (PROPERTIES) >=R (PROPERTIES) → {R (Pi) ⊆ (A (Pi))}.
- **No match:** If the properties of the advertised service and that of requested service properties don't have any of the above relation.

Finding match can be done using semantic similarity [13][14][15][16] and/or Semantic distance [17][18] as we discuss below.

- **Semantic Similarity:** A similarity function is defined as a real valued function i.e. . sim(x, y) : C × C -> [0, 1] . On a set measuring the degree of similarity between x and y, where C is the set of well-formed concepts, sim(x, y) is a measure the degree to which y is similar to x; sim(x, y) = 1 means fully similar, namely the case x = y; sim(x, y) = 0 means no common between the two concepts.

$$sim(x,y) = p(\frac{|A(x) \bigcap B(x)|}{A(x)}) + (1-p)(\frac{|A(x) \bigcap B(x)|}{A(y)})$$

- **Semantic Distance:** Semantic distance [17][18] is the degree of meaning relatedness between concepts or measure of similarity between two meanings.
- **Precision ratio:** The ratio of relevant retrieved services to the total number services retrieved is precision ratio [19].
- **Recall ratio:** Recall ratio [19] is the fraction of the services that are relevant to the query that are successfully retrieved.
- **Indiscernibility:** Information system I = (U, A), U-nonempty set of objects and A-nonempty finite set of attributes such that a: U -> Va, for a∈A. By P ⊆A, there is an associated equivalence relation called indiscernibility IND (P) defined as $IND(P) = \{(x,y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}$

- **Lower approximation:** Lower approximation or positive region is the union of all equivalence classes in [x]p which are subsets of the target get i.e $\underline{P}X = \{x \mid [x]_p \subseteq X\}$
- **Upper approximation:** Upper approximation or negative region is the union of all equivalence classes in [x]p which have non-empty intersection with the target set i.e. $\overline{P}X = \{x \mid [x]_p \bigcap X \neq \varnothing\}$
- **Accuracy of rough set classification:** Accuracy of rough set classification can be calculated as a ratio of cardinality of lower approximation to that of upper approximation i.e. $\alpha_p(X) = \frac{\underline{P}X}{\overline{P}X}$
- **Boundary region:** The boundary region is given by the set difference of upper and lower approximation sets.
- **Reduct:** A subset of attributes which can, by itself, fully characterize the knowledge in the information system, such an attribute set is called reduct.
- **Core:** The set of all indispensable relation of I is called the core of C. $CORE(C) = \bigcap RED(C)$

In the following, we discuss the related attempts in web services discovery:

The published literature can be categorized into two distinct areas: incrementally improving algorithms for web services discovery without the Rough Set based approach [8][10][11][20][21][22][23][24][25[26][27] and those that aim to do so using Rough Set based approach. Our focus here is only on the latter, as we believe the same is a better way of ensuring greater match when discovery a web service and for the uncertainties associated with the service properties.

The author in [29] first introduced the notion of rough set based match making algorithm. In [30], the indiscernibility relation of rough set is used as a similarity measure for match making that improves match making. In [31], *non-reduct* attributes are ranked for their contribution in the cluster. The pattern is then formed by the conjunction of most contributing attributes of that cluster. The algorithm in [32] induces minimal rules from a decision table. The algorithm also proposes an approach to classification unseen objects using rules w.r.to upward/downward unions of decision classes. The algorithm in [33] considers rough set based resource discovery considering weighted matching of QoS parameters and lower approximation class as a positive match of requested and advertised services. The algorithm in [34] uses rough sets and matches using different rules considering the depth of the subclass relation between requested and advertised resources in the ontology tree.

All the above attempts use rough sets to reduce the dataset, as per observations functional properties should not be reduced, as these are the mandatory properties of services for execution whereas the non-functional properties may be reduced. Our approach focuses on the same, as we discuss below.

3 The Proposed Approach

We use a multi-strategy approach for service discovery/matching. The outline of our approach is shown in Fig-1. The dataset of semantic web services is OWLS-TC4 [35]. To improve the effect of the dataset we have added *Quality of Service* parameters viz. Semantic functionality matching using RiWordnet, QoS parameters matching using rough set and Input/output matching in terms of their confidence level in percentages.

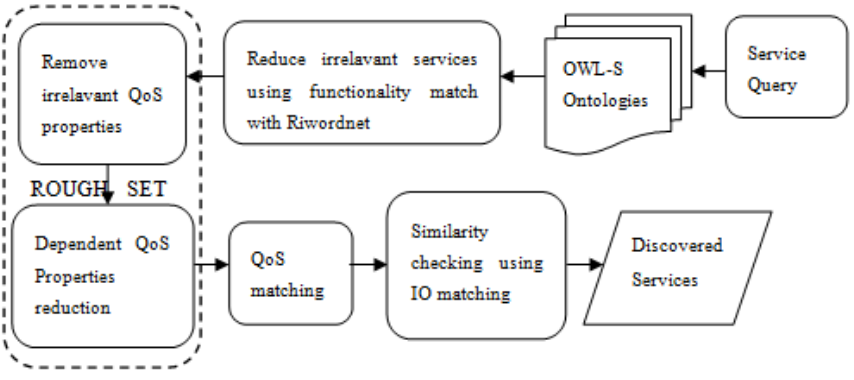


Fig. 1. Our multi-strategy approach of service discovery in semantic grid/web

3.1 Semantic Functionality Matching Using RiWordnet

Input, output properties can be similar for many services irrespective of having different purpose and domain. On the other hand, the functionality can differentiate services easily, so functionality matching is given higher priority [8]. As for example, division and addition services of two numbers have the equal *number* and similar *type* of input and output parameters, but their functionalities are essentially different. Hence, the designated name of functionality can differentiate such services. However, amidst heterogeneous participating entities in the computing grid, the syntactically given names of the same services from different publishers are expected to be different even though they serve to achieve the same functionality. Hence, it is essential to emphasize a *semantic match.* We use a published database (e.g. RiWordnet) lookup for finding the meaning of the terms used to depict the services and use the same in either accepting or rejecting a match. We use here synonym and antonym look up for the same. The non-matching services are eliminated upfront, while the matching ones are further investigated for refined match. Thus, we remove few irrelevant services straightaway.

3.2 QoS Parameters Matching Using Rough Set Approach

Uncertainty of service properties can be there in service description as a large number of service publishers advertises service properties indifferently. Rough set approach is the best mathematical tool to handle the uncertainty [6]. Rough set approach, first removes irrelevant services by considering only the queried properties from advertised services' properties. Dependent properties removal includes removal of indecisive properties. Here we consider null values of QoS as indecisive properties [29]. Instead of all properties for all the advertised services checked at a time, we have considered removing attributes while checking the maximum null value at the interval of every 100 services. The basic flow of the match discovery is illustrated in Fig.2

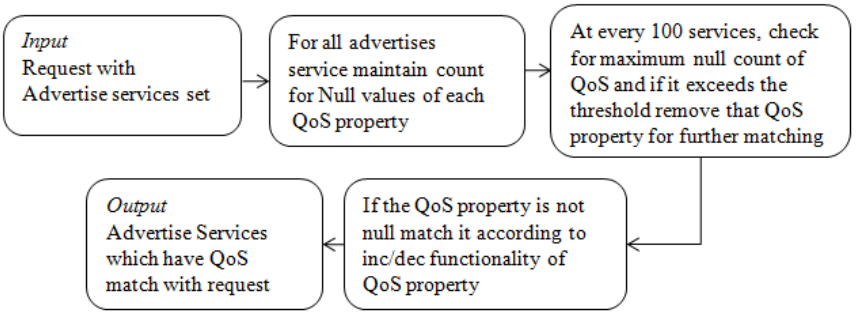


Fig. 2 (a). QoS matching flow

```
reliability:high, guaranteedbandwidth:high, delay:low, utilization:high, packetloss:low, security:high, ubiquity:high
```

Fig. 2 (b). QoSparam.txt for defining whether QoS parameters are increasing/decreasing

We only consider QoS parameters i.e. reducible properties for irrelevant and dependent properties removal. As regards to I/O matching, we give priority to output matching, because only if all the output properties of the solicited service are matched, then only it becomes meaningful to check for input parameters.

4 Pseudo Code

The proposed algorithm using functionality matching with RiWordnet, rough set based QoS matching and input, output matching is as shown in Fig. 3(a) –(e).

```
Input: Advertise services dataset ADV[1□N] and Query Q
Output: Advertise services which matches with Q
ADV consists:A(I1, I2, I3,…Im) input parameters
A(O1, O2, O3,…On) output parameters
A(Q1,Q2, Q3,…Qk) QoS parameters
Request or query is considered as having
R(i1,i2,i3,…in) input parameters
R(o1, o2, o3,…oq) output parameters
R(q1,q2, q3,…ql) QoS parameters
  Functionality name is say Fa & Fr for A and R
```

Fig. 3(a). The Proposed Algorithm: Service properties declaration

```
1.For i<-0 to length[ADV]
2.     Do
3.          If(FunctionalityMatch()==true)
4.                  IF(RoughSetQoSMatch()==true)
5.       IOMatch();
```

Fig. 3(b). The Proposed Algorithm: Main method

```
1.//Functionality matching
2.Function FunctionalityMatch()
3.  Begin
4.  If No "_" check for direct synonym match of Fr and Fa and re-
turn result
5.  Make parts by "_" of Functionali-ty(both request and adver-
tise service) If it contains "_"
6.  Check all noun parts matching
7.  If antonyms of parts of Fr contains part of Fa return no
match
8.  Check if all noun request functionality parts' synonyms match
with those of Fr return match
9.  End //Function FunctionalityMatch
```

Fig. 3(c). The Proposed Algorithm: Functionality matching

```
1.  Function IOMatch()
2.  Begin
3.  OutputsList aOLst = getOutputs(adv);
4.  OutputsList rOLst = getOutputs(req);
5.  If(aOLst.containsAll(req)) //Output match
6.  InputList aILst = getIn-puts(adv);
7.  InputList rILst = getIn-puts(req);
8.  If(aILst.containsAll(req)) //Input match
9.  Lowerapprox.add(adv);
10. End //Function IOMatch
```

Fig. 3(d). Input/Output matching

```
1.   //Rough set approach to remove irrelevant & dependent proper-
     ties.
2.   Function QoSMatching()
3.   Begin
4.   Initialize QoSCount [1..l] =0;
5.   //Checking for request and all advertised services
6.   //QoSparams = (q1,q2, q3,…ql)
7.   Check for all QoS param of request and advertise services
8.   If advertise service QoS is null
9.   Maintain count of that QoS
10.  At every 100 advertise services check the max null QoS param
     and if it crosses threshold remove that QoS property for fur-
     ther match
11.  Else// Check if QoS param is increasing/decreasing
12.  QoS_inc_dec(QoS param)
13.  if(QoS_inc_dec(qi)==true)//high value of qi preffered
14.  if(A[qi]<R[qi])
15.  return false;
16.  else if(A[qi]>R[qi])
17.  return false;
18.  return=true;
19.  end //Function QoSMatching
20.  function QoS_inc_dec(QoSparam)
21.  begin
22.  read file "QoSparam.txt" line by line;
23.  if(line contains QoSparam)
24.  high_low=take substring of line after ':'
25.           if(high_low=="high")  return true;
26.           else return false;
27.  end //QoS_inc_dec
```

Fig. 3(e). QoS matching with rough set

5 Methodology of Implementation

Quality of service parameters like security, cost, performance, manageability [36] can be described in services using different QoS based semantic web services languages [37] e.g. OWL-Q, OnQoS, QoSOnt, WSMO-QoS, QoS-MO, DAML-QoS, WS_QoSOnto [38]. OWL-Q though considered one of the better language, suffers from the limitation of not supporting QoS priority and independence of QoS parameters [38]. In addition, none of these QoS based language's API or simulator is available openly for implementation. And, OWL-S is a web service description language that allows us to add QoS or any other parameters in services. Hence, so we describe web services using OWL-S language. We have used Java language [39][40][41], Netbeans IDE [42], Apache web server [43] and OWL-S API [44] for our implementation.

We use the following evaluation metrics for comparing algorithm with and without rough set:

- **Time of discovery:** Total time to find set of services matching requested service.
- **Number of services discovered:** Total number of services discovered by the match algorithm.
- **Recall ratio:** Recall ratio is a ratio of number of relevant retrieved services to that of relevant services. i.e.
  |(relevant services ∩ retrieved services)|/|relevant services|

5.1 Experimental Setup

We use Java (the language suited for distributed and multi-threaded applications), Netbeans IDE (platform independent and open source IDE for Java), Apache web server for our implementation. We also use OWL-S API (to describe web services), OWLS-TC4 dataset (a semantic web service collection of 1083 services of nine domains: namely education, medical care, food, travel, communication, economy, weapon, geography, simulation [35]) and RiWordnet API for the purpose already described earlier.

5.2 Test Application

Among the domains of services OWLS-TC4 provides, we feel medical care system domain can have maximum critical QoS parameters that can be exploited for our experimentations. Hence, we have chosen the medical care system as test application. We use *reliability, guaranteed bandwidth, delay, utilization, packet loss, security, and ubiquity* as the seven important seven QoS parameters of medical care system [45][46].

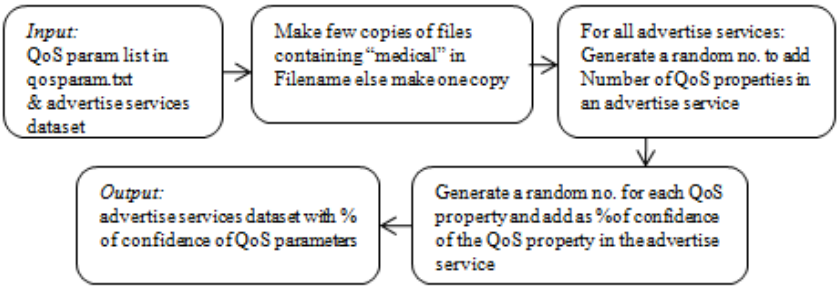


Fig. 4. Quantifying QoS parameters to advertise services

The OWLS-TC4 includes medical care system services, but does not include quality of parameters. We have made few copies of web services of each medical OWL-S file for advertised services, so can test for matching algorithm well. We have added random but less than or equal to total QoS parameters (here 7) to queries and advertised services. Thus, we have generated dataset where we can find the uncertainty of QoS parameters.

6 Performance Results and Analysis

From the performance results as shown in Table.1, we can say, the algorithm without a rough set does not give results in many cases whereas the algorithm with rough set gives matching results. We get improvement in 100% discovered services and improved recall ratio by giving penalty of only 17% execution time. The reason of better results is: algorithm with rough set ignores null value and check for other QoS parameters for match, whereas algorithm without rough set returns "no match" for the same.

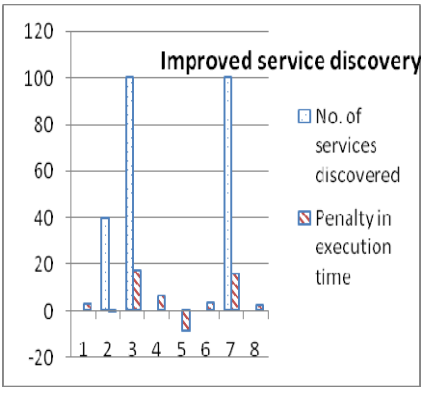
Null values are considered as +ve match because when QoS parameters are not specified does not mean its zero or not matching. The query and few examples of advertised services properties are in Table.2.

Table 1. Results of algorithm with and without rough set

Query Service (.Owls file)	Algo without a rough set				Algo with rough set				improved result in no. of services %	penalty in execution time %
	No.of service match	Execution times (MS)	Precision	Recall	No.of service match	Execution times (MS)	Precision	Recall		
CAD_medical_service	2	30017	1	1	2	31011	1	1	0	3.311
medicalclinic_diagnosticprocess_service	5	31690	1	0.714	7	31569	1	1	40	-0.381
medicalclinic_diagnosticprocesstimeduration_service	3	27419	1	0.5	6	32178	1	1	100	17.356
medicalclinic_diagnosticprocesstimeinterval_service	7	29321	1	1	7	31230	1	1	0	6.510
medicalclinic_diagnosticprocesstimemeasure_service	0	32495	1	1	0	29873	1	1	0	-8.068
medicalclinic_experimenting_service	0	28540	0	0	1	29528	1	1	0	3.461
medicalclinic_investigating_service	1	26148	1	0.5	2	30358	1	1	100	16.100
_medicaldoctor_UNOservice	3	28991	1	1	3	29821	1	1	0	2.862

Table 2. QoS data of few query and matching services using our algorithm

Query/Advertise service(.owls file)	Ubiquity	Guaranteed bandwidth	Utilization	Reliability	Security	Packet loss	Delay
medicalclinic_diagnosticprocess_service	15	16	55	44	Null	Null	Null
medicalclinic_diagnosticprocess_service6	25	Null	91	Null	Null	Null	Null
medicalclinic_diagnosticprocess_service18	42	Null	91	Null	Null	Null	Null
medicalclinic_experimenting_service	29	75	56	30	97	34	71
medicalclinic_experimenting_service7	97	Null	94	Null	Null	Null	Null



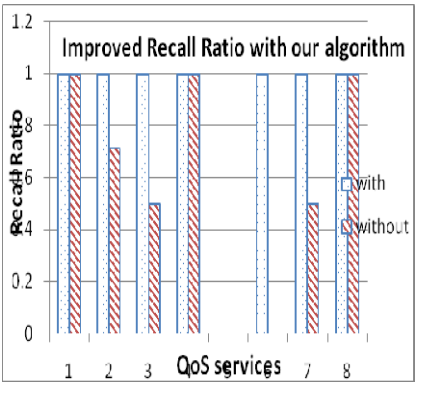


Fig. 5. Improved results in number of web services & recall ratio by fewer penalties in execution time

7 Conclusion and Future Work

In this paper, we have presented a multi-strategy approach to service discovery. The algorithm that we propose, dynamically discovers the set of services by functionality matching with Riwordnet, lower approximations of rough set and IO matching.

From the performance results, we observe that there is indeed an increased number of services discovered and improved recall ratio by giving fewer penalties in execution time compared to the existing algorithm without rough set.

We have used the IO parameters as irreducible parameters in our algorithm, i.e. we do not apply rough set to them.

As part of our future work, we intend to further improve the algorithm by using heuristics to reduce the execution time of the algorithm. In addition, we also intend to incorporate the precondition based matching as well as using a combination of services when a single service is not found as a match.

References

[1] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1 (2011), http://www.w3.org/TR/wsdl

[2] Badr, Y., Abraham, A., Biennier, F., Grosan, C.: Enhancing Web Service Selection by User Preferences of Non-functional Features. In: Proceedings of the 4th International Conference on Next Generation Web Services Practices (NWESP 2008), pp. 60–65 (2008)

[3] Keller, Lausen (eds.): Functional Description of Web Services (2011), http://www.wsmo.org/TR/d28/d28.1/v0.1/

[4] O'Sullivan, J., Edmond, D., ter Hofstede, A.H.M.: Formal description of non-functional service properties. Technical Report FIT-TR-2005-01 (2005)

[5] Peters, G., Weber, R., Crespo, F.: Uncertainty modeling in dynamic clustering - A soft computing perspective. In: IEEE International Conference on FUZZ-IEEE, pp. 1–6 (2010)

[6] Pawlak, Z.: Some Issues on Rough Sets. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 1–58. Springer, Heidelberg (2004)

[7] Pawlak, Z.: Rough sets. International Journal of Parallel Programming 11(5), 341–356 (1982)

[8] Thamarai Selvi, S., Balachandar, R.A., Vijayakumar, K., Mohanram, N., Vandana, M., Raman, R.: Semantic Discovery of Grid Services Using Functionality based Matchmaking Algorithm. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, pp. 170–173 (2006)

[9] RiWordnet (2011), http://www.rednoise.org/rita/wordnet/documentation/docs.html

[10] Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic Matching of Web Services Capabilities. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 333–347. Springer, Heidelberg (2002)

[11] Thamarai Selvi, S., Balachandar, R.A., Vijayakumar, K., Raman, R., Mohanram, N.: Semantic Matchmaking of Grid Services using Parameter Matchmaking Algorithm.

In: Proceedings of IASTED International Conference on Computational Intelligence, pp. 43–48 (2006)
[12] Srinivasan, N., Paolucci, M., Sycara, K.: Adding OWL-S to UDDI, implemen-tation and throughput. In: Proceedings of 1st International Workshop on Semantic Web Services and Web Process Composition, pp. 6–9 (2004)
[13] Thiagarajan, R., Manjunath, G., Stumptner, M.: Computing Semantic Simi-larity Using Ontologies. HP Labs Tech Report HPL-2008-87 (2008)
[14] Andrea Rodríguez, M., Egenhofer, M.J.: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering (15), 442–456 (2003)
[15] Han, L., Berry, D.: Semantic-supported and agent-based decentralized grid resource discovery. In: Proceedings of Future Generation Computer Systems, pp. 806–812 (2008)
[16] Andreasen, T., Bulskov, H., Knappe, R.: From ontology over similarity to query evaluation. In: Bernardi, R., Moortgat, M. (eds.) 2nd CoLogNET-ElsNET Symposium - Questions and Answers: Theoretical and Applied Perspectives, pp. 39–50 (2003)
[17] Hjørland, B.: Semantic distance (2010), `http://www.iva.dk/bh/lifeboat_ko/CONCEPTS/semantic_distance.html` (July 17, 2006)
[18] Goranson, T.: Semantic Distance (2010), `http://www.eil.utoronto.ca/ICEIMT04/goranson.pdf` (2004)
[19] Recall and Precision ratio (2010), `http://en.wikipedia.org/wiki/Precision_and_recall`
[20] Şenvar, M., Bener, A.B.: Matchmaking of Semantic Web Services Using Semantic-Distance Information. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 177–186. Springer, Heidelberg (2006)
[21] Liu, M., Shen, W., Hao, Q., Yan, J.: An weighted ontology-based semantic similarity algorithm for web service. Journal of Expert Systems with Applications 36(10), 12480–12490 (2009)
[22] Zhang, W., Li, Y., Liu, F., Ma, F.: Ontology-Driven Resource Selecting in the Grid Environments. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006, Part I. LNCS, vol. 3991, pp. 818–821. Springer, Heidelberg (2006)
[23] Akkiraju, R., Goodwin, R., Doshi, P., Roeder, S.: A Method for Semanti-cally Enhancing the Service Discovery Capabilities of UDDI. In: Proceedings of IJCAI 2003 Workshop on Information Integration on In IIWeb, pp. 87–92 (2003)
[24] Corella, M.Á., Castells, P.: A Heuristic Approach to Semantic Web Services Classification. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 598–605. Springer, Heidelberg (2006)
[25] Karimpour, R., Taghiyareh, F.: Conceptual discovery of Web services using WordNet. In: Proceedings of IEEE APSCC, pp. 440–444 (2009)
[26] Thamarai Selvi, S., Balachandar, R.A., Swaminathan, V., Paramasivan, V., Sampath, A.: Semantic Description and Discovery of Grid Services Using WSDL-S and QoS based Matchmaking Algorithm. In: Proceedings of 14th International Conference on Advanced Computing and Communications, pp. 113–116 (2006)
[27] Bellur, U., Kulkarni, R.: Improved Matchmaking Algorithm for Semantic Web Services Based on Bipartite Graph Matching. In: Proceedings of IEEE International Conference on Web Services, pp. 86–93 (2007)

[28] Bener, A.B., Ozadali, V., Ilhan, E.S.: Semantic matchmaker with precondi-tion and effect matching using SWRL. Journal of Expert Systems with Applications 36(5), 9371–9377 (2009)
[29] Li, M., Yu, B., Huang, C., Song, Y.-H., Rana, O.F.: Service Matchmaking with Rough Sets. In: Proceedings of 6th IEEE International Symposium on Cluster Computing and the Grid, pp. 123–130 (2006)
[30] Upadhyaya, S., Arora, A., Jain, R.: Rough Set Theory: Approach for Similar-ity Measure in Cluster Analysis. In: Proceedings of the International Conference on Data Mining, pp. 353–356 (2006)
[31] Arora, A., Upadhyaya, S., Jain, R.: Integrated approach of reduct and clustering for mining patterns from clusters. Journal of Inform. Technol. 8(2), 173–180 (2009)
[32] Khoo, L.P., Tor, S.B., Zhai, L.Y.: A Rough-Set-Based Approach for Classification and Rule Induction. The International Journal of Advanced Manufacturing Technology 15(6), 438–444 (1999)
[33] Li, M., Yu, B., Rana, O., Wang, Z.: Grid Service Discovery with Rough Sets. IEEE Transactions on Knowledge and Data Engineering 20(6) (2008)
[34] Ataollahi, I., Analoui, M.: Resource discovery using rough set in grid environment. In: The Proceedings of 14th International Conference of CSI, pp. 341–348 (2009)
[35] OWLS-TC4 semantic web services collection (2011), http://www.semwebcentral.org/frs/download.php/488/OWLS-TC4_SWRL.zip
[36] Thirumaran, M., Dhavachelvan, P., Abarna, S., Aranganayagi, G.: Architecture for Evaluating Web Service QoS Parameters using Agents. International Journal of Computer Applications 10(4), 15–21 (2010)
[37] Tran, V.X., Tsuji, H.: A Survey and Analysis on Semantics in QoS for Web Services. In: Proceeding of International Conference on Advanced Information Networking and Applications, pp. 379–385 (2009)
[38] Kritikos, K., Plexousakis, D.: Requirements for QoS-based Web Service Description and Discovery. In: Proceeding of International Computer Software and Applications Conference, pp. 467–472 (2007)
[39] Java Language (2010), http://java.sun.com
[40] Gosling, J., McGilton, H.: The Java Language Environment – A Whitepaper. Technical Report, Sum Microsystems (October 1995)
[41] Gosling, J.: Java: An Overview (2010), http://labs.oracle.com/features/tenyears/volcd/papers/7Gosling.pdf (February 1999)
[42] Netbeans IDE (2010), http://www.netbeans.org/index.html
[43] Apache Web Server (2011), https://www.apache.org/
[44] OWL-S API (2010), http://www.mindswap.org/2004/owl-s/api/index.shtml
[45] Salehi, L., Mirhadi, P.: E-health applications implementation considerations. In: Proceedings of the 4th Kuala Lumpur International Confrence on Biomedical Engineering, vol. 21, pp. 858–861 (2008)
[46] Istepanian, R.S.H., Philip, N.Y., Martini, M.G.: Medical QoS provision based on reinforcement learning in ultrasound streaming over 3.5G wireless systems. Journal of IEEE J. Sel. A. Commun. 27(4), 566–574 (2009)

A Novel and Distributed Method of Distance Measurement and Localization for MWSN Based on AOA

Anil Bhawarkar, P.S. Patheja, and Akhilesh A.Waoo

Department of Computer Science, BIST Bhopal MP India
{anibhawarkar09,pspatheja}@gmail.com,
akhilesh_waoo@rediffmail.com

Abstract. The primary trouble in sensor networks is localization, such that shaping the position of the sensors. Node localization is an enabling technology for mobile wireless sensor networks (MWSNs) for the reason that sensor nodes deployed in an area of interest frequently require position information for routing and application - specific tasks, for instance - temperature and pressure monitoring. This paper present method of localization which is an angle of arrival based distributing localization scheme for sensor node and outlines recent work in which we have propose a efficient approach to localize a sensor node using angle of arrival method and RSSI weight is used to improve performance.

Keywords: Wireless sensor network, Localization, AOA, MWSN.

1 Introduction

Wireless sensor networking is a rising technology that promises a wide range of possible applications in both civilian and military areas. A wireless sensor network (WSN) usually consists of a huge number of low cost, low power, and multifunctional sensor nodes that are deployed in a region of interest. These sensor nodes are small in size but are equipped with sensors, embedded microprocessors, and radio transceivers. Consequently, they have not only sensing, but also data processing and communicating capabilities. They communicate over short distance via a wireless medium and collaborate to accomplish a common task [1]. Node localization is an enabling technology for wireless sensor networks (WSNs) because sensor nodes deployed in an area of interest usually need position information for routing and application - specific tasks, for example, temperature and pressure monitoring [2]. In many applications, a WSN is deployed to help improve localization accuracy in environments where the channel condition poses a challenge on range estimation [3]. In such environments, cooperative localization provides a potential for many applications in the commercial, public safety, and military sectors [3, 4].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 773–778.
springerlink.com

There are Challenges in mobile wireless sensor network are self organization of topology; configuration, energy, maintenance, and data process. A good localization algorithm ought to calculate a position as fast as possible and should be resistant to environmental influences with imprecise distances.

This paper is organized as follows : section 2 described basic concept of angle of arrival, section 3 deals with related work, proposed work is discussed in section4, section 5 shows result and analysis, section 6 consist conclusion and future work.

2 Localization through Angle of Arrival

Angle of arrival is defined as the angle between the propagation track of an incident signal and some reference direction, which is known as orientation. AOA are measured in degrees in a clockwise direction from the North. Whether the orientation is zero or pointing to the North, the AOA is absolute otherwise relative. Basic approach to find AOA measurements is to apply an antenna array on each sensor node.

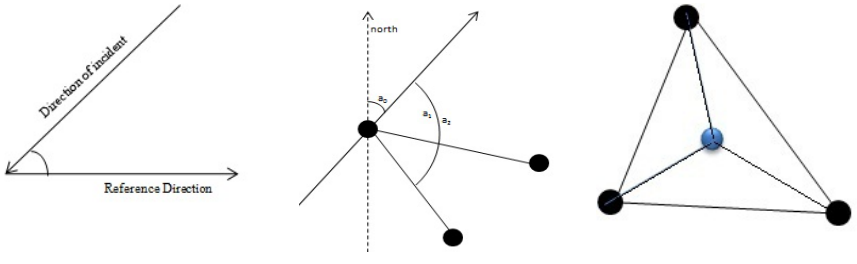


Fig. 1 (a) Demonstration of orientation, (b) AOA measurement when orientation is known (c) AOA measurement when orientation is unknown

The orientations of the unknowns may or may not be known at the time of deployment. Localizations under both scenarios can be solved using triangulation [5]. Existing models for the distribution of the AOA received at a wireless node include: Laplacian[10] , von Mises[11] and Gaussian AOA distribution[12]. These methods play important roles in the accuracy of the AOA measurements.

3 Related Work

3.1 Previous Work

Rong Peng and Mihail L. Sichitiu[5]: In this paper author focus on localization technique based on angle of arrival information between neighbor nodes they propose a new localization and orientation scheme that consider beacon

information multiple hopes away such that the communication between sensor nodes and the base station require multiple hopes.

Clement Saad and Adberrahim Benslimane and Jean-Claude Konig[6]: In this paper author concerned in new distributed technique for wireless sensor network they named their method AT-angle. This method is basically extended version of angle of arrival. This new method AT-angle facilitate two important properties first- a node detect if its estimated position is closed to its real position, second – Some wrong location information can be eliminated regarding to define sensor zone to minimize errors.

Patryk Mazurkiewicz and Kin K. Leung[7]: This paper author presented a distributed and scalable algorithm referred to as COBALT (Clique Of Node Based Localization Technique) produces a unique solution for any topology of connected nodes which solves the problem of localization for 3-D network of wireless sensor. They focus in this work is an enhancing the local map prior to stitching them to the global map. Enhancing the local map results in achieving much better accuracy COBALT has no rigidity conditions this property was achieved by using rich spatial measurement including : range, angle of arrival, and earth gravity direction.

Baoli Zhang and Fengqi Yu[8]: This paper tackle the problem in estimating localization of randomly deployed sensor nodes and increases equipped with directional antenna. This approach is energy efficient because sensor network uses the property of when if received the signal enough to calculate the location it stops receiving signal from beacons which made it energy efficient.

Fu-Kai Chan and Chih-Yu Wen[9]: This paper presented a network based positioning system, describe a distributed AOA aided TOA positioning algorithm and outline recent work in which author developed an efficient principal approach to localize a mobile sensor using time of arrival and angle of arrival information employing multiple seed in the line of sight scenario.

3.2 Paper Contribution

Study and implement angle of arrival method and improved model for angle of arrival with the help of the values weight of Received signal strength signal, and then calculate it with basic angle of arrival method result to improved result.

4 Proposed Work

4.1 Why Angle of Arrival

There are four types methods of range based localization technique – RSSI (Received Signal Strength Indicator), TOA (Time of arrival), TDOA(Time Difference Of Arrival) and AOA(Angle Of Arrival), The reason behind the AOA method has been chosen because the RSSI is the easiest method to obtained the location of a sensor node is utilized then other type of method measurement

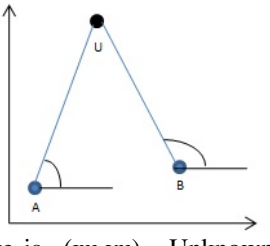
although required specialized hardware and can provide much better accuracy. So we focus on AOA and RSSI method is only used to improve method's efficiency.

4.2 Assumptions

We are extending the basic AOA method [6] and assumptions about sensor nodes. All Anchor node have ominidirectional antenna (do not allow to measure AOA) and other unknown sensor nodes are able to calculate angle form incoming signal from anchor node

4.3 Improved Angle of Arrival Localization Algorithm

Step 1 Node distribution in specific area: Sensor nodes are placed in ad-hoc fashion in a particular region.



Step 2 Anchor node location calculations: Initially some anchor nodes location is known it was hand placed or calculated by GPS system[14]. A and B are anchor nodes and its coordinates are (xa,ya) and (xb,yb), unknown node is U and its unknown coordinate is (xu,yu). Unknown node U does not its location, A and B sends their coordinate information to U.

Step 3 Location calculations through AOA: Unknown node U does not its location, A and B sends their coordinate information to U.

Angle A=$\emptyset_a$ and B= $\emptyset_b$

i) Select unknown sensor node U(x_u,y_u)
ii) Select Anchor node A(x_a,y_a) and B(x_b,y_b)
iii) Calculate Angle of arrival and location of unknown node tan $\emptyset_i$= (y_u-y_i)/ (x_u-x_i)

Step 4 Compute weighted factor anchor nodes:

i) Anchor nodes RSSI are $RSSI_a$ and $RSSI_b$.
ii) Calculate weight (W_{ab}) of both anchor nodes in behalf of current selected unknown node

Step 5 Improved AOA = AOA + RSSI weight

Improved AOA = (x_u,y_u)+W_{ab}

Step 6 Distance measurement: Basic distance measurement method is applied here Distance between U (x_u,y_u) and A (x_a, y_u)

xd = x_u- x_a
yd = y_u – y_a
Distance (D_{ua}) = SquareRoot(xd*xd + yd*yd)

Distance between U (x_u,y_u) and B(x_b,y_b) xd = x_u- x_b
yd = y_u – y_b
Distance (D_{ub}) = SquareRoot(xd*xd + yd*yd)

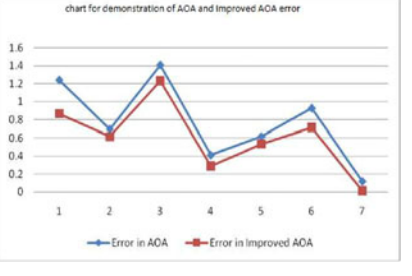
5 Result and Analysis

In this paper we have use network simulator 3, our simulation result shows that this algorithm shows better result than basic Angle of arrival localization algorithm. In the analysis of this algorithm we found that only angle of arrival shows better result but we can use weight of RSSI to get better result. When the localization is efficiently done then we can measure the distance of unknown sensor from anchor node very accurately.

We prefer the wireless standard 802.11 used Random way point mobility model and node initially placed in random location in simulation area, and moved in randomly chosen direction in constant speed 10m/s. **Mobility model random way point**: Minimum speed 0mps, Maximum speed 10 mps, Pause speed 20 sec, Simulation Time 120s.

Table 1. Efficiency table

Node Id	Error in AOA	Error in Improved AOA
1	1.242	0.867
2	0.698	0.612
3	1.410	1.231
4	0.412	0.289
5	0.618	0.532
6	0.930	0.712
7	0.120	0.012



6 Conclusion and Future Work

We presented a distributed angle of arrival based localization of mobile wireless sensor network with some assumption are – every unknown sensor node have the capability to detect the angle of incident signal from neighboring sensor node all sensor nodes are capable to calculate the weight of RSSI in time interval. We have improved the accuracy of AOA localization.

For future enhancement we suggest to improve the performance of this algorithm in terms of cost efficient, energy efficient and security.

Acknowledgement. I express my sincere gratitude and acknowledgement towards P S Patheja sir and Akhilesh A Waoo sir, who guided me. It was their constant support and inspiration without which my effort would not have taken this shape. I sincerely thank them for this and seek them support for all my future endeavors.

References

[1] Akyildz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)

[2] Patwari, N., Ash, J.N., Kyperountas, S., Hero, A.O., Moses, R.L., Correal, N.S.: Locating the nodes: Cooperative localization in wireless sensor networks. IEEE Signal Processing Magazine 22(4) (July 2005)

[3] Pahlavan, K., Akgul, F.O., Heidari, M., Hatami, A., Elwell, J.M., Tingley, R.D.: Indoor geolocation in the absence of the direct path. IEEE Wireless Communications Magazine 13(6), 50–58 (2006)

[4] Peng, R., Sichitiu, M.L.: Angle of Arrival Localization for Wireless Sensor Networks. In: Proc. of the Third Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Reston, VA (September 2006)

[5] Saad, C., Benslimane, A., Konig, J.-C.: AT-Angle: A Distributed Method for Localization using Angles in Sensor Networks. In: IEEE Computers and Communications, ISCC (2008)

[6] Mazurkiewicz, P., Leung, K.K.: Performance of Angle-and-Range-Based Localization of Wireless Sensors. IEEE Communications, Computers and Signal Processing (2009)

[7] Zhang, B., Yu, F., Chan, F.-K., Wen, C.-Y.: Adaptive AOA/TOA Localization Using Fuzzy Particle Filter for Mobile WSNs. In: IEEE Vehicular Technology Conference, VTC Spring, IEEE (2011)

[8] Spencer, Q.H., Jeffs, B.D., Jensen, M.A., Swindlehurst, A.L.: Modeling the statistical time and angle of arrival haracteristics of an multipat channel. IEEE Journal Selected Areas in Communication 18(3) (March 2000)

[9] Abdi, A., Barger, J., Kaveh, M.: A parametric model for the distribution of the angle of Barger, and Kaveh, arrival and the associated correlation function and power mobile. IEEE Transactions on Vehicular Tec Spectrum at the Station 51(3) (May 2002)

[10] Pedersen, K.I., Mogensen, P.E., Fleury, B.H.: A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments. IEEE Transactions on Vehicular Technology 49(2) (March 2000)

[11] Zhu, Y., Zheang, B., Yu, F., Ning, S.: A RSSI Localization Algorithm Using a Mobile An chor Node for Wireless Sensor Network, pp. 123–126. IEEE (2009)

[12] Parkinson, B., et al.: Global Positioning system: Theory and application. In: Progress in Astronomic and Aeronautics, vol. I (1996)

Artificial Weed Colonies with Neighbourhood Crowding Scheme for Multimodal Optimization

Ratul Majumdar[1], Ankur Ghosh[1], Aveek Kumar Das[1], Souvik Raha[1], Koushik Laha[1], Swagatam Das[1], and Ajith Abraham[2]

[1] Department of E.T.C.E, Jadavpur University, Kolkata, India
[2] VSB - Technical University of Ostrava, Czech Republic
{majumdar.ratul,ankurg708,aveek23,rahasouvik37, lahakoushik1}@gmail.com, swagatamdas19@yahoo.co.in, ajith.abraham@ieee.com

Abstract. Multimodal optimization is used to find multiple global & local optima which is very useful in many real world optimization problems. But often evolutionary algorithms fail to locate multiple optima as required by the system. Also they fail to store those optima by themselves. So we have to use other selection scheme that can detect & store multiple optima along with evolutionary algorithms. Hence we use niching which is a very powerful tool in detecting & storing multiple optima. Niching methods were introduced to EAs to allow maintenance of a population of diverse individuals so that multiple optima within a single population can be located .Crowding which is a very primitive branch of niching is used here as the selection scheme with Invasive Weed Optimization (IWO) which is a ecologically inspired algorithms depicting behaviors of plants . For multimodal optimization the total search space is divided into several niches in which separately IWO is applied to find the optima in niches. The niches will also store this optima within themselves.

Keywords: Invasive Weed Optimization, Niching, Neighboring Crowding.

1 Introduction

In practical optimization problems, it is often desirable to simultaneously locate multiple global and local optima of a given objective function. A multimodal optimization task amounts to finding multiple optimal solutions and not just one single optimum, as it is done in a typical optimization study. If a point-by-point classical optimization approach is used for this task, the approach must have to be applied several times, every time hoping to find a different optimal solution.

Evolutionary Algorithms (EAs) [1, 2], due to their population-based approach, provide a natural advantage over classical optimization techniques. They maintain a population of possible solutions, which are processed every generation, and if

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 779–787.
springerlink.com

the multiple solutions can be preserved over all these generations, then at termination of the algorithm we will have multiple good solutions, rather than only the best solution. *Niching* [3 – 5] is a generic term referred to as the technique of finding and preserving multiple stable *niches*, or favorable parts of the solution space possibly around multiple solutions, so as to prevent convergence to a single solution. the most popular niching techniques used in conjunction with the evolutionary computation community include crowding [7], fitness sharing [6], restricted tournament selection [8], and speciation [9]. Most of existing niching methods, however, have difficulties which need to be overcome before they can be applied successfully to real-world multimodal problems. Some identified issues include: difficulties to pre-specify some niching parameters; difficulties in maintaining discovered solutions in a run; extra computational overhead, and poor scalability when dimensionality is high.

In this paper we propose a simple yet very powerful hybrid EA that synergistically combines the features of two EAs: Invasive Weed Optimization (IWO) [10] and Neighbourhood Crowding technique[19] for multimodal optimization. The reason for employing the IWO in multimodal optimization process is its high explorative power [10] over the bound-constrained search space [11]. Each weed initialized in the search space, produces seeds around it, in a controlled hyper-space thus creating virtual subpopulation of seeds. Hence after a sufficient number of generations the colony is basically spread in sub-regions surrounding promising local and global optima. The crowding scheme is applied along with IWO as described in the proposed algorithm in order to preserve the niches throughout the computation and the maximum number of global peaks can be obtained. [12-17]

2 Evolutionary Multi-modal Optimization Using Niching

If a single-objective optimization problem has more than one optima, it can be considered as multimodal optimization problem. The objective of locating & storing different optima in a single run makes it more complicated than single global optimization. *Niching methods*, formed for extending EAs to multimodal optimization, address this problem by maintaining the diversity of certain properties within the population - and this way they allow parallel convergence into multiple optimal solutions in multimodal domains. The concept of niching is inspired by the way organisms evolve in nature. As Mahoud described [3], *"A niching method must be able to form and maintain multiple, diverse, final solutions, whether these solutions are of identical fitness or of varying fitness. A niching method must be able to maintain these solutions for an exponential to infinite time period, with respect to population size."* The process involves the formation of subpopulations within a population. Each subpopulation aims to locate one optimal solution and together the whole population is expected to locate multiple peaks in a single run. Several niching methods were proposed in literature e.g. crowding, deterministic crowding, fitness sharing], derating, restricted tournament selection, parallelization, clustering, clearing and speciation .We shall briefly review only crowding technique below.

Crowding and restricted tournament selection: In 1975, De Jong introduced the classical crowding method [26] known as De Jong's crowding that tries to maintain population diversity by allowing competition for limited resources among similar individuals in the population. Hence, effectively the competition takes place within each niche. Generally the similarity is measured using Euclidean distance between individuals. The algorithm compares an offspring with some randomly selected individuals from the current population. The most similar individual will be replaced if the offspring is a better solution. A parameter *CF* called crowding factor is used to control the size of the sample. *CF* is generally set to 2 or 3. Because of this low *CF* values, replacement errors is one of the main problem for crowding. Mahoud tried to improve the original crowding by proposing a scheme of deterministic crowing. It eliminates the *CF,* reduces the replacement errors, and restores selection pressure.

In very similar spirit to crowding, the restricted tournament selection [8] method selects a random sample of *w* (window size) individuals from the population and determines which one is the nearest to the offspring, by either Euclidean (for real variables) or Hamming (for binary variables) distance measure. The nearest member within the *w* individuals will compete with the offspring and the one with higher fitness will survive in the next generation.

Apart from the above, several other niching methods have also been developed over the years, including fitness sharing clearing, speciation, derating, parallelization, and clustering. To the best of our knowledge, IWO has not been applied to solve the multimodal optimization problems till date.

3 IWO and Its Proposed Modification

IWO is a population-based algorithm based on trial and error method that copies the colonizing behavior of weeds. Weed grows its population entirely or predominantly in a geographically specified area which can be substantially large or small without any control of any external factor in a very random manner. Initially a certain number of weeds are randomly spread over the entire search range. These weeds will eventually grow up and execute the following steps and the algorithm proceeds.

3.1 Classical Invasive Weed Optimization

There are four steps in the classical IWO algorithm as described below:

Initialization: A finite number of weeds are initialized randomly in the search space.

Reproduction: Each member of the population is allowed to produce seeds depending on its own, as well as the colony's lowest and highest fitness, such that, the number of seeds produced by a weed increases linearly from lowest possible seed for a weed with the worst fitness to the maximum number of seeds for a weed with the best fitness.

Spatial Distribution: The generated seeds are randomly scattered over the *d*-dimensional search space by perturbing them with normally distributed random numbers with zero mean and a variable variance. This step ensures that the produced seeds will be generated around the parent weed, leading to local search around each plant. However, the standard deviation (*sd*) of the random function is made to decrease over iterations. If $sd_{\max}$ and $sd_{\min}$ are the maximum and minimum standard deviations and if pow is a real number, then the standard deviation for a particular iteration can be given as in eqn (1): (iter is current iteration)

$$sd_{ITER} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^{pow} \left(sd_{\max} - sd_{\min} \right) + sd_{\min}, \qquad (1)$$

Competitive Exclusion: Some kind of competition between plants is needed for limiting maximum number of plants in a colony. Initially, the plants in a colony will reproduce fast and all the produced plants will be included in the existing colony, until the number of plants in the colony reaches a maximum value $pop_{\max}$. However, it is expected that by this time the fitter plants have reproduced more when compared to weaker plants. From then on, only the fittest plants up to pop_max, among the existing ones and the reproduced ones, are taken in the colony and steps 2 to 4 are repeated until the maximum number of iterations has been reached, i.e. the colony size is fixed from thereon to $pop_{\max}$. This method is known as competitive exclusion and is also a selection procedure of IWO.

3.2 The Proposed Crowding IWO Algorithm

The crowding IWO algorithm is based on the application of Invasive Weed Optimization (IWO) with the selection scheme used for multimodal optimization is a niching method crowding. At first we initialize some plants or particles within the search space randomly. Then their individual fitness are calculated as per the function used . Now depending on the fitness of the individuals & the best & worst fitness of the colony each plant produces certain number of seeds which linearly decreases from best to worst fitness. Now these seeds are also spread randomly in the search space with a particular standard deviation, functions that decreases with iteration & hence plants with better fitness produces large number of seeds close to it which becomes a potential solution for the test function.

Now the crowding scheme is used here as the selection scheme where the entire population is divided into several subpopulation . That's how the niches are formed. The number of subpopulation is a niching parameter that has to be given by the user. The plants within a particular niche produces seeds & the seed most identical to its parent plant i.e. which has a minimum Euclidean distance to its parent plant is selected .If the seed has a better fitness than its parent then the seed replaces its parent plant. Now in this way total population is updated & if it

exceeds the maximum number of member of the population then the plants with better fitness are kept & others are removed from the search space. In this way the niches turn out to be potential solutions for multimodal optimization.

3.3 Pseudo Code

Initialize a population within search area randomly
Find the fitness value for the given function
While(iter<maxit)
$\sigma iter$ = ((maxit-iter)^mi/(maxit^mi)*(σ ini- σ fi)+ σ fi;
Find the seeds of the plants
Seed(1)=(fit(1)-maxfit)*(max no. _seeds-min no._seeds)/(max_fit-min_fit);
Convert the seeds into plants;
Update the total no. of plants;
For j=1 to plants
For i=1:seeds_of_plants (j)
Pos_seed (i)=pos_plant(j)+ σ iter*randl;
End
Update position of plants
End
If pos_plant<LB
Pos_plant=LB;
End
If pos_plant>UB
Pos_plant=UB;
End
Find the Euclidean distance between parent plant &its off-springs
Total population is distributed into sp number of subpopulation;
Size_subpop=total_pop/sp;
Within each subpop if the seed with min distance with its parent has a better fitness
Replace the parent with that seed
Update total population
If(no._plants>max_plant)
Eliminate the plants with poor fitness & keep the no_plants maxplant with better fitness
Iter=iter+1;
End while

---

4 Experiments and Results

The computing was achieved on a Pentium 4 computer running at 3.0 GHz equipped with 1.5 GB of RAM. Results are presented for benchmark functions

given in the tables below where both composite & simple test functions. The results of this algorithm is also represented in a tabular form giving a comparison between the other standard algorithms.

Parametric Setup:

The following parameters were used while performing the experiments:
Maximum number of iterations (maxit):1000
Initial value of standard deviation (σ ini):3
Final Value of standard deviation (σ fi)=.0001
Maximum number of seeds from each plant (Max no. _ seeds):5
Minimum number of seeds from each plant (Min no. _ seeds):0
Number of subgroups formed within the population: sp
Maximum number of weeds (Max_plant):150
The composite functions in this experiment are the composite functions in [12]

Table 1. Benchmark Functions

Name	Dim	Test Function	Range	Peaks
Two-Peak Trap	1	$f_1(x)=\begin{cases}\frac{160}{15}(15-x) & for\ 0\le x<15\\ \frac{200}{5}(x-15) & for\ 15\le x\le 20\end{cases}$	$0\le x\le 20$	1
Central Two Peak Trap	1	$f_2(x)=\begin{cases}\frac{160}{10}x & for\ 0\le x<10\\ \frac{160}{5}(15-x) & for\ 10\le x<15\\ \frac{200}{5}(x-15) & for\ 15\le x\le 20\end{cases}$	$0\le x\le 20$	1
Equal Maxima	1	$f_3(x)=\sin^6(5\pi x)$	$0\le x\le 1$	5
Decreasing Maxima	1	$f_4(x)=\exp\left[-2\log(2)\cdot(\frac{x-0.1}{0.8})^2\right]\cdot\sin^6(5\pi x)$	$0\le x\le 1$	1
Uneven Maxima	1	$f_5(x)=\sin^6(5\pi(x^{\frac{3}{4}}-0.05))$	$0\le x\le 1$	5
Uneven Decreasing Maxima	1	$f_6(x)=\exp\left[-2\log(2)\cdot(\frac{x-0.08}{0.854})^2\right]\cdot\sin^6(5\pi(x^{\frac{3}{4}}-0.05))$	$0\le x\le 1$	1
Six Hump Camel Back	2	$f_7(\vec{X})=-4\left[\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+\left(-4+4x_2^2\right)x_2^2\right]$	$-1.9\le x_1\le 1.9$ $-1.1\le x_2\le 1.1$	2
Himmelblau's Function	2	$f_8(\vec{X})=200-(x_1^2+x_2-11)^2-(x_1+x_2^2-7)^2$	$-4\le x_1,x_2\le 4$	4

The *peak accuracy* measure [20] is calculated as follows: for each global optimum to be found, the closest individual x in the population is taken and the absolute difference in fitness values is computed after maximum number of function evaluations has been elapsed. Then, all these differences are summed and divided by the number of global optima to be found. The peak accuracy calculation is shown below: (Table 3 shows a comparison of peak accuracy values of test functions and the 1st 7 composite functions)

$$peak\ accuracy = \sum_{i=1}^{\#\,glo\ peaks} \frac{\left| f(peak_i) - f(\vec{X}) \right|}{\#\,glo\ peaks} \tag{2}$$

Table 2. Average Number of Peaks found for Test Functions and Comparisons with other Algorithms

Function	ε	r	sp	CIWO	CMA-ES	CDE	SDE	FER PSO	SPSO
f_1	0.05	0.5	150	**1**	**1**	**1**	**1**	0.72	0.48
f_2	0.05	0.5	150	**1**	**1**	**1**	**1**	**1**	0.44
f_3	0.000001	0.01	175	**5**	4.92	3.84	4.72	4.84	4.88
f_4	0.000001	0.01	150	**1**	1	0.72	1	1	1
f_5	0.000001	0.01	175	**5**	4.88	3.96	4.6	5	4.92
f_6	0.000001	0.01	150	**1**	1	0.6	1	1	1
f_7	0.000001	0.5	200	1.6	1.6	0.04	2	**1.96**	0.08
f_8	0.005	0.5	160	**3.88**	3.72	0.32	3.72	3.68	0.84
CF_1 of [12]	0.5	1	200	**1.3**	1.08	0	1.8	1.08	0
CF_2 of [12]	0.5	1	300	1	1.52	1.2	1.2	2	0
CF_4 of [12]	0.5	1	375	**3**	0	0	0	0	0
CF_5 of [12]	0.5	1	200	**2**	1.12	1.12	1.32	2	0
CF_7 of [12]	0.5	1	200	1	0	0	**1.8**	1.52	0
CF_8 of [12]	0.5	1	250	**2**	0	0	1.4	1.5	0
CF_{10} of [12]	0.5	1	300	**2.5**	0	0	1.1	1.1	0
CF_{13} of [12]	0.5	1	350	**1**	0	0	0.9	0.3	0
CF_{15} of [12]	0.5	1	375	**2**	0	0	1.6	1.2	0

Table 3. Peak accuracy of simple test functions and first seven composite functions

Function	r	CIWO	CMA-ES	CDE	SDE	FERPSO	SPSO
f_1	0.5	**5.32 e-15**	9.63e-09	9.46e-08	1.23e-08	5.24e-02	8.74e-02
f_2	0.5	**7.23 e-11**	1.87e-07	8.76e-06	3.43e-07	9.65e-04	9.45e-02
f_3	0.01	**8.65e-11**	9.56e-08	7.43e-05	9.53e-07	5.65e-07	3.12e-07
f_4	0.01	**1.65 e-12**	8.54e-09	9.43e-06	4.03e-09	8.34e-09	2.16e-09
f_5	0.01	**3.92 e-10**	4.36e-07	5.39e-05	8.27e-07	5.45e-09	9.58e-08
f_6	0.01	**6.47 e-09**	2.89e-07	8.97e-05	4.51e-07	7.41e-07	2.98e-07
f_7	0.5	1.26 e-06	2.63e-05	3.42e-04	**5.33e-08**	7.38e-08	3.58e-04
f_8	0.5	**2.72 e-05**	6.52e-04	4.27e-02	8.57e-04	8.69e-04	5.21e-02
$CF_1$of [12]	1	32.45	35.48	75.68	35.48	**30.96**	1.05e+02
CF_2 of [12]	1	26.58	32.16	38.92	32.53	**25.54**	1.10e+02
CF_4 of [12]	1	**1.2e+02**	1.89e+02	1.60e+02	1.39e+02	1.23e+02	2.49e+02
CF_5 of [12]	1	30.71	42.56	45.62	**30.55**	39.52	1.07e+02
CF_7 of [12]	1	27.64	59.87	17.32	**25.66**	32.62	1.47e+02

5 Conclusions

In this paper we proposed a multimodal evolutionary optimization technique that summarizes concepts powerful modern optimizing technique IWO and a selection technique of niching called crowding. The crowding IWO algorithm was tested for the optimization of fifteen benchmark functions (including five composite functions). To justify its development, results were directly compared with nine state-of-the-art evolutionary multi-modal optimizers based on the performance metrics like average number of peaks found and peak accuracy. The results of our experimental studies suggest that crowding IWO can provide a statistically superior and more consistent performance than the other standard multimodal optimization algorithms.

References

1. Bäck, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Oxford Univ. Press (1997)
2. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
3. Mahfoud, S.W.: Niching methods for genetic algorithms, Technical Report (TR No.: 95001), Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign (1995)
4. Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. IEEE Transactions in Evolutionary Computation 2(3), 97–106 (1998)
5. Singh, G., Deb, K.: Comparison of multimodal optimization algorithms based on evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1305–1312. ACM Press, Seattle (2006)

6. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41–49 (1987)
7. Thomsen, R.: Multimodal optimization using Crowding-based differential evolution. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, pp. 1382–1389 (2004)
8. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Proceedings of the 6th International Conference on Genetic Algorithms, San Francisco, pp. 24–31 (1995)
9. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proc. of 3rd IEEE Congress on Evolutionary Computation, pp. 798–803 (1996)
10. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1, 355–366 (2006)
11. Chakraborty, P., Roy, G.G., Das, S., Panigrahi, B.K.: On population variance and explorative power of the invasive weed optimization algorithm. In: Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), December 9-11, pp. 227–232. IEEE Press, Coimbatore (2009)
12. Qu, B.Y., Suganthan, P.N.: Novel multimodal problems and differential evolution with ensemble of restricted tournament selection. In: IEEE Congress on Evolutionary Computation, Barcelona, Spain (July 2010)
13. Shir, O.M., Emmerich, M., Bäck, T.: Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. Evol. Comput. 18(1), 97–126 (2010)
14. Li, X.: Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Transaction on Evolutionary Computation 14(1) (February 2010)
15. Li, X.: Efficient differential evolution using speciation for multimodal function optimization. In: Proceedings of the Conference on Genetic and Evolutionary Computation, Washington DC, USA, pp. 873–880 (2005)
16. Li, X.: Multimodal function optimization based on fitness-euclidean distance ratio. In: Proc. Genet. Evol. Comput. Conf., pp. 78–85 (2007)
17. Gan, J., Warwick, K.: A variable radius niching technique for speciation in genetic algorithms. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2000), San Francisco, USA, pp. 96–103 (2000)
18. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Doctoral Dissertation, University of Michigan (1975)
19. Li, X.: Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Transaction on Evolutionary Computation 14(1) (February 2010)
20. Thomsen, R.: Multimodal optimization using Crowding-based differential evolution. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, pp. 1382–1389 (2004)

Virtual Learning System: A Conceptual Framework of Network Optimization

R. Soundhara Raja Pandian[1], S. Thangalakshmi[2], and S. Saravanan[3]

[1] Webstudio, IC&SR, Indian Institute of Technology Madras, India
[2] Department of Electrical and Electronics Engineering, GKM College of Engineering and Technology, Chennai, India
[3] Department of Civil Engineering, National Institute of Techology, Trichy, India
{rsrp.iitm,thangalakshmiprakash,ssaravananirs}@gmail.com

Abstract. The demand for network communications is increasing every year, but the available resources are not increasing at the same rate. Bandwidth and network infrastructures are major issues in network related problems in remote areas for various applications. The current study uses Genetic Algorithm (GA) based optimization method to allocate the bandwidth related issues. In addition, it also considers the identification of optimal server location with minimal cost. In this regard, Geographical Information System (GIS) is used to identify the spatial location of network components (server setup) optimal location. The proposed methodology is demonstrated with a case study of National Programme on Technology Enhanced Learning (NPTEL) developed by Ministry of Human Resources Department (MHRD), Government of India, where the network optimization is of much importance.

Keywords: Bandwidth, Infrastructure, Genetic Algorithm, Geographical Information System, Load balancer.

1 Introduction

In recent years, there is a constructive growth in communication field over the world with dedicated network systems. Application of advanced networking systems opened up many possible implementations across various domains such as military, mobile communication, agriculture, education and medicine which are a few worth mentioning here. This study focuses on education field that uses network applications effectively. Education is a key factor that improves countries economy through scientific development. Therefore, the educational system should be re-framed from the current traditional class room teaching into sophisticated computer based studies in order to transform the knowledge effectively. This is quite possible with advanced development in communication and multimedia technology. There are attempts that have been made by few on-line educational

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 789–795.
springerlink.com

systems (i.e. NPTEL, Massachusetts Institute of Technology's Open Course ware (MITOCW), Common wealth of learning, British Open University and Australian Open University are entities that provide similar services). These systems contain online teaching materials (e.g., web and video format courses) in favor of making fuller use of the country's top academic talent faculties in cost-effective ways [8]. However, the major drawback associated with these systems lacks in interaction between students and faculties and separated from traditional classroom teaching methodology. In addition, proper online enrollment, conducting examination are the major gaps that exist in providing the education.

These problems could be solved through Virtual University (VU) concepts to some extent. The VU offers educational programs with communication and media technologies. The goal of virtual universities is framed to provide access to the student community who would not be able to get a chance in appearing for the physical campus education. The VU concept can aid in improving the quality of education with the help of reputed universities where there is limited number of direct admissions exist. In India, Indian Institutes of Technology (IITs) and IISC are the premier educational institutes offering quality-engineering studies. Hence, the VU concept is an alternative method of teaching where the students do not have exposure into IIT system. The NPTEL is an open access online educational system developed by IITs and IISc, which has received lot of attention among engineering students and industrialists. Therefore, these online courses can be directly included in VU without any additional efforts in newly creating online courses. However, the availability of bandwidth and network components is not able to effectively cater to the needs with current NPTEL system and its networking devices. Therefore network optimization is a necessary criterion that should be enhanced to improve the overall service [7]. Numerous researchers explored the possibility of applying the optimization algorithm in network bandwidth related issues (i.e. fixed point optimization algorithm [3], dual decomposition [5], game theory [2] and backbone topology [1])

In this study, the major focus has been attributed towards solving the network related issues with optimization algorithm to distribute the network slots based on the available bandwidth. It also considers the application of GIS in order to route the signal spatially where the server location is nearer.

2 Methodology

The main objective of the study focuses on developing an optimized solution to install server and network devices at the optimal location as well as allocation of slots based on the number of users and the distance between user and servers. Although this is associated with critical issues in design of communication channels and a network system, the dedicated networks are the key components in providing the services in a better way. As of now, Internet is one of main sources in providing access to the users. It is well known that the bandwidth of Internet connection is low. Still, there are options to provide this access via dedicated network lines such as NKN and VPN without bandwidth difficulties.

The National Knowledge Network (NKN) is an e-infrastructure, which has high-speed network at 10 Gbps bandwidth. It is used to access the very large databases in a cost effective manner in order to connect more than 5000 nodes across the country. A virtual private network (VPN) is a computer network that uses a public telecommunication infrastructure to provide remote offices or individual users with secure access to their organization's network. It encapsulates data transfers using cryptographic techniques between two or more network devices connected by a public network to ensure the security of information and data that is transferred.

Cloud computing is a recently developed concept to connect remote server and users in order to access high speed computing and sophisticated software. Pandian and Kasiviswanathan [6] studies show that the possible implementation of Cloud Software as a Service (SaaS) to connect the local user (i.e. engineering students) with the high-speed server to make use of preferred application.

3 Case Study

In this study, it was found that the current NPTEL system could connect 1500 users effectively at the given specified time. Figure 1 shows the number of users visited NPTEL website from 2008 to 2011 and it is very clear that the curve moves in a linearly increasing trend and it is confirmed with the study [4]. Therefore, it is a prior requirement that the bandwidth and network issues should be designed by considering the future increasing demand. The NKN covers spectrum of application ranges from education and research, agriculture, climate change, military, heath care and industry.

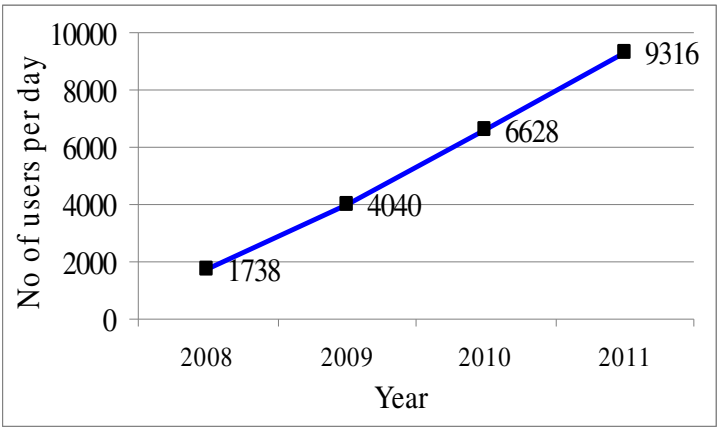


Fig. 1. Visitors history of NPTEL website (Source: Google Analytics)

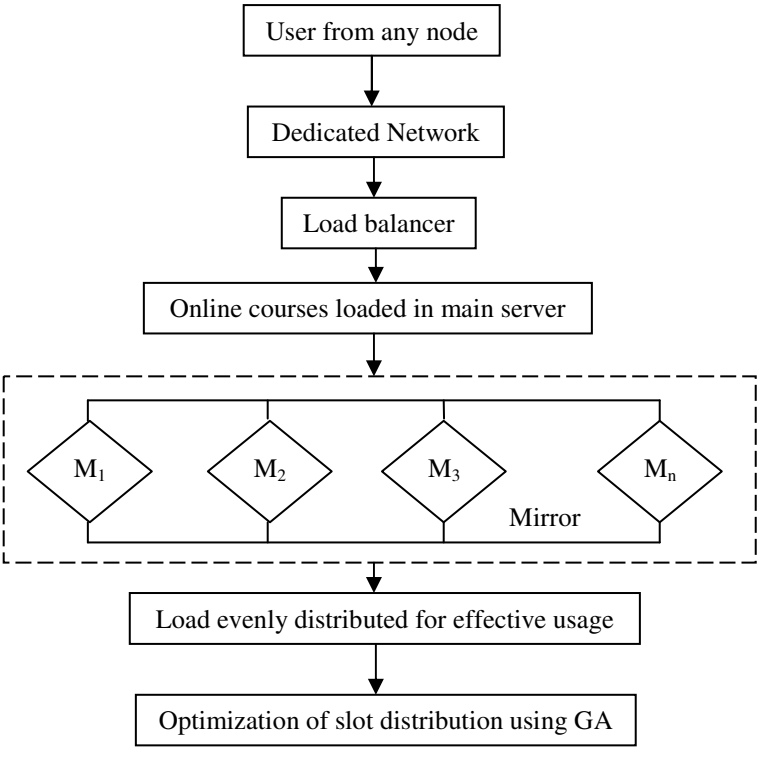


Fig. 2. Proposed methodology

The NPTEL is one among the applications proposed using this network. This involves various steps to be followed for effective implementation. Therefore the current study would be an attempt in addressing the various levels of complex issues that are involved in this process and possible solutions can be met. The proposed methodology is given in the flowchart (Fig. 2).

The optimal location of server is an important step and this could be solved using GIS based educational topography that contains the nodes of colleges and university location and number of visitors who use NPTEL website.

According to the flowchart given in fig.2, the user from any node of India will initially contact main server which includes load balancer and other networking devices such as router and switches that calculates the load based on bandwidth availability and distance from the user to nearest mirror. In this stage, the optimization helps in selecting the mirror, which has the slot as well as location-wise near. The GA based optimization algorithm routes the user in different servers based on nearer server location and the available bandwidth.

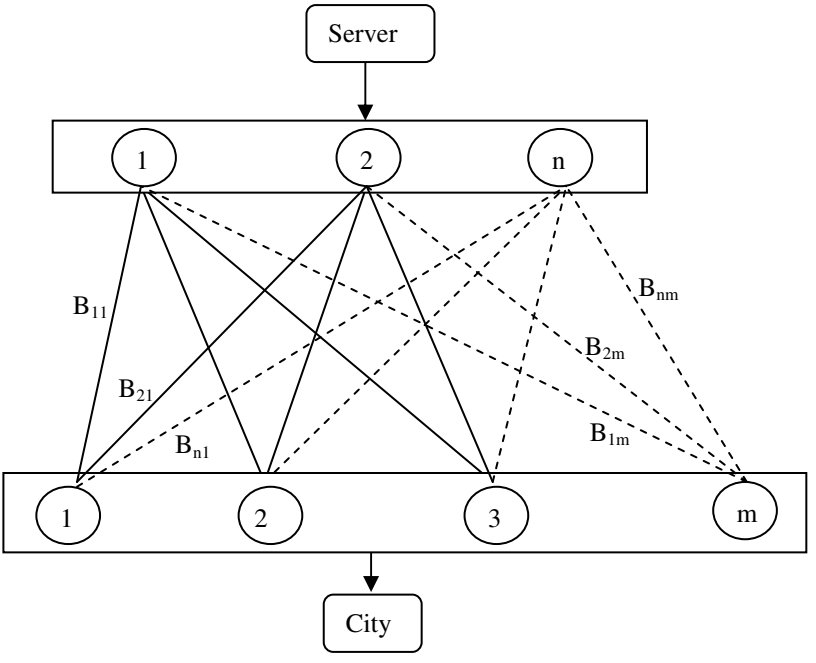


Fig. 3. Network connecting server and users

The available network connection that is being adopted so far connects the server and user based on Internet Protocol (IP) using GIS. The major limitation pertains to this type of connection is that connects the server locally and fails to connect the server which is located other than local server. Therefore, the present study is focused on connecting the user and server even if the server is located far from the user. The primary objective is to distribute the available bandwidth based on distance. In this way, connecting the next nearest server ensures the optimum bandwidth allocation. Figure 3 shows the typical servers that are connected with different cities. Each city has a specified number of users subsequently the bandwidth used by different users will vary based on the application (i.e. web course or video course). The following equations describe the objective functions and constraints of the proposed method.

Objective function:

$$Minimize : Z = \sum_{1}^{m} \sum_{1}^{n} B_{nm} * D_{nm} \tag{1}$$

Constraints:

City m

$$B_{1m} + B_{2m} + ...B_{nm} \geq (\sum_{j=1}^{j=k} b_k)_m \tag{2}$$

Server n

$$B_{n1} + B_{n2} + ...B_{nm} \leq S_n \tag{3}$$

Where, B is the total bandwidth in Gbps, S indicates the total server capacity, b refers to the bandwidth that is used by single user and the total number of user is k. The n, m are total number of servers and cities respectively. It is to be noted that all the variables in the equations are non-negative quantities.

The objective function is formulated in such a way that based on the distance (D in Km/Gbps) the bandwidth is allocated between the server and user with relevant constraints as mentioned earlier. The distance is calculated using GIS and it is coupled with optimization algorithm such as GA. Here it is worth mentioning that the proposed methodology is associated with some of the challenges, which need to be taken care (i.e. Cyber security, Data access and ownership, Encryption needs, High performance servers and effective networking devices and Very high implementation cost).

4 Conclusion

Virtual University (VU) concepts can certainly help in getting good quality of education where the students are not having the physical access. This requires dedicated network devices and servers to access services that are in general costlier. Therefore, network optimization is a major concern. Network optimization reduces the cost involved in hardware selection and its maintenance. This study aimed in addressing the optimization involved spatially where the server to be located and temporally where slot distribution to be allocated based on the available bandwidth. In this regard, GIS has been suggested in finding the optimal location of mirror based servers, and GA based optimization algorithm could be the better choice for optimal allocation of bandwidth as these techniques are powerful tools and applied in various fields due to their promising results.

Acknowledgment. We would like to thank Prof. Mangala Sunder Krishnan, Professor and NPTEL Coordinator, IIT Madras for his support and encouragement as well as three anonymous reviewers for the excellent comments to improve the quality of the manuscript. We would also like to extend our thanks to Mr. K.S. Kasiviswanathan, IIT Madras for his support and technical inputs.

References

[1] Gerla, M., Suruagy, M.J.A., Pazos, R.: Topology design and bandwidth allocation in ATM nets. IEEE Journal on Selected Areas in Communications 7(8), 1253–1262 (1989), doi:10.1109/49.35570

[2] Haïkel, Y., Mazumdar, R.R., Catherine, R.: A game theoretic framework for bandwidth allocation and pricing in broadband networks. IEEE/ACM Transactions on Networking (TQN) 8(5) (2000)

[3] Iiduka, H., Uchida, M.: Fixed Point Optimization Algorithms for Network Bandwidth Allocation Problems with Compoundable Constraints. IEEE Communications Letters 15(6), 596–598 (2011), doi:10.1109/LCOMM.2011.040711.101369
[4] Kareena, B., Manoj, K.D.: National Knowledge Commission – A Step towards India's Higher Education Reforms on India's Higher Education. International Research Journal of Finance and Economics 53, 46–58 (2010) ISSN 1450-2887
[5] Lin, X., Johansson, M., Boyd, S.P.: Simultaneous routing and resource allocation via dual decomposition. IEEE Transactions on Communications 52(7), 1136–1144 (2004), doi:10.1109/TCOMM.2004.831346
[6] Pandian, R.S.R., Kasiviswanathan, K.S.: Effective Use of Cloud Computing Concepts in Engineering Colleges. In: IEEE International Conference on Technology for Education (T4E), pp. 233–236 (2011)
[7] Sanjay, P.A.: Performance Based Reliability Optimization for Computer Networks. In: Proceedings of the IEEE, Southeastcon 1997, Virginia Tech, Blacksburg, VA (1997)
[8] Taylor, W.: Unlocking the Gates. Princeton University Press (2011)

A Variant of Quantum Genetic Algorithm and its Possible Applications

Pawan Kumar Tiwari and Deo Prakash Vidyarthi

School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India
pawantiwari6@gmail.com, dpv@jnu.ac.in

Abstract. There are many problems solutions of which require a great deal of computation. Over the years, nature inspired computing has become a successful tool to solve such problems. Quantum Genetic Algorithm (QGA), in the recent, often being used and is gaining momentum. A variant of QGA is being proposed in this work, which consist of multiple chains and uses the concept of continuous basis for continuous domain problem.

Keywords: Combinatorial optimization problems, Quantum computing, Continuous basis, Genetic Algorithms.

1 Introduction

Many computational techniques and models have been developed by taking inspiration from natural phenomenon or from deeply observing the living systems. Evolutionary computation, originated during early 1960's, has taken attention of leading researchers, industrialist, and scientists since last decades, due to its wide range of applicability in solving different types of combinatorial optimization problems on a huge search space in a certain time/cost. Particle Swarm Optimization (PSO) technique is inspired by the behavior of the bird flocking. Ant Colony Optimization (ACO) is derived by the ant's behavior for search of food. Genetic Algorithm (GA) is derived from the Darwin's theory of reproduction in nature. Artificial Neural Networks (ANN) is inspired basically by human brain functioning. All these computational methodologies belong to the soft computing discipline [1].

Quantum theory plays an important role in explaining many complicated behaviors of atomic structure. It has provided uniform framework for the construction of various modern physical concepts. It was the greatest scientific achievement of 20^{th} century. The quantum computer concept was first proposed by Feynman in early 80's. In 1985 the concept of quantum Turing machine based on Feynman principle was introduced in the literature [2]. Peoples are working in the direction of constructing a realistic quantum computer. Today's computers use

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 797–811.
springerlink.com

binary bits of data that are either ones or zeros. Quantum computer would use quantum bits or 'qbits' which can be both zeros and ones at the same time. This concept has been borrowed from concept of 'quirks' of quantum mechanics. This 'quirks' gives quantum computers a huge edge in performing particular type of calculations. Recently it was reported in the news paper that a team of physicist of Rice university claimed that they are very near to build a quantum computer by creating a tiny 'electron super highway' which they claim could be useful in producing the new computers that will use quantum particles in place of digital transistors found in today's microchips. In the race to build to a quantum computers, the physicists are using various approaches to create qbits.

Quantum computing forces us to reexamine various branches of computer science specially the branches which have their root in artificial intelligence. Quantum computing is a research area that includes concept like quantum mechanical computers and quantum algorithms. Some well known quantum algorithms had been already reported in the literature [3,4]. During last decade some work has been reported in combining Quantum computing concept with nature inspired evolutionary computation [8, 9, 10, 14]. The performance of the quantum evolutionary algorithm shows that it has better performance than traditional evolutionary algorithm such as conventional genetic algorithm.

Ying had tried to explain the interplay between quantum theory and artificial intelligence and invited the researchers for filling the gaps in this direction [2]. The paper proposes the concept of new multiple chain quantum inspired genetic algorithm for providing the better solution of some NP complete problem [5]. General structures of the problem that can be solved by evolutionary algorithm technique have been also elaborated in detail.

2 Quantum Computing

A 'qbit' is the smallest unit of information in quantum computing, which may be in the 1 state or 0 state or in the superposition of the two states. In Quantum mechanics "State" means all the aspects together, whatever information one can have for the system should be contained in the description of the state. If we say that we know the state of the system at any instant, that means we know everything about the system at that instant that is possible to know. For Quantum systems of a single particle, the information about the position at time t is given by $\Psi(x,t)$. At a given time t, there may not be a definite position of the particle as the particle may be found at various positions with probabilities proportional to $|\Psi(x,t)|^2$. But to know anything about its position at a time t we must know its wave function $\Psi(x,t)$ at a given instant of time t. At a particular time t the wave function Ψ will be a function of x only. For realistic situation wave function should be finite continuous and square integrable.

The wave function $\Psi(x)$ is only one way to represent the state. P.A.M. Dirac who contributed a lot in the formulation [6] and development of quantum mechanics, introduced the ket symbol $\left| \ \right\rangle$ for the abstract state. An identifying symbol which is written inside the ket symbol, such as $\left|\Psi\right\rangle$, denotes a particular state. The state space F is, in general, infinite dimensional. We need infinite number of functions to form a basis for F. The basis functions are so chosen that they are orthogonal to each other, and also the norm of each of them is unity. If we denote the basis functions by $\left|\Phi_i\right\rangle$'s . The orthonormality condition of wave function can be written as $\left\langle \Phi_i \middle| \Phi_j \right\rangle = \delta_{ij}$ where δ_{ij} is 1 if i=j and zero otherwise. This quantity δ_{ij} is called kronecker delta. In this new notation any state vector can be represented in the following way

$$\left|\Psi\right\rangle = \sum_i c_i \left|\Phi_i\right\rangle \tag{1}$$

The expression $\sum_i c_i \left|\Phi_i\right\rangle$ is called a linear combination of $\left|\Phi_i\right\rangle$'s. The number c_i is called the component of $\left|\Psi\right\rangle$ in the direction of $\left|\Phi_i\right\rangle$. For obtaining the component c_n, we take the scalar product of the basis function $\left|\Phi n\right\rangle$ with the given wave function $\left|\Psi\right\rangle$ as given below.

$$\begin{aligned}
\left\langle \Phi_n \middle| \Psi \right\rangle &= \sum_i c_i \left\langle \Phi_n \middle| \Phi_i \right\rangle \\
&= \left\langle \Phi_n \middle| c_1\Phi_1 + \ldots + c_n\Phi_n \right\rangle \\
&= c_1\left\langle \Phi_n \middle| \Phi_1 \right\rangle + c_2\left\langle \Phi_n \middle| \Phi_2 \right\rangle + \ldots + c_n\left\langle \Phi_n \middle| \Phi_n \right\rangle \\
&= c_n\left\langle \Phi_n \middle| \Phi_n \right\rangle \qquad (\because \left\langle \Phi_m \middle| \Phi_n \right\rangle = 0 \ for \ m \neq n) \\
&= c_n.1 \\
&= c_n \\
\therefore \quad c_n &= \left\langle \Phi_n \middle| \Psi \right\rangle
\end{aligned}$$

The probability of any measurement of dynamical variable "D" in state "n" = $\left|c_n\right|^2$

$= \left\langle \Phi_n \middle| \Psi \right\rangle^2$.

And $\left|c_1\right|^2 + \left|c_2\right|^2 + \ldots + \left|c_n\right|^2 = 1$ (from the principal of probability).

This is kind of a discrete basis because we can count them to be the first function, this is second and so on. This kind of basis had been already reported in the area of quantum inspired evolutionary techniques [6].

The paper proposes the expansion of Quantum Computing with continuous functions for evolution inspired computation which is not yet reported in the literature but may have wide role to play. In the field of quantum mechanics the concept of continuous set of functions as a basis is very important. In such a case, the basis function are not indexed by a discrete level I but by a variable which can take any value from $-\infty$ from to $+\infty$. The Set of Dirac delta function $\delta(x-x_0)$ also written as $|x_0\rangle$ is such a basis. We can have infinite number of Dirac delta function each centered at x_0. We can't label them as the first function the second function etc. because between any two unequal values of x_0 there are infinitely many other values. But we can write any function $\Psi(x)$ in terms of these delta functions. Such a basis is called a continuous basis. Note that $|x_0\rangle$ themselves are not part of the state space, as they do not represent state of any realistic system. But any realistic state can be represented as a linear combination of these states. For a continuous basis the expansion of state function $|\Psi\rangle$ will be integration over the whole range of the continuous label describing the basis. For example, using $|x_0\rangle$ basis,

$$|\Psi\rangle = \int_{-\infty}^{+\infty} \Psi(x_0)|x_0\rangle dx_0 \ . \tag{2}$$

In this expansion, $\Psi(x_0)$ is the "component" of $|\Psi\rangle$ in the direction of $|x_0\rangle$. Is such a Continuous basis orthonormal? Such a basis function can be chosen to orthogonal to each other but their normalization is not assured. The delta functions and the plane wave functions are not square integrable hence can't be normalized. For a continuous basis orthonormality can be interpreted in the following manner. A continuous basis $|\lambda\rangle$ is called orthonormal if $\langle\lambda_1|\lambda_2\rangle = \delta(\lambda_1-\lambda_2)$. What does this eqn. mean? If $\lambda_1 \neq \lambda_2$, i.e, if we take different basis functions $|\lambda_1\rangle$ and $|\lambda_2\rangle$, the scalar product is zero. But if $\lambda_1 = \lambda_2$ i.e. square of norm of basis function $|\lambda_1\rangle$ is infinity.

Often the continuous basis function are not themselves valid wave function. i.e. they do not belong to the set of all possible wave function. But any realistic wave function can be expanded in terms of these.

In case of continuous basis probability of getting the position x in the range x_0 to $x_0 + \mathrm{d}\, x_0$ is $|\Psi(x_0)|^2 dx_0$.

3 Problem Structure

Many of the real world problems can't be solved by the traditional mathematical techniques (linear/nonlinear programming). Most of these problems are decision problems or optimization problems. What would be the basic structure of such problems and why the traditional mathematical techniques are unable to find the suitable solution? This paper considers these objectives.

3.1 General Structure and Objective of the Problem

To solve a problem, the objective function may be given by real word system (biology, medical science or socio economy) of arbitrary complexity. Whenever we create a model for a system, this directly indicate that we want to predict the behavior or performance of the system (machine, scheduling, stock market, prediction of diseases etc.) under certain conditions where experimenting on the actual object is impossible either due to cost/time or any other reason. Usually in real word system the problem can't be formulated in terms of direct cause-and-effect relations, as the exact formulation requires a huge number of details. In most of the problems the objective is to find a vector $x \in S$ of free parameters of the system under consideration, such that a certain quality criterion $f : (S, \Omega) \rightarrow (R, U)$ is maximized (or minimized), where R is the set of real number, Ω is the corresponding topology on the domain set S and U is the usual topology on the set of real number.

$$f(\vec{x}) \rightarrow \max \tag{3}$$

f may be a function of single variable (usually in ordinary calculus) or it may be a function of several variables (in most real world problem). In real world problems more than one quality criterion is also of our concern. In this case we can represent f as follows:

$$f \equiv [f_1, f_2, ..., f_n] \tag{4}$$

(In the case of Combinatorial Optimization Problem (COP) the Topology Ω on the set S is the discrete topology).

The aim is to find the global optimum or near optimal solution of the problem. Mathematically global optimum is defined as follows:

Find a vector $\vec{y} \in S$ such that

$$f(\vec{x}) \leq f(\vec{y}) \ \forall \vec{x} \in S. \tag{5}$$

Existence of several local maxima (equivalently local minima) can be defined as follows: Let N_x denote the family of sets of all neighbourhoods of $x \in S$ on the topological space (S, Ω). Then for each $N \in N_x$ we can find a $z \in N$ such that

$$f(\vec{x}) \leq f(\vec{z}) \ \forall x \in N, \forall N \in N_x. \tag{6}$$

Moreover all the restrictions/ constraints of the real world problem can be expressed explicitly or implicitly in the form of the set of function viz. $g_j : S \rightarrow R$, (whenever there will be only one constraint g_j will be denoted by g.)

After applying all the demands of the problems in real world situation we will try to find a subset $F \subseteq S$ of solutions known as feasible solutions and the definition of the set F can be described as follows:

$$F = \{\vec{x} \in S : g_j(\vec{x}) \geq 0\} \tag{7}$$

3.2 Why Traditional Mathematical Techniques Failed

There should be no doubt that traditional mathematical techniques were very successful in many applications of small real world problems. Research is also growing in these fields for providing solutions of complex mathematical real world problem. But theoretically it has its own limitations. For example, linear programming problem (LPP) uses piecewise linearity of relation between f(x) and $h_j(x)$. Although the LPP have the ability to find the solution in a finite number of steps and can be employed at reasonable cost for larger problems, it can't be used due to oversimplifications of non linearity towards linearity that throws away the optimal solutions at a large distance. The mathematical concept behind this is that optimum solution of a LPP is always located in a corner of the polyhedrons made up of restrictions, whereas the optimum solutions of a NLPP can be found anywhere in the interior of the feasible area. Moreover, the extension of LPP to convex or concave, usually purely quadratic programming, yields only gradual improvements of the area of applications. Theoretically it has excluded the possibility of existence of eqn (2), i.e. the existence of several local maxima/minima. Moreover, most of the existed models are in the form of algorithm and there is no mention of the corresponding topology in that space and hence differentiability as well as continuities of the functions makes a concern.

3.3 Approach of the EA Techniques and Reason of Success

In real world problems other important factors like strong nonlinearity, large dimensionality, non differentiability, and noisy and time varying objective functions frequently make the problem very difficult, if not unsolvable. But even in the later case, the identification of an improvement of currently known best solution through optimization is often already a big success for practical problems (e.g. in medical area, stock market etc.). In many cases evolutionary algorithms provides an effective and efficient methods to achieve this. Evolutionary techniques should not be considered as a readymade jacket, but rather as a general concept which can be fitted to most of the real world problem after certain

modification that were beyond the limit by traditional mathematical methods. Once a successful evolutionary computation has been developed it can be incrementally adapted to the problem under considerations, to changes of the requirements of the project, to modifications of the model, and to change of hardware resources [7].

4 Designing of an EA Methods and Scope

In most real world applications the search space is defined by a set of objects, e.g. processors, number of machines, capacity, reliability of each nodes, consumption of energy etc. All of these types of things which are subjected to the optimization problems constitute the phenotype space. On the other hand the evolutionary operators often work on abstract mathematical objects like binary string in the case of genetic operators and graphical structures (set of vertices and set of paths) in case of ACO operators say a evolutotype space. For associating these two viz. phenotype and evolutotype spaces, we need a coding functions or mapping. There are two different approaches for this.

1. For an algorithm, construct a decoding function according to the requirements of the algorithms.
2. Design the representation as close as possible to the characteristic of the phenotype space. In this case, there is no need of looking towards decoding functions desperately.

Each one of them has their own benefit and drawbacks. On one side in the first approach empirical and theoretical results are available for the standard instances of evolutionary algorithms. On the other side a complex coding function can further increase the mathematical difficulty and nonlinearity of the problem taken in hand, which can delay the progress of finding the optimum solution with a significant amount. Most of the work prefers natural problem related representations.

5 Proposed Quantum Genetic Algorithm

We have elaborated QGA in the following subsections.

5.1 Genetic Algorithm (GA)

Genetic algorithms (GA) [11, 12] are derived from the principles of Darwinian natural selection and evolution. The simple genetic algorithm starts off with a population of randomly generated chromosomes, each representing a candidate solution to the concrete problem. It advances towards better chromosomes by applying three natural operators; selection or reproduction, crossover and mutation. The population evolves over time through a successive iteration process

of competition and controlled variation. Each state of population is called generation.

Selection operator stochastically collects the *fitter* member of the population as per a predefined fitness function. The crossover combines two good solutions into potentially better solutions by exchanging genetic materials. Mutation randomly alters the information contained in the population, adding diversity back into the population. Mutation plays a very significant role towards avoiding a premature convergence. Crossover is also termed as exploiter and Mutation as exploration operator.

5.2 Quantum Inspired -GA

QGA borrow fully the concept of quantum mechanics on the top of genetic algorithms. In fact it is a probabilistic algorithm which uses the concept of superposition of quantum states, entanglements and intervention. For encoding the chromosomes of the population Q bit is used and for updating the solution rotation gate is used [15,16,17,18]. A Q bit individual as a string of m Q-bit is defined as follows:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & ..... & \alpha_m \\ \beta_1 & \beta_2 & ..... & \beta_m \\ : & : & ..... & : \\ \eta_1 & \eta_2 & ..... & \eta_m \end{bmatrix}$$

Where $|\alpha_i|^2 + |\beta_i|^2 .... + |\eta_i|^2 = 1, i = 1,2,.....m.$

The basis of state space of a quantum system of m Q bit is:

$$\{\ |0,0....0\rangle, |0,0....1\rangle,...|1,1....1\rangle\ \}.$$

For example suppose we have following three qbits with three pair of amplitudes (assume that each chromosome of qbit has only two component α_i and β_i.):

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

The states can be represented as follows:

$|\Psi\rangle = \sqrt{3}/4\sqrt{2}\ \ |000\rangle + 3/4\sqrt{2}\ |001\rangle + 1/4\sqrt{2}\ |010\rangle - \sqrt{3}/4\sqrt{2}$
$|011\rangle + 3/4\sqrt{2}\ |101\rangle + \sqrt{3}/4\sqrt{2}\ |100\rangle - 1/4\sqrt{2}\ |110\rangle - \sqrt{3}/4\sqrt{2}\ |111\rangle.$

This means that probability to represent the state $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$ and $|111\rangle$ are 3/32, 9/32, 1/32, 3/32,3/32,9/32, 1/32 and 3/32 respectively. Thus the above three bit Q strings contains the information of 8 states. Similarly the state function of four bit Q strings contains the information of 16 states.

5.2.1 Initial Population & Q Bit Encoding

Randomly generate an initial population $P_Q(t)$ as $P_Q(t) = \{ p_1(t), p_2(t), p_3(t), \ldots, p_N(t) \}$, where $p_i(t)$ is defined as follows:

$$p_i(t) = \begin{bmatrix} \alpha_{i1}(t) & \alpha_{i2}(t) & \ldots & \alpha_{im}(t) \\ \beta_{i1}(t) & \beta_{i2}(t) & \ldots. & \beta_{im}(t) \\ \vdots & \vdots & \vdots & \vdots \\ \eta_{i1}(t) & \eta_{i2}(t) & \ldots. & \eta_{im}(t) \end{bmatrix}.$$

5.2.2 Evolutionary Rotation Operations

Basic QGA updating strategies are based on the quantum rotation gate**.** The Quantum Rotation Gate $U(\Delta\theta)$ for a chromosome of length unto 2 had been defined in the literature as:

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}$$

The direction and magnitude of rotation angle $\Delta\theta$ are crucial. Methods of obtaining $\Delta\theta$ directly impact the convergence speed and search efficiency of algorithms. Evolutionary rotation operation is used for moving the phases of chromosomes in each qbits to the phases of those in global optimal chromosomes. By studying the phase relation between these two we can determine it. Updated value of the chromosomes can be determined as follows:

$$\begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \end{bmatrix} = U(\Delta\theta) \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix}$$

In general a Quantum gate acting on Quantum register Consisting of n qbits can be conveniently described by a 2^n x 2^n matrix U given by the Eqn. (8) that will satisfy Eqn. (9) [2].

$$U=\begin{bmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,2^n-1} \\ u_{1,0} & u_{1,1} & \dots & u_{1,2^n-1} \\ \vdots & \vdots & \vdots & \vdots \\ u_{2^n-1,0} & u_{2^{n-}-1,1} & \dots & u_{2^n-1,2^n-1} \end{bmatrix} \tag{8}$$

$$U(U^T)^C = I \tag{9}$$

where $U^T)^C$ denote the transpose conjugate of U.

In general the updated value of chromosome can be given as:

$$\begin{bmatrix} \alpha(t+1) \\ \beta(t+1) \\ \vdots \\ \eta(t+1) \end{bmatrix}_{2^n x 1} = \begin{bmatrix} u_{0,0} & u_{0,1} & \dots & u_{0,2^n-1} \\ u_{1,0} & u_{1,1} & \dots & u_{1,2^n-1} \\ \vdots & \vdots & \vdots & \vdots \\ u_{2^n-1,0} & u_{2^{n-}-1,1} & \dots & u_{2^n-1,2^n-1} \end{bmatrix} \mathrm{X} \begin{bmatrix} \alpha(t) \\ \beta(t) \\ \vdots \\ \eta(t) \end{bmatrix}_{2^n x 1} .$$

Some other Evolutionary Rotation gate operator had been already reported in the literature like Hadamard gate, H_e gate and influenced global gate, each of them talking about chromosomes having two components. The paper also proposes the way of extension of these rotation gate for a chromosome having in general 2^n quality component.

Through the Quantum rotation Gate the conversion between any superposition state can be achieved which increases the diversity of the population and has high degree of parallelism.

5.2.3 Quantum Mutation Operator

Through quantum mutation operator we increase the diversity of individual and reduce the probability of immature convergence so it also increases the searching ability in local space. In general Quantum not operation had been used widely for mutation operation in quantum inspired evolutionary techniques. For this at each generation we generate a random number between 0 and 1. If the generated random number is less than the mutation probability P_m , we select a chromosome randomly and swap the positions of two quality parameter in that. For e.g. if the randomly chosen chromosome is $[\alpha(t) \quad \beta(t) \quad \dots \quad \eta(t)]^T$ and two randomly chosen quality parameter is first and last then after mutation it will be converted into the following manner:

$$\begin{bmatrix} \alpha'(t) \\ \beta'(t) \\ : \\ \eta'(t) \end{bmatrix}_{2^n x1} = \begin{bmatrix} \eta(t) \\ \beta(t) \\ : \\ \alpha(t) \end{bmatrix}_{2^n x1} \quad \text{if rand () < } P_{m..}$$

5.2.4 Perturbation or Disaster Operation

In order to solve the QGA problem being prone to be trapped in local extreme value better we perturb the population. It is found that by using the QGA analysis that when the best individual of the current generation is a local extreme value, it is very hard for the algorithm to extricate itself. Therefore when the best individual does not change in successive generations, the algorithm is trapped in the local extremes. At this point of time a perturbation should be applied to the population to extricate itself out of the local optimal and start a new search.

5.2.5 Proposed Algorithm

Procedure of QGA can be summarized in the following steps:

Step1: At t=0 initialize population $P_Q(t) = \{ p_1(t), p_2(t), \ldots, p_N(t) \}$.

Step2: Decode $P_Q(t)$ in the form of solution according to the problem. Evaluate the fitness value of each individual solution in decoded population $P_Q(t)$ and store the best solution among $P_Q(t)$ into PoP(t)

Step 3: If a stopping condition is satisfied, then output the results; otherwise go on to the following steps.

Step4: t=t+1 update $P_Q(t)$ to $P_Q(t+1)$ by applying evolutionary rotation gate U from left on $P_Q(t)$.

Step5: Decode $P_Q(t+1)$ in the form of solution according to the problem. Evaluate the fitness value of each individual solution in decoded population $P_Q(t+1)$ and store the best solution among $P_Q(t+1)$ and PoP(t) into PoP(t+1)

Step 6: If the disaster step is gotten execute disaster operation.

Step 7: Execute mutation operator.

Step 8: Whether termination is reached else go to step 4.

6 Possible Applications

6.1 Knapsack/Multiple Knapsack Problem (MKP)

It is a combinatorial optimization problem in which for a given set of n items (i=1,2,…n), each having a specific weight w_i and a value p_i the goal is to select a subset of items for packing in a knapsack of capacity c, such that to maximize the total value of the item packed in the knapsack $\sum_{j=1}^{n} p_j.x_j$ (here it is the f of eqn 2.1) and subject to the constraint $\sum_{j=1}^{n} w_j.x_j \leq c$ such that x_j =0 or 1, j=1,2,…n. (here $g = c - \sum_{j=1}^{n} w_j.x_j$). Another constraint in this problem is that no item can be partitioned and once packed can not be taken out of the knapsack. MKP can be thought as a resource allocation problem, where we have m resources (the knapsacks), n objects and object j has a profit p_j. Each resource has its own budget c_i (knapsack capacity) and consumption r_{ij} of resource i by object j. Objective function (f as described in eqn (1)) for MKP will be same as that of knapsack problem described above. But the g_j will look as

$$g_j \equiv c_j - \sum_{j=1}^{n} r_{ij}.x_j \quad i = 1,2,.....m; j = 1,2,.....n \tag{10}$$

There are m constraints in this problem, so MKP is also called m-dimensional knapsack problem. Problems in cargo loading, cutting stock, bin packing, budget control financial management and scheduling problems may also be formulated as Multiple Knapsack Problem [19].

6.2 Traveling Salesman Problem (TSP)/ Multiple TSP

The Goal in TSP is to find the minimal Hamiltonian circuit of the graph that represents the city network. In other words, to find a permutation π of the city indices {1,2,…n} whose traveling length f(π) is minimal where f(π) is given by the following equation.

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \tag{11}$$

In symmetric TSP the Hamiltonian graph is fully connected with symmetric distance metric i.e $d_{ij} = d_{ji}$.

MTSP is a generalization of the TSP, where more than one salesman is allowed to be used in the solution. Moreover the characteristics of the multiple TSP seem more appropriate for real life applications, and it is also possible to extend the problem to a wide variety of vehicle routing problems by incorporating some additional constraints. Possible variations may be the inclusion of time windows, single versus multiple depot and other special restrictions according to the domain of the problem.

6.3 Assignment Problems

In assignment problem the objective is to assign a number of jobs to equal number of machines/persons at a minimum cost/time. It is a special case of linear programming problem known as 0-1 programming problem. In the literature several methods have been reported for generalized assignment problems [20,21].The three Index assignment problems also known as the 3-dimensional assignment problem is a well known NP-hard problem [20]. The 0-1 programming model for this problem looks like as given in the following Eqns.

$$\min \sum_{i\in I,\, j\in J,\, k\in K} x_{ijk}.c_{ijk} \tag{12}$$

Subject to

$$\begin{cases} \sum_{j\in J,\, k\in K} x_{ijk} = 1, \quad \forall i \in I, \\ \sum_{i\in I,\, j\in J} x_{ijk} = 1, \quad \forall k \in K, \\ \sum_{i\in I,\, k\in K} x_{ijk} = 1, \quad \forall j \in J, \end{cases} \tag{13}$$

$x_{ijk} \in \{0,1\}, \quad \forall \{i, j, k\} \in I \times J \times K,$ where I, J, K are three disjoint sets with cardinality of each set taken as n.

A distributed computing system has emerged as a powerful platform for providing higher computational power. The task assignment problem on such systems is a NP-complete problem. It is a mapping which assigns task of a program among different processors of distributed computer systems in order to optimize some characteristic parameters. The goal can be achieved by maximizing the utilization of resources and reducing the communication time between processors. These two criteria of the goal increases the complexity of the problem and a conflict arises in the problem as first criteria requires to distributes the tasks among different processors where second criteria require to distribute them on the same processors. Because of the intractable nature of the task assignment problem, it is desirable to develop good heuristic algorithm/hybrid algorithm according to the need of the problem.

6.4 Vehicle Routing Problem (VRP)

The Vehicle Routing Problem can be described as the problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered customers. As the size of the problem increases, it becomes a tough task to find an optimal solution. This problem also belongs to the NP-Class Problem. Many hybrid approach using neural network, genetic algorithm and quantum inspired evolutionary techniques have been reported in the literature [22].

6.5 Scheduling Problem in a Computational Grid

Scheduling of task is an integrated part of parallel and distributed computing systems. An intensive research has been done in this area and the results of those were widely accepted. But, with the emergence of Open Grid Service Architecture (OGSA) in computational grid, new scheduling models and algorithms are in demand for addressing new concerns in grid environment. The complexity of the scheduling problem increases with the size of the grid and becomes highly difficult to solve it effectively. Several Scheduling problems consider MKP as a sub problem. In such problems, constraints relating to different nodes/grids changes with respect to time depending upon the load of the traffic and sudden requirement of specific type of service. Many evolutionary techniques for providing the better schedule using construction method had been reported [16, 23, 24].

7 Conclusion

As discussed above, there is a variety of problems for which exact algorithm doesn't exists. Evolutionary techniques provide much efficient solutions for such problems. It is the point of research that for what type of problem which evolutionary technique may be applied and how efficiently it is possible to map the parameters of the various applications to that of the different heuristic methods. In recent, ACO, GA, PSO, ANN etc. have been applied on such problems. Sometimes, two or more methods are mixed together making a hybrid algorithm to explore good results for specific type of problem [13, 20]. The concept of Quantum continuous basis given in section 2 and multiple chain concept will be helpful in developing new hybrid quantum inspired evolutionary algorithm for many real world problem.

References

[1] Kosko, B.: Neural Networks and Fuzzy Systems A Dynamical Systems Approach to Machine Intelligence. PHI (1992)

[2] Mingscheng, Y.: Quantum computation, quantum theory and AI. Artificial Intelligence 174, 162–176 (2010)

[3] Grover, K.L.: A fast quantum mechanical algorithm for database search. In: Proc. 28th ACM Symp. Theory Comput., Philadelphia, Pa, pp. 212–221 (1996)

[4] Shor, W.P.: Algorithms for Quantum Computation: Discrete logarthims and factoring. In: Proc. 35th Symp. Found. Comput. Sci., Los Alamitos, CA, pp. 20–22 (1994)
[5] Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
[6] Verma, H.C.: Quantum Physics, 2nd edn. Surya Publications, Ghaziabad
[7] Michalewicz, Z.: A hierarchy of evolution programs: An experimental study. Evolutionary Computation 1(1), 51–76 (1993)
[8] Han, K., Kim, J.: Genetic quantum algorithms and its applications to combinatorial optimization problem. In: Proc. of 2000 Congr. on Evol. Comput., USA, pp. 1354–1360 (2000)
[9] Han, K., Kim, J.: Quantum inspired evolutionary algorithm for a class of combinatorial optimization. Computer Journal of IEEE Trans. on Evol. Comput. 6(6), 55–59 (2000)
[10] Han, K., Kim, J.: A quantum inspired evolutionary algorithm with a new termination criterion, he gate and two phase scheme. IEEE Trans. Evol. Comput. 6(6), 580–593 (2002)
[11] Goldbeg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Pearson (2005)
[12] Mitchell, T.: Machine-Learning. Mcgraw-Hill Series in Computer Science (1997)
[13] Zhao, N., Wu, Z., Zhao, Y., Quan, T.: Ant colony optimization algorithms with mutation mechanism and its applications. Expert System with Applications 37, 4805–4810 (2010)
[14] Hogg, T., Portnov, D.: Quantum optimization. Inform. Sci. 128(3-4), 81–97 (2000)
[15] Li, C.P., Song, P.K., Shang, H.F.: Double chains quantum genetic algorithm with application to neuro-fuzzy controller design. Advances in Engineering Software 42, 875–886 (2011)
[16] Niu, Q., Zhou, F., Zhou, T.: Quantum Genetic Algorithm for Hybrid Flow Shop Scheduling Problems to Minimize Total Completion Time. In: Li, K., Fei, M., Jia, L., Irwin, G.W. (eds.) LSMS 2010 and ICSEE 2010. LNCS, vol. 6329, pp. 21–29. Springer, Heidelberg (2010)
[17] Gu, J., Gu, X., Gu, M.: A novel parallel quantum genetic algorithm for stochastic job shop scheduling. J. Math. Anal. Appl. 355, 63–81 (2009)
[18] Xiao, J., Yan, Y., Zhang, J., Tang, Y.: A quantum inspired genetic algorithm for k-mean clustering. Expert Systems with Applications 37
[19] Okada, S., Gen, M.: Fuzzy multiple choice knapsack problem. Fuzzy Sets and Systems 67(1), 71–80 (1994)
[20] Huang, G., Lim, A.: A hybrid genetic algorithm for the three index assignment problem. Europian Journal of Operational Research 172(1), 249–257 (2006)
[21] Maunder, J., Bhunia, A.K.: Elitist genetic algorithm for assignment problem with imprecise goal. Europian Journal of Operational Research 177(2), 684–692 (2007)
[22] Santosh, H.G., Ochi, L.S., Marinho, E.H., Drummond, A.L.M.: Combining an evolutionary algorithm with data mining for solving a single vehicle routing problem. Neurocomputing 70, 70–77 (2006)
[23] Tiwari, P.K.: Optimizing QoS Parameters in Computational Grid Using ACO, M.tech Thesis, JNU, New Delhi, India (2010)
[24] Khanbary, L.M.O., Vidyarthi, D.P.: A GA based fault-tolerant model for channel allocation in mobile computing. IEEE Trans. Veh. Technol. 57(3), 1823–1833 (2008)

Parameter Tuning of Statcom Using Particle Swarm Optimization Based Neural Network

Sarika Varshney, Laxmi Srivastava, and Manjaree Pandit

Madhav Institute of Technology & Science, Gwalior, M.P., India
sarika.varshney@rediffmail.com,
{srivastaval,manjaree_p}@hotmail.com

Abstract. Improvement of voltage profile in modern power system is a challenging task due to its stressed operating conditions. Use of Flexible AC Transmission System (FACTS) devices like SVC and Stacom can be an economic solution to this problem. Statcom is a second generation FACTS device, which is basically voltage source converter that can be used in voltage control mode or reactive power injection mode. For stable operation and control of power systems it is essential to provide real time solution to the operator in energy control centers.

In this paper, particle swarm optimization (PSO) is proposed for development of feed forward neural networks for estimation of the control and operating parameters of Statcom used for improving voltage profile in a power system. The PSO-ANN provides result with faster convergence speed and avoids the local minima problem.

Two PSO based three-layered feed-forward neural networks are developed to estimate the control/ operating parameters of statcom used for improving voltage profile at various loading condition of power system. The effectiveness of the proposed method is demonstrated on IEEE 30-bus power system. The results obtained clearly indicate the superiority of the proposed approach for parameter tuning of Statcom.

Keywords: Statcom voltage and its phase angle, Reactive power in the connected bus, Reactive power in statcom, Vector Quantization clustering technique, PSO-ANN.

1 Introduction

Introduction of restructuring in emerging power system, and the present changes in generation sources have increased the complexity and uncertainty for power system operation and control. Therefore, it is likely that power systems have to be operated under stressed conditions and may lead transmission systems to work closer to their voltage and line loading limits [1]. Reinforcing a power system can be done by increasing the voltage level or adding transmission lines. However, these solutions require considerable investment. Flexible AC Transmission System (FACTS) devices can be an economic solution to these problems [2].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 813–824.
springerlink.com

For providing voltage support, shunt FACTS devices, such as Statcom and SVC are typically used. Statcom is a second generation FACTS device used for shunt reactive power compensation. The principle of statcom is that by adjusting reactive power, the voltage magnitude of the connected system can be controlled. The statcom is a combination of a voltage source converter and a reactance, which is connected in shunt to power system. The converter supplies leading current to the AC system if the converter output voltage is made higher than the corresponding AC system voltage and then it supplies reactive power to the AC system by capacitive operation. Conversely, the converter absorbs lagging current from the AC system, if the converter output voltage is made less than the AC system. In this condition, it absorbs reactive power from the AC system by inductive operation. If the output voltage is equal to AC system voltage, no reactive power will be exchanged.

Like other FACTS devices, statcom is an expensive device; therefore the device is to be installed at optimal location so that voltage profile of the power system may be improved effectively. Simple heuristic approaches are usually applied for determining the optimal location of FACTS devices in a small power system. However, more scientific methods are required for placing and sizing FACTS devices in a larger power network. FACTS sizing and allocation constitutes a challenging problem in power systems. Generator buses where voltages are regulated by the generator do not need statcom installation. Shunt FACTS devices like statcom, SVC are normally connected to the load bus in a power system [3-5].

Since operating condition of a power system is dynamic in nature, therefore the control parameters of statcom are required to be tuned for proper operation of the power system. Conventional method used for tuning control parameters of statcom like Newton Raphson method needs to be run for any change in operating state and therefore is time consuming. Artificial neural networks are proposed here for this task, as they have ability to synthesize complex mappings accurately and rapidly for estimation of the control and operating parameters of Statcom used for improving voltage profile in a power system.

The back-propagation (BP) algorithm is the most commonly used technique for training neural network, but it suffers from slow convergence speed and it easily gets stuck in local minima [6]. Particle Swarm Optimization (PSO) algorithm can be a solution to avoid this problem. It is a population based stochastic optimization technique developed by J. Kennedy and R. Eberhart in 1995. It models the cognitive as well as the social behavior of a flock of birds (solutions) which are flying over an area (solution space) in search of food (optimal solution) [6]. PSO has been applied to train neural networks in place of using back-propagation algorithm and found to be faster than back-propagation algorithm [7-12].

In this paper, statcom has been used to improve voltage profile of a power system and the statcom has been connected to an appropriate load bus. This has been observed that the reactive power requirement from statcom to maintain the connected bus voltage constant depends on the operating condition of the power system. In this paper, two PSO-ANNs are developed, one to estimate the statcom voltage and its phase angle and other PSO-ANN to estimate the reactive power in

the statcom connected bus and in the statcom at various loading conditions. The effectiveness of the proposed approach has been demonstrated on IEEE 30-Bus system.

2 Statcom for Voltage Profile Improvement

When the statcom is used in voltage control mode it can be treated as an augmented PV bus. In this mode of operation the specified quantities are active and reactive power injections and voltage magnitude. The state variables to be solved are, i) the voltage *Vs* of the statcom, ii) angle δ_s of statcom voltage and iii) the voltage angle δk of the bus to which statcom is connected. Thus three equations are needed to solve for the state variables associated with statcom connected bus. Two of these equations are the modified real and reactive power equations associated with statcom connected bus *k* which can be expressed as

$$P_k(mod) = \sum_{j=1}^{N} |V_k||V_j||Y_{kj}| \cos(\theta_{kj} - \delta_k + \delta_j) + |V_k||V_s||Y_s| \cos(\theta_s - \delta_k + \delta_s) \quad (1)$$

$$Q_k(mod) = \sum_{j=1}^{N} |V_k||V_j||Y_{kj}| sin(\theta_{kj} - \delta_k + \delta_j) + |V_k||V_s||Y_s| sin(\theta_s - \delta_k + \delta_s) \quad (2)$$

The additional equation needed is the reactive power supplied by the statcom to the AC system, which can be expressed as

$$Qs = -Im[Vs*Ys(Vs-Vk)]-|Vs|2|Ys|sin(\theta s)+|Vk||Vs|Ys|sin(\theta s-\delta k+\delta s) \quad (3)$$

These equations are linearized and solved along with other load flow equations associated with remaining buses in the system using Newton Raphson method [5].

3 Methodology

The schematic diagram of the methodology carried out in this paper is shown in the Fig.1. A large number of load patterns were generated by perturbing the real and reactive loads at various buses randomly in wide range. Statcom is connected at appropriate bus. Statcom equations (1)-(3) were solved along with AC load flow equations to compute the voltage *Vs* of the statcom, angle θs of statcom voltage, the reactive power *Qs* in the statcom and *Qk* in the statcom connected bus *k* to maintain the statcom connected bus voltage constant at different system loading conditions. The real and reactive loads, total real and total reactive power loads are considered as the inputs for the neural networks PSO-ANN1 and PSO-ANN2 trained using particle swarm optimization algorithm. Vector Quantization clustering technique is used for reducing the dimension of input data. The outputs for PSO-ANN1 are voltage magnitude and phase angle of statcom voltage, while the outputs for PSO-ANN2 are reactive powers in bus *k* and in the statcom.

The number of inputs and outputs are decided before starting the training of a PSO-ANN, while the optimum number of hidden neurons is obtained by having some trials. The PSO-ANN architecture which requires less training time and

provides more accurate testing results is considered as the optimum structure of a PSO-ANN. The neural networks PSOANN1 and PSOANN2 have been developed for estimation of statcom voltage magnitude and phase angle and for estimation of reactive powers in the connected bus and statcom respectively as shown in Fig. 1.

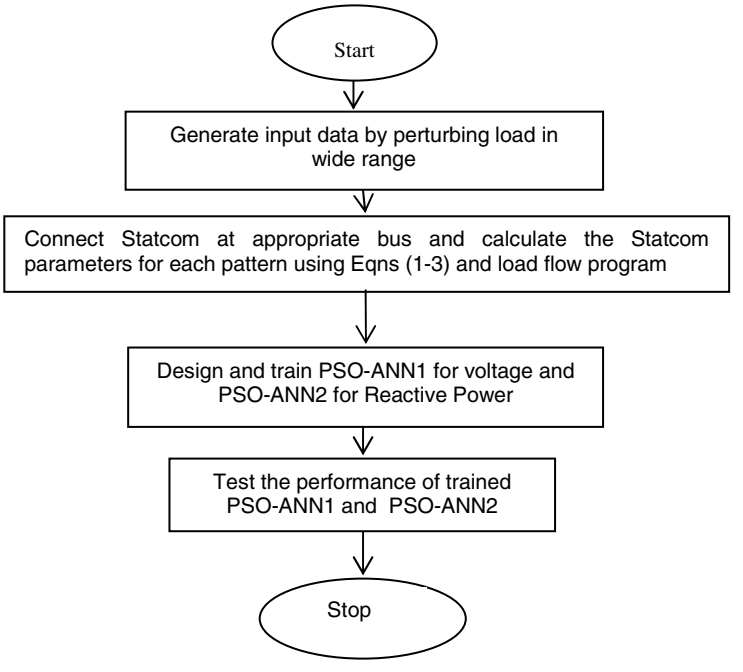


Fig. 1. Schematic diagram for developing two PSO-ANNs for tuning of statcom parameters

3.1 Particle Swarm Optimization Technique

PSO is an evolutionary computation technique developed by Eberhart and Kennedy in 1995, and was inspired by the social behavior of bird flocking and fish schooling [13]. It utilizes a population of individuals, called particles, which fly through the problem hyperspace with some given initial velocities. In each iteration, the velocities of the particles are stochastically adjusted considering the historical best position of the particles and their neighborhood best position, where these positions are determined according to some predefined fitness function. Then, the movement of each particle naturally evolves to an optimal or near-optimal solution.

PSO technique finds the global best solution by simply adjusting the trajectory of each individual toward its own best location and toward the best particle of the entire swarm at each time step (iteration). The PSO method is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. In the PSO algorithm, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each

particle, according to its own flying experience and the flying experience of the other particles in the search space.

The position vector and the velocity vector of the i^{th} particle in the d-dimensional search space can be represented as $X_i=[x_{i1},x_{i2},\ldots\ldots\ldots,x_{id}]$and $V_i=[v_{i1}, v_{i2},\ldots\ldots\ldots,v_{id}]$ respectively. According to a user defined fitness function, say the best position of each particle (which corresponds to the best fitness value obtained by that particle at iteration k) is $P_i=[p_{i1}, p_{i2},\ldots, p_{id}]$, and the fittest particle found so far at iteration k is $P_g=[p_{g1}, p_{g2},\ldots, p_{gd}]$. Then, the new velocities and the positions of the particles for the next fitness evaluation are calculated using the following two equations:

$$vi(k) = w_i.vi(k-1) + c_1.rand_1(p_i - xi(k-1) + c_2.rand_2\left(p_g - xi(k-1)\right) \tag{4}$$

$$xi(k) = xi(k-1) + vi(k) \tag{5}$$

Where c_1 and c_2 are constants known as acceleration coefficients, and $rand_1$ and $rand_2$ are two separately generated uniformly distributed random numbers in the range [0, 1]. The first part of (1) known as “inertia” or “momentum” or “habit” represents the previous velocity. This provides the necessary momentum for particles to roam across the search space. The second part, known as the "cognitive" or “memory” component, represents the personal thinking of each particle. The cognitive component encourages the particles to move toward their own best positions found so far. The third part is known as the "social knowledge" or “cooperation” component, which represents the collaborative effect of the particles, in finding the global optimal solution. The social component always pulls the particles toward the global best particle found so far.

3.2 Particle Swarm Optimization-Artificial Neural Network

The PSO algorithm is too different from any traditional methods of ANN training. PSO not only trains one network but it can train multiple networks. PSO selects a number of neural networks and initialize them to some random weights and start training each one. PSO compares fitness (error) of these neural networks. The network with the least error is taken as the global best [13].

In PSO-ANN model as shown in Fig.2., each neural network (i.e. particle) contains a position and velocity. The position is related to the weight of neural network W, while the velocity here refers to updating of the ANN’s weights ΔW. The velocity is used to control how much the position i.e. weight is updated. The particles are evaluated using a fitness function.The fitness function of each particle is the value of the error function evaluated at the current position vector of the particle [9].Particles have a tendency to duplicate their individual past behavior that has been successful (cognition) as well as to follow the successes of the other particles (socialization). The neural network weight matrix is rewritten as an array to form a particle. Particles are then initialized randomly and updated afterwards according to equation (6) and (7):

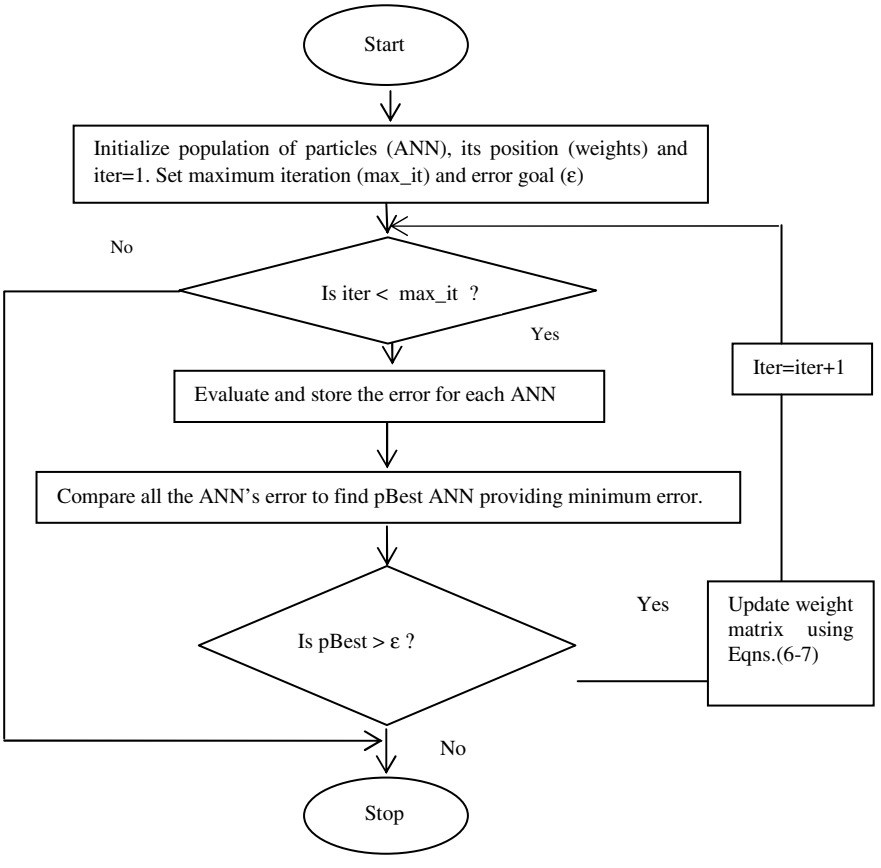


Fig. 2. Flow chart of PSO-ANN

$$W(t+1) = W(t) + \Delta W(t+1) \tag{6}$$

$$W(t+1) = w.\Delta W(t) + c_1.rand().[\ pBest(t) - W(t)\ ] + c_2.rand().\ [gBest(t) - W(t)] \tag{7}$$

where w, c_1 and c_2 are inertia, cognitive and social acceleration constants respectively. Inertia constant controls the exploration and exploitation of the search space because it dynamically adjusts velocity. Cognition and social are the acceleration constants which changes the velocity of a particle towards *pBest* and *gBest*, (generally somewhere between *pBest* and *gBest*).

The particles have memory and each particle keeps track of previous best position and corresponding fitness. The previous best value is called as *pBest*. Thus, *pBest* is related only to a particular particle. It also has another value called *gBest* which is the best value of all the particles pbest in the swarm. The basic concept of PSO technique lies in accelerating each particle towards its p*Best* and the *gBest* locations at each time step. Acceleration has random weights for both *pBest* and *gBest* locations.

3.3 Vector Quantization Clustering Technique

In a neural network application, if a large dimension input data are used, then the number of weights will increase and this will adversely affect the training of neural network. To overcome this problem, dimension of the input data must be reduced. An unsupervised learning Vector Quantization clustering technique is adopted here in this paper to reduce the dimension of input data [14],[15]. In this method, the first pattern creates a starting cluster. And subsequent clusters are formed on the basis of Euclidean distance between input patterns and different existing cluster centers. Euclidean distance between input pattern and any allocated cluster centre is calculated as:

$$d = \left\|X^p - C_j\right\| = \left[\sum_{i=1}^{n}(X_i^p - C_{ji})^2\right]^{1/2} \tag{8}$$

Where, Xp is pth input vector, Cj is jth cluster and d is Euclidean distance. If the Euclidean distance of any input vector Xp from cluster k (i.e. Ck) is less than the Euclidean distance of Xp from cluster j (i.e. Cj), then k is selected as closest cluster to the input vector Xp. The cluster k can be selected as closest cluster using Eq. (8)

$$\left\|X^p - C_j\right\| < \left\|X^p - C_j\right\| \quad \begin{cases} j = 1, \ldots., m \\ j \neq k \end{cases} \tag{9}$$

Where, M is the number of allocated clusters. Once the closest cluster 'k' for the input vector has been determined, the distance $\left\|X^p - C_j\right\|$must be tested against the threshold distance ρ, such as:

If $\left\|X^p - C_j\right\|$< ρ, the pth input pattern is assigned to kth cluster, i.e. Ck.

Otherwise, if $\left\|X^p - C_j\right\|$ > ρ then a new cluster is allocated to pth input pattern. Every time the closest cluster center selected on the basis of comparison with threshold r is updated as:

$$C_x = \frac{1}{n}\sum_{X \in S_n} X \tag{10}$$

Where, Sn denotes the set of input vectors and n is number of input vectors in a cluster. The process of finding closest cluster is repeated for all the system variables $S_j\{S_j=[s_{1j}; s_{2j}; s_{3j}; \ldots; s_N]^T; j = 1; 2; 3; \ldots; n\}$: The variable nearest to the cluster centre in a group has been selected as the representative variable for that cluster. These representative variables are used as the selected input features for the proposed PSO-ANN. In the present paper VQ clustering technique has been employed for reduction of dimension of input data for each neural network.

4 Solution Algorithm

The solution algorithm for statcom voltage and reactive power estimation has been summarized as follows:

i. Newton-Raphson load flow program is run to determine voltage profile of the power system in base case condition.

ii. A statcom is connected at an appropriate load bus to maintain its voltage 1.0 pu.
iii. With the statcom in power system, a large number of load patterns are generated randomly by perturbing the load at various buses.
iv. For each loading scenario, Newton-Raphson load flow program along with statcom equations is run to calculate statcom voltage and reactive power to maintain 1.0 pu voltage at the statcom connected bus.
v. Two PSO-ANNs are developed for statcom voltage and reactive power as discussed below-
   a) For PSO-ANN, a population of ANN's is constructed, with different initial weights.
   b) For each ANN (particle), iterate over the training data set and calculate the sum of squared error.
   c) Compare all the ANN's error, to find the best ANN in the neighborhood.
   d) If one of the network has achieved the minimum error required (error goal) record its weights and go to step (f) otherwise go to step (e).
   e) For each ANN, apply the PSO algorithm to update its position i.e. its weight (W) and velocity vector i.e. (ΔW).
   f) The PSO-ANN has been trained to provide the set error goal.
vi. Performance of both the PSO-ANNs is evaluated for testing patterns by comparing voltage and reactive power values as obtained from NR method

5 Result and Discussion

The effectiveness of the proposed method is demonstrated on a IEEE 30-bus system [17] for estimating statcom voltage and reactive power to maintain the statcom connected bus voltage constant (1.0 p.u.) at different loading conditions. This system comprises of 1 slack bus, 5 PV (generator) buses, 24 PQ (load) buses and 41 lines. Perturbing the active and reactive loads at various buses in wide range (±30%), 600 loading patterns were generated. For each of the loading pattern, using NR load flow solution and (1) - (3), Statcom voltage and reactive power were evaluated.

The active and reactive loads (non-zero) at various buses were considered as possible inputs for the PSO-ANNs, which are 42. The dimension of the input data has been reduced using VQ cluster technique. Total 10 input variables were selected using threshold value of $\rho = 0.03138$. The selected 10 input variables out of total 42 variables are real loads at bus nos. 2, 5, 7, 14, 21 and reactive loads at bus nos. 5, 8, 10, 21, 26. In addition to these 10 inputs, total real and reactive loads are also used as inputs for PSO-ANNs, thus making the total inputs 12 for the PSO-ANNs. The optimum structures of the two PSO-ANNs (12-37-2), and (12-42-2) were obtained by having some trials. Out of 600 patterns, 480 patterns were used for training the PSO-ANNs, while remaining 120 patterns were used for testing the performance of the trained PSO-ANNs.

Testing performance of PSO-ANN1 is shown in Table 1 that compares the statcom voltage and angle estimated by PSO-ANN1 with the actual value of

statcom voltage and angle for 10 testing patterns only due to limited space. Testing performance of PSO-ANN1 for all the 120 previously unseen patterns is shown in Fig. 3 and Fig. 4 for statcom voltage magnitude and phase angle respectively. As can be observed from Fig.4 and Fig. 5, the trained PSO-ANN1 provides accurate estimation for statcom voltage magnitude and phase angle for all the testing patterns. The maximum error is 2.468% and root mean square error is 0.0719p.u.

Table 2 compares the reactive powers in the statcom connected bus and in statcom as estimated by the trained PSO-ANN2 with those as calculated by AC load flow for 10 testing patterns only. However, testing performance of the trained PSO-ANN2 for all the 120 testing patterns is shown in Fig. 5 and Fig. 6. As can be observed from Fig. 5 and Fig. 6, the trained neural network PSO-ANN2 provides accurate values of reactive power all the testing patterns. The trained PSO-ANN2 provided a maximum error of 2.895% while the rms error was 0.0002 p.u.

Table 1. Testing performance of PSO-ANN1(12-37-2) for Statcom voltage and angle

	Statcom Voltage(p.u.)				Statcom Voltage Angle (degree)			
TP	Actual	PSO-ANN	Error(p.u.)	% Error	Actual	PSO-ANN	Error(p.u.)	% Error
1	1.0017	1.00344	-0.00174	0.17413	-18.2737	-18.6902	0.4165	2.27924
2	1.0023	0.99979	0.00251	0.25024	-20.182	-20.074	-0.10797	0.53496
3	1.0009	1.00236	-0.00146	0.14554	-19.3161	-19.5658	0.24968	1.29261
4	1.0004	1.00059	-0.00019	0.01857	-17.2315	-17.1568	-0.07472	0.43361
5	0.9991	1.00419	-0.00509	0.50949	-16.8841	-17.076	0.19186	1.13632
6	0.9995	1.00303	-0.00353	0.3536	-16.7417	-16.929	0.18725	1.11848
7	1.0023	1.00261	-0.00031	0.03077	-18.443	-18.4517	0.00872	0.04729
8	0.9982	0.99973	-0.00153	0.15362	-16.8372	-17.1839	0.34666	2.05888
9	1.003	0.99951	0.00349	0.34803	-19.8503	-19.5326	-0.31773	1.60063
10	1.0021	0.99921	0.00289	0.28811	-20.1944	-19.9785	-0.21587	1.06897

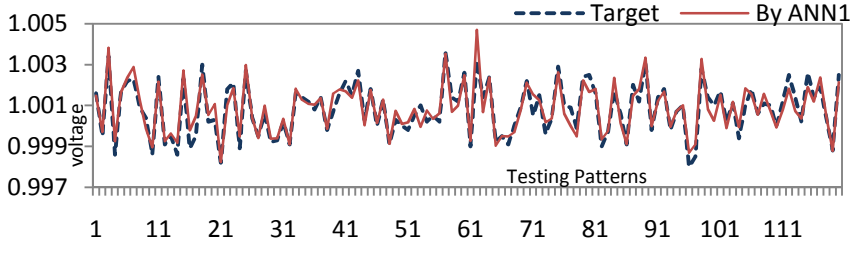


Fig. 3. Testing performance of PSO-ANN1 for Statcom Voltage Estimation

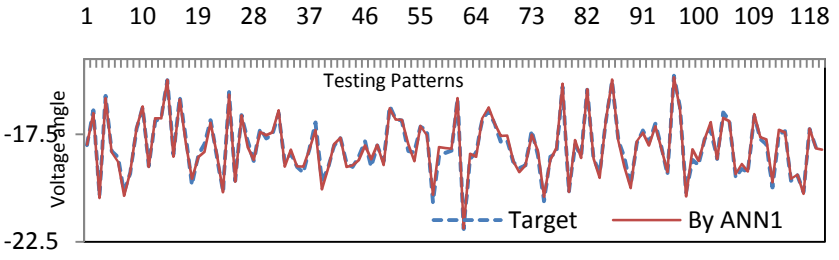


Fig. 4. Testing performance of PSO-ANN1 for Statcom Voltage Angle Estimation

Table 2. Testing performance of PSO-ANN2(12-42-2)

TP	Reactive Power at Bus (p.u.)				Reactive Power for Statcom (p.u.)			
	Actual	By PSO-ANN	Error(p.u.)	% Error	Actual	By PSO-ANN	Error(p.u.)	% Error
1	0.0272	0.02718	0.00002	0.0766	-0.0273	-0.02726	-0.00004	0.1418
2	0.0225	0.02255	-0.00005	0.20173	-0.0225	-0.0226	0.0001	0.45527
3	0.0158	0.01614	-0.00034	2.13363	-0.0159	-0.01619	0.00029	1.81303
4	0.0174	0.01715	0.00025	1.42738	-0.0175	-0.01723	-0.00027	1.56167
5	0.0077	0.00789	-0.00019	2.4073	-0.0077	-0.00789	0.00019	2.4515
6	0.0212	0.02166	-0.00046	2.19123	-0.0213	-0.02173	0.00043	1.99974
7	0.0261	0.02605	0.00005	0.18434	-0.0262	-0.02611	-0.00009	0.3312
8	0.022	0.02204	-0.00004	0.17212	-0.022	-0.02209	0.00009	0.41547
9	0.0121	0.01237	-0.00027	2.20607	-0.0121	-0.01241	0.00031	2.54667
10	0.0141	0.01405	0.00005	0.381	-0.0141	-0.01409	-0.00001	0.05322

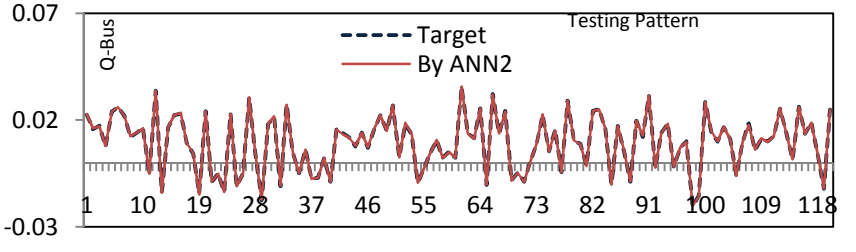


Fig. 5. Testing perform ance of PSO ANN2 for Bus Reactive Power Estimation

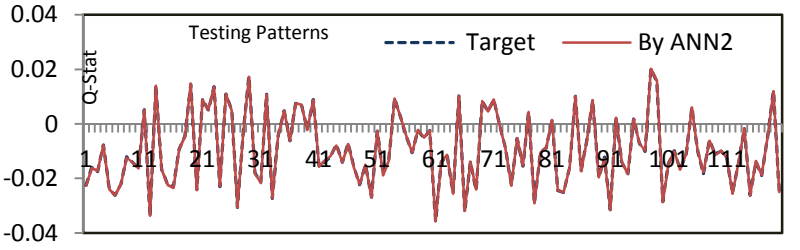


Fig. 6. Testing performance of PSO-ANN2 for Statcom Reactive Power Estimation

6 Conclusion

In this paper, PSO-ANN based approach has been proposed for estimating the control parameters of statcom used for improving voltage profile under varying load conditions of power system. The parameters of the statcom have been tuned such that voltage at the statcom connected bus is maintained at 1.0pu. Two PSO-ANNs were developed, one for estimation of statcom voltage magnitude and phase angle and other for estimation of reactive powers in statcom connected bus and in statcom respectively.

The effectiveness of the proposed method is implemented on IEEE 30-Bus system. The results clearly have shown the faster and accurate estimation of operating/ control parameters of the statcom installed for improving the voltage profile. As the approach is fast and accurate and is able to reduce significantly the training time of PSO-ANNs, this approach can be used for statcom parameter tuning in practical power systems.

Acknowledgement. The authors sincerely acknowledge the financial support provided by Department of Science and Technology (D.S.T), India under Research grant No.SR/S3/EECE/0064/2009 dated 22-01-10 and the Director, MITS, Gwalior, India to carry out this work.

References

[1] Kundur, P.: Power system stability and control. McGraw-Hill, New York (1994)
[2] Hingorani, N.G., Gyugyi, L.: Understanding FACTS; Concepts and Technology of Flexible AC Transmission Systems. IEEE Press, New York (2000)
[3] Mori, H., Goto, Y.: A parallel tabu search based method for determining optimal allocation of FACTS in power systems. In: Proc. of the International Conference on Power System Technology (PowerCon 2000), vol. 2, pp. 1077–1082 (2000)
[4] Yorino, N., El-Araby, E.E., Sasaki, H., Harada, S.: A new formulation for FACTS allocation for security enhancement against voltage collapse. IEEE Trans. on Power Systems 18(1), 3–10 (2003)
[5] Acha, E., Fuerte, C.R., Ambriz, H., Angeles, C.: FACTS: Modelling and Simulation in Power Networks. John Willey & Sons, Ltd.

[6] Haykin, S.S.: Neural Networks- A Comprehensive Foundation, 2nd edn. Prentice-Hall (1998) ISBN 1327-3350
[7] Mohaghegim, S., del Valle, Y., Venayagamoorthy, G.K., Harley, R.G.: A comparison of PSO and the back propagation for training RBF neural networks for identification of a Power System with Statcom. In: Proc. on Swarm Intelligence Symposium, pp. 381–384 (2005)
[8] Da, Y., Xiurun, G.: An improved PSO-based ANN with simulated annealing technique. Neurocomputing 63, 527–533 (2005)
[9] Chau, K.W.: Application of a PSO-based neural network in analysis of outcomes of construction claims. Automation in Construction 16, 642–646
[10] Ribero, P.F., Kyle Schlansker, W.: A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control
[11] Geethanjali, M., Mary Raja Slochanal, S., Bhavani, R.: PSO trained ANN based differential protection scheme for power transformers. Neuorocomputing 71, 904–918 (2008)
[12] Malviya, R., Pratihar, D.K.: Tuning of neural networks using particle swarm optimization to model MIG welding process. In: Swarm and Evolutionary Computation, doi:10.1016/j.swevo.2011.07.001
[13] Hu, X., Shi, Y., Eberhart, R.: Recent Advances in Particle Swarm. In: Proceedings of the Congress on Evolutionary Computation, USA (2004), doi:6.2004
[14] Karayiannis, N.B.: Soft learning vector quantization and clustering algorithms based on mean type aggregation operators. Int. J. Fuzzy Syst. 4(3) (2002)
[15] Pandey, S.N., Tapaswi, S., Srivastava, L.: Growing RBFNN-based soft computing approach for congestion management. Neural Comput. & Applic. 18, 945–955 (2009), doi:10.1007/s00521-008-0205-3
[16] Pandey, S.N., Pandey, N.K., Tapaswi, S., Srivastava, L.: Neural Network-Based Approach for ATC Estimation Using Distributed Computing. IEEE Transactions on Power Systems 25, 1291–1300 (2010)
[17] Saadat, H.: Power System Analysis. Tata Mc-Graw-Hill Publishing Company Ltd.

Determining the Effects of Single Input Layer as Angular Velocity of Rotor Blade on Blade's Frequency Parameters by Regression Based Neural Network Method

Atma Sahu[1,*] and S. Chakraverty[2]

[1] Coppin State University, Baltimore MD 21216, USA
asahu@coppin.edu
[2] National Institute of Technology, Rourkela - 769 008, Orissa, India
sne_chak@yahoo.com

Abstract. The aim of this paper is to demonstrate the use of Regression Based Neural Network (RBNN) method to study the problem of the natural frequencies of the rotor blade for micro unmanned helicopter [3]. The training of the traditional ANN (Artificial Neural Network) model and proposed RBNN model has been implemented in the MATLAB environment using NNT (Neural Network Tools) built-in functions. The graphs for angular velocity (Omega) of the micro-unmanned helicopter are plotted for estimation of the natural frequencies (f1, f2, f3) of transverse vibrations. The results obtained in this research show that the RBNN model, when trained, can give the vibration frequency parameters directly without going through traditional and lengthy numerical solutions procedures. Succeeding this, the numerical results, when plotted, show that with the increase in Omega, there is increase in lagging motion frequencies. Additionally, it is found that the increase in the lower mode natural frequencies is smaller than that of the higher modes. This finding is in agreement with the results reported in earlier research [3],[4],[5] carried out by employing Rayleigh-Ritz and FEM respectively.

Keywords: Transverse vibrations, artificial neural network, harmonic motion, mean square error, micro-unmanned helicopters, Rotor Blade Vibrations.

1 Introduction

The micro-unmanned helicopters are quite different from the conventional manned helicopters in their design scheme. Therefore, in the case of micro-unmanned helicopter, the rotor mechanism is altered in order to optimize the

* Corresponding author.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 825–832.
springerlink.com

manufacturing costs [3] without compromising on its needed functionality. In this paper, however, for the prototype engineering design requirements, the vibrations of helicopter rotor blades, whether manned or unmanned, are of a major concern. The purpose of this paper is to use a Regression Based Neural Network (RBNN) method [1] to solve the problem of studying the natural frequencies of the rotor blade for micro unmanned helicopter [3]. With this R&D effort resulting in appropriate mathematical calculations, the design engineers are able to overcome blade resonance problems (maybe by putting damper on the blade or any other vibrations correction method).The authors choose not to go into the fluid (air) resistance motion problem of blades' airfoil system (Appendix Figure 5).

2 Transverse Vibrations Analysis

In this paper, the rotor manipulation mechanism is based on the use of the inertia characteristic of the rotor and its elastic features as considered by Jungang Lu [2]. Also, an equally important characteristic in rotor parametric manipulation is the blade shape change that can be affected by the leading and trailing edges of the entire airfoil system (Appendix Figure 8). However, the authors in this research paper will limit the scope to RBNN-based analysis of the transverse vibrations of the rotor blade. Also, it is reasonable to assume that the blade length is very large compared to its width. For this reason, Euler-Bernoulli beam theory is adequate for our purposes. The scheme of the blade and notations (see Appendix Figure 8) are adopted in this paper from Lü [3] to make comparisons of this work easier and comprehensible. In this paper, ANN model for blade vibrations has been undertaken. The training of network is performed using the pattern calculated with the help of Boundary Characteristic Orthogonal Polynomials (BCOPs) in the Rayleigh-Ritz method.

We adopt below the kinetic energy (KE) and potential energy (PE) equations as derived by Lü et al. [3]. Considering the conditions of small deflections, the KE of the rotor blade of length L is given by T as follows:

$$T=\frac{1}{2}\int_0^L \rho\left\langle \dot{u}^2+\left[u^2+(a+x)^2\right]\dot{\theta}^2+2\dot{\theta}\dot{u}(a+x)\right\rangle dx \quad ;$$

and PE is given by U as follows:

$$U=\frac{1}{2}\int_0^L EIu_x^2 dx+\frac{1}{2}\int_0^L \left\langle \frac{1}{2}\rho\dot{\theta}^2(L^2-x^2)+\rho\dot{\theta}^2 a(L-x)\right\rangle (u_x)^2 dx$$

For harmonic motion, the blade deflection is given by $u(x,t)=Y(x)\sin(\omega t+\phi)$; using u(x) in T and U above, Lagrangian is obtained. Following notations are used: L = Length of the blade (m), ρ = mass in unit length of the blade (kg/m),

$\omega = \dot{\theta}$ = angular velocity of rotor,EI=flexural rigidity of the blade (N m^2), XO'Y = Inertia reference frame, xOu = Flying reference frame, V_{ji} = Hidden layer weights, W_{kj} = Output layer weights.

Substituting the linear combination of BCOPs in the Rayleigh-Ritz method for T and U, we may turn it to a standard eigenvalue problem. The solution of the standard eigenvalue problem then gives the natural frequencies at various rotational speeds [2]. The computations have been carried out by taking EI=1.392 Nm, L=0.15 m, ρ=0.1260 Kg/m. [2]. As such natural frequencies have been computed for the blade at various rotational speeds for the simulation in RBNN model. In the following paragraphs, ANN architecture is described for estimation of natural frequencies for given values of Omega which is the angular velocity parameter.

3 Identification of the RBNN Model: Solution Technique

Three-layer architecture for regression based artificial neural network approach is considered here to understand the proposed model for solving the present problem. Figure 1 show the neural network used in the process. The input layer consists of

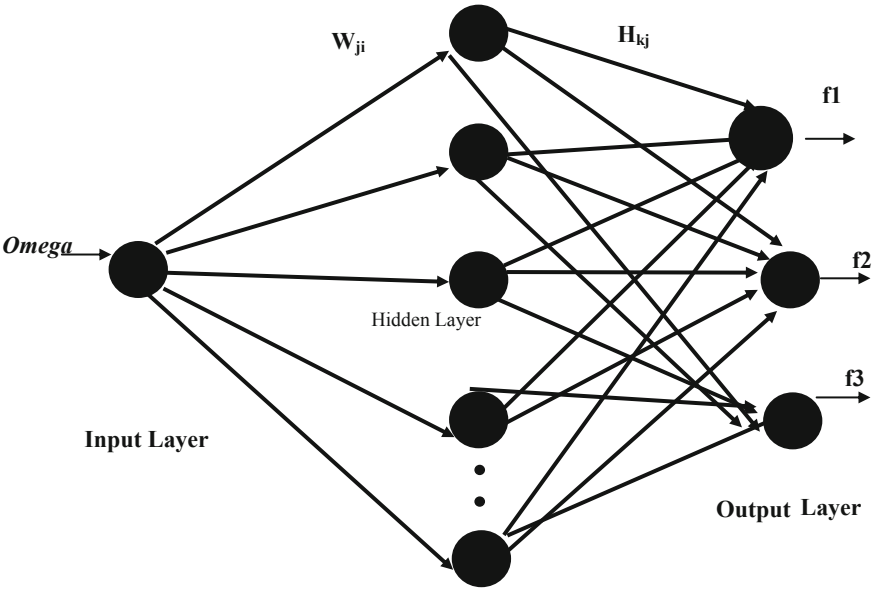


Fig. 1. ANN architecture used for estimation of frequencies for given values of Omega

single input as Omega and the output layer consists of three outputs in the form of the corresponding frequency parameter f1, f2 and f3. Three cases of the number of nodes depending upon the proposed parameter of the methodology have been considered in the hidden layer to facilitate a comparative study on the architecture of the network. The output of the network is computed by regression analysis combined with neural activation function performed at two stages, i.e., the stage of hidden layer and the stage of output layer. Number of neurons in the hidden layer depends upon the degree of the regression polynomial that is used to fit the data between input and output. If we consider a polynomial of degree n, then number of nodes in hidden layer will be (n+1) and the corresponding (n+1) coefficients of this polynomial (say, a_i, $i = 0, 1, ..., n$) are taken as the initial weights from input layer to the hidden layer. Architecture of the network for a polynomial of nth degree is shown in Figure 1.

4 Numerical Results

The training of the traditional ANN (Artificial Neural Network) model and proposed RBNN model has been implemented in the MATLAB environment using NNT (Neural Network Tools) built-in functions. Also, in the following paragraphs, the graphs for angular velocities (Omega) of the micro-unmanned helicopter are plotted for estimation of the natural frequencies (f1, f2, f3).

4.1 The Experiment 1

The training of the traditional ANN model and proposed RBNN model has been implemented for estimation of the frequencies with respect to omega values. In the traditional model, the output of the network is computed by built-in transfer functions, namely, tan-sigmoid (tansig) and linear (purelin) of the NNT (Neural Network Tool) performed at two stages, i.e., the stage of hidden layer and the stage of output layer. The connection weights interconnecting the neurons between different layers are taken through a random number generator built-in function in the NNT. The neural network based on this feedforward back propagation algorithm has been trained with Levenberg-Marquardt training function of the NNT.

4.2 The Experiment 2

In proposed RBNN model, regression polynomials of degree three are fitted to the training patterns. The coefficients of this polynomial are taken as the connecting weights for the hidden layer, as described earlier. The output of the neurons in the

hidden layer is calculated using activation function. At this stage, the error of the RBNN model is calculated and a decision is taken as to whether the network has been trained or not. If the tolerance level of the error is not achieved, the procedure is repeated; otherwise, we say that the network has got trained. In this case the network has been converged with the desired accuracy as shown in the Figure 1 for the problems under consideration. The output of the network f1, f2 and f3 and the MSE (Mean square error) between neural and desired output is calculated. In this Figure, f1, f2 and f3 represents the desired output values, NN represents these values obtained by the traditional ANN models, and RBNN represents the values of these parameters obtained from the proposed model with four nodes in the hidden layer. The performance of the proposed model is given in the Figure2. The pattern characteristics of the traditional ANN model and RBNN model for degree four is incorporated in Figures 3. The performance of the proposed model for degree four is given in the Figures 4.

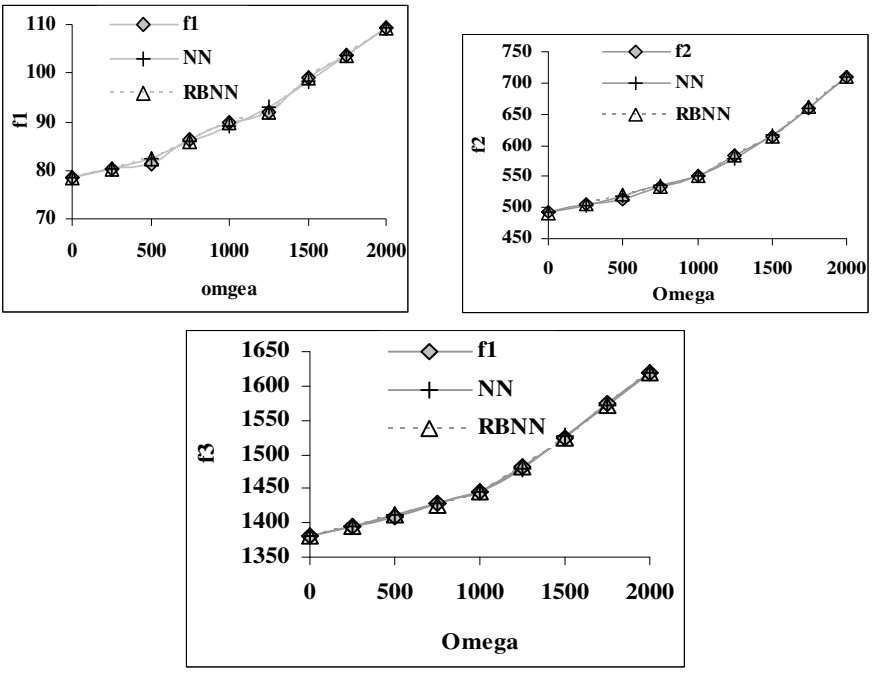


Fig. 2. Results of training of *RBNN* and *ANN* models for Omega Vs Frequency F1, F2 and F3 (D-3)

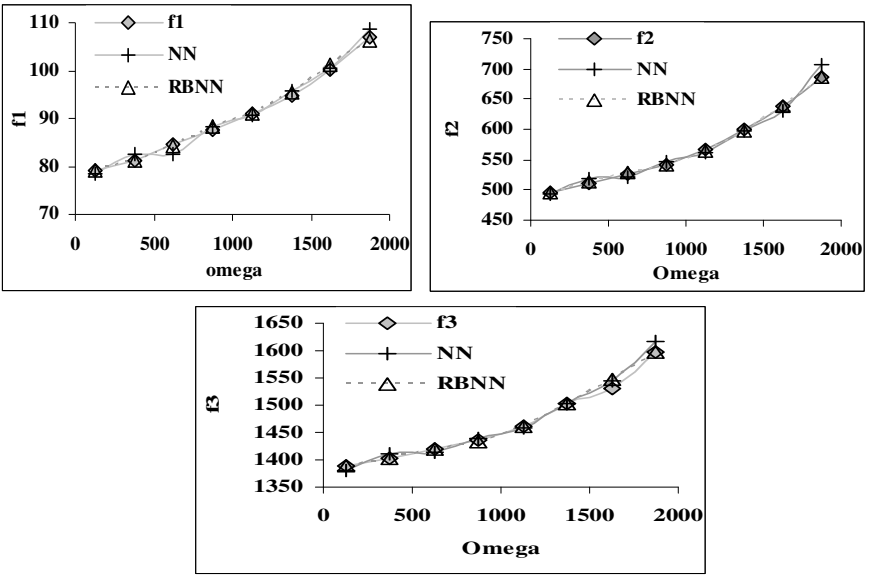


Fig. 3. Performance of the proposed Regression based Neural Network for Omega Vs Frequency F1, F2 and F3 (D-3)

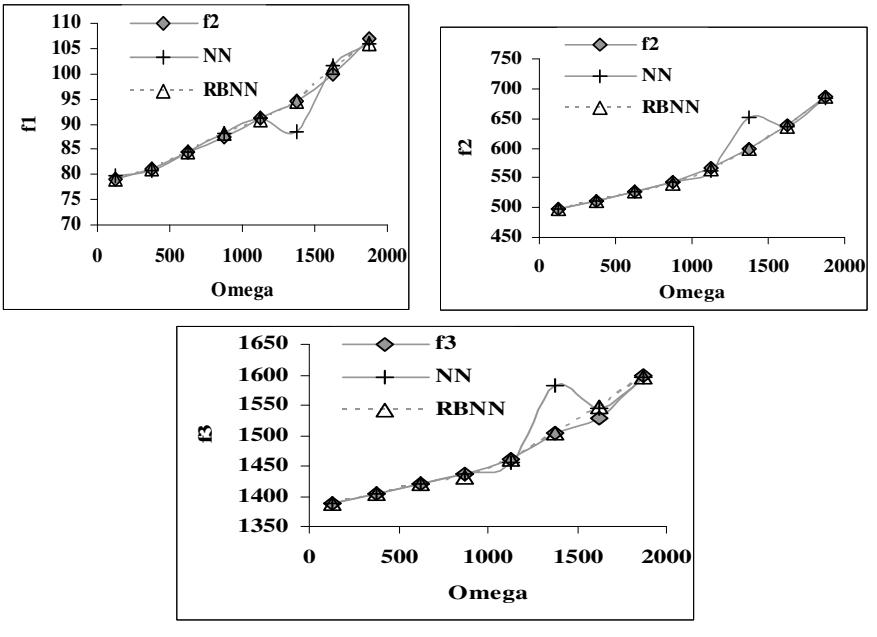


Fig. 4. Performance of the proposed Regression based Neural Network for Omega Vs Frequency F1, F2 and F3 (D-4)

5 Conclusions

The RBNN method employed to solve fourth order partial differential equation for rotor blade in this paper gives a direct estimation of frequencies without going through traditional and lengthy numerical solutions procedures. The numerical results, when plotted, show that with the increase in Omega (angular velocity), there is increase in lagging motion frequencies. The increase in the lower mode natural frequencies is smaller than that of the higher modes. This finding is in agreement with the results reported in earlier research ([3],[4],[5]) that have been carried out by employing Rayleigh-Ritz and FEM respectively. Furthermore, RBNN soft-computing method used in this research is useful to solve other beam, plates and shell vibration problems and guide engineers immensely in their structures design needs. Last of all, NN-methods in general [2, 6] have attracted extensive attention in recent past as NN-approaches have led to many efficient algorithms help in exploring the intrinsic structure of data set.

References

[1] Chakraverty, S., Singh, V.P., Sharma, R.K.: Regression-based weight generation algorithm in network for estimation of frequencies of vibrating plates. Computer Methods in Applied Mechanics and Engineering 195, 4194–4202 (2006)

[2] Kumar, S., Mittal, G.: Rapid detection of microorganism using image processing parameters and Neuro Network. Food Bio Process Technology 3, 741–751 (2010)

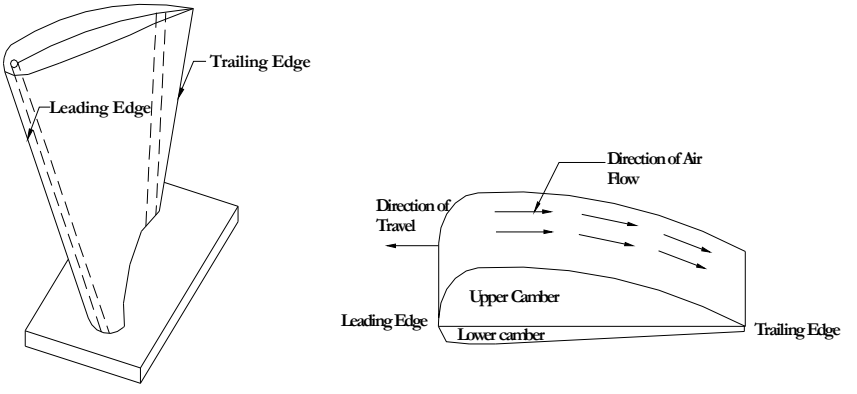
[3] Lü, J., Jiadao, W., Chan, D.: Transverse vibration of blade for unmanned micro helicopter using Rayleigh-Ritz method. University of Science and Technology Beijing 10(6), 40–43 (2003)

[4] Rao, R.S., Gupta, S.S.: Finite element vibration analysis of rotating Timoshenko beams. Sound and Vibration 242(1), 103 (2001)

[5] Sahu, A.: Theoretical Frequency Equation of Bending Vibrations of an Exponentially Tapered Beam Under Rotation. Vibration and Control 7(6), 775–780 (2001)

[6] Wang, J., Liao, X.-F., Yi, Z. (eds.): ISNN 2005. LNCS, vol. 3498. Springer, Heidelberg (2005) ISBN-10 3-540-25913-9

Appendix



Leading and Trailing Edges of a Blade

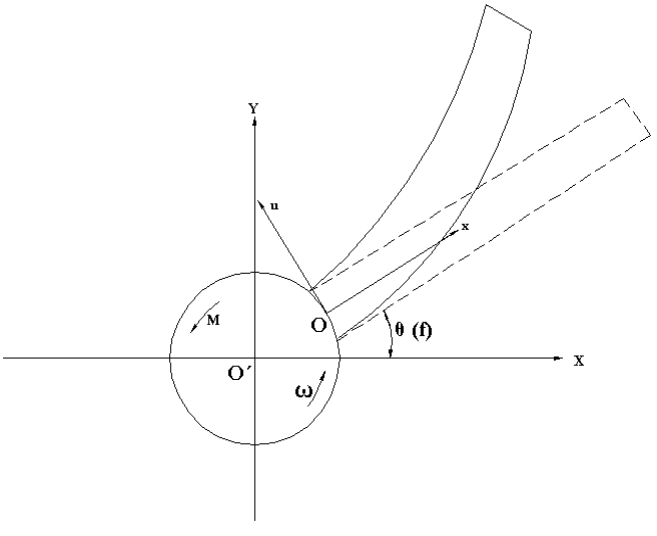


Fig. 5. Helicopter Blade Scheme

Support Vector Regression with Chaotic Hybrid Algorithm in Cyclic Electric Load Forecasting

Wei-Chiang Hong[1], Yucheng Dong[2], Li-Yueh Chen[3], B.K. Panigrahi[4], and Shih-Yung Wei[1]

[1] Department of Information Management, Oriental Institute of Technology, 58 Sec. 2, Sichuan Rd., Panchiao, Taipei, 220, Taiwan, R.O.C.
samuelsonhong@gmail.com
[2] Department of Organization and Management, Xi'an Jiaotong University, Xi'an, 710049, China
ycdong@mail.xjtu.edu.cn
[3] Department of Global Marketing and Logistics, MingDao University, 369 Wen-Hua Rd., Peetow, Changhua, 52345, Taiwan, R.O.C.
liyuchen@mdu.edu.tw
[4] Department of Electrical Engineering, Indian Institutes of Technology (IITs), India
bkpanigrahi@ee.iitd.ac.in

Abstract. Application of support vector regression (SVR) with chaotic sequence and evolutionary algorithms not only could improve forecasting accuracy performance, but also could effectively avoid converging prematurely. However, the tendency of electric load sometimes reveals cyclic changes due to seasonal economic activities or climate seasonal nature. The applications of SVR model to deal with cyclic electric load forecasting have not been widely explored. This investigation presents a SVR-based electric load forecasting model which applied a novel hybrid algorithm, namely chaotic genetic algorithm-simulated annealing algorithm (CGASA), to improve the forecasting performance. In addition, seasonal adjustment mechanism is also employed to deal with cyclic electric loading tendency. A numerical example from an existed reference is used to elucidate the forecasting performance of the proposed seasonal support vector regression with chaotic genetic algorithm, namely SSVRCGASA model. The forecasting results indicate that the proposed model yields more accurate forecasting results than ARIMA and TF-ε-SVR-SA models in existed papers. Therefore, the SSVRCGASA model is a promising alternative for electric load forecasting.

Keywords: Support vector regression (SVR), Chaotic genetic algorithm-simulated annealing (CGASA), Seasonal adjustment mechanism, Cyclic electric load forecasting.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 833–846.
springerlink.com

1 Introduction

The electric power industry is getting into its free competitive era, due to the rise of power market privatization and deregulation. Thus, for the market participants in a regional or a national system, accurate electric load forecasting has become the most important issue. However, the electric load forecasting is complicated and the electric load data presents nonlinear data patterns caused by the influencing factors, such as climate factors (temperature and humidity), social activities (human social activities) and seasonal factors (seasonal climate change and load growth). In the past decades, various approaches have been proposed, such as Box-Jenkins' ARIMA models [1-4], exponential smoothing models [5,6], multiplicative autoregressive (AR) model [7], Bayesian estimation model [8], and the state space and Kalman filtering technology [9,10]. The disadvantage of these methods is time consuming in the modeling process and is unable to avoid the observation noise in the forecasting process, particularly for the situation when the number of variables is increased.

The second approach is regression model, which is based on the cause-effect relationships between electric load and relevant independent variables (weather, holiday, temperature, wind conditions, humidity, and so on), such as linear regression, lots of explanatory variables, such as "weather" (relative humidity, heating, and cooling degree-days), "holiday", "temperature", and "economic and geographic" (GDP, electricity price, and population), are considered into the regression model [11,12]. These forecasting approaches are difficult to take some significant improvement in terms of forecasting accuracy due to their theoretical limitations, such as unable to capture the rapid variational process changes underlying of electric load from historical data pattern. In addition, these models are based on linear assumption, however, these independent variables are unjustified to be used because of the terms are known to be nonlinear. Knowledge based expert system (KBES) approach [13] extracts rules from received relevant information (e.g., daily temperature, day type, load from the previous day, and so on), then, derives training rules and transforms the information into mathematical equations. This approach has the rule-based mechanism to transform new rule from received information, i.e., the expert capability is trained and increased by the existence presuming to increase forecasting accuracy [14,15]. Recently, artificial neural network (ANN) has received wide successful applications in improving load forecasting accuracy [16-18]. These experimental results indicate that the ANN model is superior to ARIMA and regression models in terms of forecasting accuracy. ANN-based models seem to obtain improved and acceptable performance in load forecasting issue and to provide the possible nonlinear extensions to the ARIMA models in load time series. However, the training procedure of ANNs models is not only time consuming but also possible to get trapped in local minima and subjectively in selecting the model architecture [19].

Support vector regression (SVR) [20] has been successfully used to solve forecasting problems in many fields, such as financial time series (stocks index and exchange rate) forecasting [21-25], tourist arrival forecasting [26,27], atmospheric science forecasting [28-31], traffic flow forecasting [32-34], and so on. Meanwhile,

SVR model has also been successfully applied to forecast electric load [35-41]. The empirical results indicate that the selection of the three parameters *C* (to trade off the training errors and large weights), ε (the width of the insensitive loss function), and σ (the parameter of Gaussian kernel function) in a SVR model influences the forecasting accuracy significantly. Evolutionary algorithms are employed to determine appropriate parameter values [42]. Although, numerous publications in the literature had given some recommendations on appropriate setting of SVR parameters [43], however, those approaches do not simultaneously consider the interaction effects among the three parameters. Authors have conducted a series of relevant researches, by employed different evolutionary algorithms (such as genetic algorithms, simulated annealing algorithms, immune algorithms, and particle swarm optimization) for parameters determination, to identify that which algorithm is suited for specified data patterns. In which, all SVR with different evolutionary algorithms are superior to other competitive forecasting models (ARIMA, regression models, and ANNs etc.), however, these algorithms are almost lack of knowledge memory or storage functions which would lead to neither time consuming nor inefficiency in the searching the suitable parameters (i.e., being trapped in local optimum). Therefore, authors have also conducted several trials by employing hybrid chaotic sequence with evolutionary algorithms for parameters determination to overcome the immature convergence problem [26,28,35, 37]. In order to continue testing the stability and superiority of hybrid chaotic sequence with evolutionary algorithms, this investigation tries to employ the chaotic genetic algorithm-simulated annealing algorithm (CGASA) to determine the values of three parameters in a SVR model.

Genetic algorithm (GA) is auto-adaptive stochastic search techniques [44], it generates new individuals with selection, crossover and mutation operators. GA starts with a coding of the parameter set of all types of objective functions, thus, it has the ability in solving problems those traditional algorithms are not easily to solve. In Pai and Hong [27,40], GA is able to reserve a few best fitted members of the whole population, however, after some generations GA might lead to a premature convergence to a local optimum in the searching the suitable parameters of a SVR model. Simulated annealing (SA) is a stochastic based general search tool that mimics the annealing process of material physics [45]. Due to its new state acceptance mechanism, SA has some institution to be able to escape from local minima and reach global minimum [46]. In Pai and Hong [41], SA costs more computation time. To ensure the efficiency of SA, a proper temperature cooling rate (stop criterion) should be considered. Genetic algorithm-simulated annealing (GA-SA) hybrid algorithm is a novel trial in dealing with the challenges mentioned above. The GA-SA can firstly employ the superiority of SA algorithm to escape from local minima and approximate to the global minimum, and secondly apply the mutation process of GA to improve searching ability in the range of values. So, the hybrid algorithm has been applied to the fields of system design [47], system and network optimization [48,49], query to information retrieval system [50], continuous-time production planning [51,52], and electrical power districting problem [53]. However, there is little application of the GA-SA to SVR's parameter determination.

This investigation presented in this paper is motivated by a desire to improve the problem of maintaining the premature convergence to a local optimum of GA and the efficiency of SA mentioned above in determining the three free parameters in the SVR model. The chaotic genetic algorithm-simulated annealing algorithm (CGASA) is employed, namely SVRCGASA, to provide good forecasting performance in capturing non-linear electric load changes tendency. In the meanwhile, as indicated in existed literature [54-56] that electric load demands also reveals a cyclic trend caused by the difference in demand from month to month and season to season, the applications of SVR models to deal with cyclic trend time series, however, have not been widely explored. Therefore, this paper also attempts to apply the seasonal adjustment method [56,57] to deal with seasonal trend time series problem. The remainder of this paper is organized as follows. The SSVRCGASA model is introduced in section 2, including the formulation of SVR, the CGASA algorithm, and the seasonal adjustment process. A numerical example is presented in section 3. Conclusions are discussed in section 4.

2 Methodology of SVRCGASA Model

2.1 Support Vector Regression

The brief ideas of SVMs for the case of regression are introduced. A nonlinear mapping $\varphi(\cdot)$: $\Re^n \to \Re^{n_h}$ is defined to map the input data (training data set) $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ into a so-called high dimensional feature space (which may have infinite dimensions), $\Re^{n_h}$. Then, in the high dimensional feature space, there theoretically exists a linear function, f, to formulate the nonlinear relationship between input data and output data. Such a linear function, namely SVR function, is as Eq. (1),

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}) + b\,. \tag{1}$$

where $f(\mathbf{x})$ denotes the forecasting values; the coefficients $\mathbf{w}$ ($\mathbf{w} \in \Re^{n_h}$) and b ($b \in \Re$) are adjustable. As mentioned above, SVM method aims at minimizing the empirical risk,

$$R_{emp}(f) = \frac{1}{N}\sum_{i=1}^{N}\Theta_{\varepsilon}(y_i, \mathbf{w}^T\varphi(\mathbf{x}_i) + b)\,. \tag{2}$$

where $\Theta_{\varepsilon}(\mathbf{y}, f(\mathbf{x}))$ is the ε-insensitive loss function and defined as Eq. (3),

$$\Theta_{\varepsilon}(\mathbf{y}, f(\mathbf{x})) = \begin{cases} |f(\mathbf{x}) - \mathbf{y}| - \varepsilon, & if \quad |f(\mathbf{x}) - \mathbf{y}| \geq \varepsilon \\ 0, & otherwise \end{cases}. \tag{3}$$

In addition, $\Theta_\varepsilon(\mathbf{y}, f(\mathbf{x}))$ is employed to find out an optimum hyper plane on the high dimensional feature space to maximize the distance separating the training data into two subsets. Thus, the SVR focuses on finding the optimum hyper plane and minimizing the training error between the training data and the ε-insensitive loss function. Then, the SVR minimizes the overall errors,

$$\underset{\mathbf{w},b,\xi^*,\xi}{\text{Min}} \; R_\varepsilon(\mathbf{w},\xi^*,\xi) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}(\xi_i^* + \xi_i) . \tag{4}$$

with the constraints

$$\begin{aligned} \mathbf{y}_i - \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) - b \le \varepsilon + \xi_i^*, \quad i = 1,2,...,N \\ -\mathbf{y}_i + \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b \le \varepsilon + \xi_i, \; i = 1,2,...,N \\ \xi_i^* \ge 0, \; i = 1,2,...,N \\ \xi_i \ge 0, \; i = 1,2,...,N \end{aligned}$$

The first term of Eq. (4), employed the concept of maximizing the distance of two separated training data, is used to regularize weight sizes, to penalize large weights, and to maintain regression function flatness. The second term penalizes training errors of $f(\mathbf{x})$ and $\mathbf{y}$ by using the ε-insensitive loss function. C is a parameter to trade off these two terms. Training errors above ε are denoted as ξ_i^*, whereas training errors below $-\varepsilon$ are denoted as ξ_i.

After the quadratic optimization problem with inequality constraints is solved, the parameter vector $\mathbf{w}$ in Eq. (1) is obtained,

$$\mathbf{w} = \sum_{i=1}^{N}(\beta_i^* - \beta_i)\varphi(\mathbf{x}_i) . \tag{5}$$

where β_i^*, β_i are obtained by solving a quadratic program and are the Lagrangian multipliers. Finally, the SVR regression function is obtained as Eq. (6) in the dual space,

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\beta_i^* - \beta_i)K(\mathbf{x}_i,\mathbf{x}) + b . \tag{6}$$

where $K(\mathbf{x}_i,\mathbf{x}_j)$ is called the kernel function, and the value of the kernel equals the inner product of two vectors, $\mathbf{x}_i$ and $\mathbf{x}_j$, in the feature space $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$, respectively; that is, $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i) \circ \varphi(\mathbf{x}_j)$. Any function that meets Mercer's condition can be used as the kernel function. The Gaussian RBF kernel function is specified in this study.

It is well known that the forecasting accuracy of a SVR model depends on a good setting of hyper parameters C, ε, and the kernel parameters (σ). Thus, the

determination of all three parameter selection is further an important issue. There is no structural method or any shortage opinions on efficient setting of SVR parameters. In addition, aforementioned, GA is lack of knowledge memory functions, which leads to time consuming and a premature convergence to a local optimum in the searching the suitable parameters of a SVR model. Therefore, the CGA is used in the proposed SVR model to optimize the parameter selection.

2.2 Implementation Structure of CGA-SA Algorithm

The proposed CGA-SA algorithm consists of the CGA part and the SA part. CGA evaluates the initial population and operates on the population using three basic genetic operators to produce new population (best individual), then, for each generation of CGA, it will be delivered to SA for further processing. After finishing all the processes of SA, the modified individual will be sent back to CGA for the next generation. These computing iterations will be never stopped till the termination condition of the algorithm is reached. The proposed procedure of CGA-SA is illustrated as follows.

A. Implementation steps of CGA

Step 1: Generate initial population by chaotic mapping operator (CMO). The values of the three parameters in a SVR model in the *i*th iteration can be represented as $X_k^{(i)}, k = C, \sigma, \varepsilon$. Set $i = 0$, and employ Eq. (7) to map the three parameters among the intervals (Min_k, Max_k) into chaotic variable $x_k^{(i)}$ located in the interval (0, 1).

$$x_k^{(i)} = \frac{X_k^{(i)} - Min_k}{Max_k - Min_k}, \quad k = C, \sigma, \varepsilon . \tag{7}$$

Then, adopt Eq. (8) with $\mu = 4$ to compute the next iteration chaotic variable, $x_k^{(i+1)}$. Transform $x_k^{(i+1)}$ to obtain three parameters for the next iteration, $X_k^{(i+1)}$, by the following Eq. (9).

$$x_k^{(i+1)} = \mu x_k^{(i)}(1 - x_k^{(i)}); \quad x_k^{(i)} \in (0,1), \quad i = 0, 1, 2, ... \tag{8}$$

$$X_k^{(i+1)} = Min_k + x_k^{(i+1)}\left(Max_k - Min_k\right). \tag{9}$$

After this transformation, the three parameters, C, σ, and ε, are encoded into a binary format, and are represented by a chromosome composed of "genes" of binary numbers. Each chromosome has three genes, respectively representing three parameters.

Step 2: Evaluate fitness. Evaluate the fitness (forecasting errors) of each chromosome. In this work, a negative mean absolute percentage error (-MAPE) is used as the fitness function. The MAPE is calculated as Eq. (10),

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - f_i}{y_i}\right| \times 100\% \ . \tag{10}$$

where y_i and f_i represent the actual and forecast values, and N is the number of forecasting periods.

Step 3: Selection. Based on fitness functions, chromosomes with higher fitness values are more likely to yield offspring in the next generation. The roulette wheel selection principle is applied to choose chromosomes for reproduction.

Step 4: Crossover operations. In crossovers, chromosomes are paired randomly. The proposed scheme adopts the single-point-crossover principle. Segments of paired chromosomes between two determined break-points are swapped. Finally, the three crossover parameters are decoded into a decimal format.

Step 5: Annealing chaotic mutation. For the *i*th iteration (generation) crossover population ($\hat{X}_k^{(i)}, k = C, \sigma, \varepsilon$) of current solution space (Min_k, Max_k) are mapped to chaotic variable interval [0,1] to form the crossover chaotic variable space $\hat{x}_k^{(i)}, k = C, \sigma, \varepsilon$, as Eq. (11),

$$\hat{x}_k^{(i)} = \frac{\hat{X}_k^{(i)} - Min_k}{Max_k - Min_k}, k = \mathrm{C}, \sigma, \varepsilon, i = 1, 2, ..., q_{\max} \ . \tag{11}$$

where $q_{\max}$ is the maximum evolutional generation of the population. Then, the *i*th chaotic variable $x_k^{(i)}$ is summed up to $\hat{x}_k^{(i)}$ and the chaotic mutation variable are also mapped to interval [0, 1] as in Eq. (12),

$$\tilde{x}_k^{(i)} = \hat{x}_k^{(i)} + \delta x_k^{(i)} \ . \tag{12}$$

where δ is the annealing operation. Finally, the chaotic mutation variable obtained in interval [0, 1] is mapped to the solution interval (Min_k, Max_k) by definite probability of mutation (p_m), thus completing a mutative operation.

$$\tilde{X}_k^{(i)} = Min_k + \tilde{x}_k^{(i)}\left(Max_k - Min_k\right). \tag{13}$$

Step 6: Stop condition. If the number of generations is equal to a given scale, then the best chromosomes are presented as a solution; otherwise, go back to Step 2.

B. Implementation steps of SA

Step 1: Generate initial current state. Receive values of the three parameters from GAs. The values of forecasting error, MAPE, shown as Eq.(10), is defined as the system state (E). Here, the initial state (E0) is obtained.

Step 2: Provisional state. Make a random move to change the existing system state to a provisional state. Another set of three positive parameters are generated in this stage.

Step 3: Metropolis criterion tests. The following Metropolis criterion equation is employed to determine the acceptance or rejection of provisional state [58].

$$\begin{cases} \text{Accept the provisional state, } \textit{if } E(s_{new}) > E(s_{old}), \textit{ and } \text{p} < \text{P(accept } \text{s}_{\text{new}}), \ 0 \le p \le 1. \\ \text{Accept the provisional state, } \textit{if } E(s_{new}) \le E(s_{old}) \\ \text{Reject the provisional state, } \textit{otherwise} \end{cases} \tag{14}$$

where the p is a random number to determine the acceptance of the provisional state, $P(\text{accept } s_{new})$, the probability of accepting the new state, is given by the following probability function, $P(\text{accept } s_{new}) = \exp\left(-\frac{E(s_{old}) - E(s_{new})}{kT}\right)$ (T is the thermal equilibrium temperature, k represents the Boltzmann constant). If the provisional state is accepted, then set the provisional state as the current state.

Step 4: Incumbent solutions. If the provisional state is not accepted, then return to step 2. Furthermore, if the current state is not superior to the system state, then repeat steps 2 and 3 until the current state is superior to the system state, and set the current state as new system state. Previous studies [45] indicated that the maximum number of loops (N_{sa}) is 100d to avoid infinitely repeated loops, where d denotes the problem dimension. In this investigation, three parameters (σ, C, and ε) are used to determine the system states. Therefore, N_{sa} is state to 300.

Step 5: Temperature reduction. After the new system state is obtained, reduce the temperature. The new temperature reduction is obtained by the Eq.(15),

$$\text{New temperature} = (\text{Current temperature}) \times \rho . \tag{15}$$

where $0 < \rho < 1$.

The ρ is set at 0.9 in this study [59]. If the pre-determined temperature is reached, then stop the algorithm and the latest state is an approximate optimal solution. Otherwise, go to step 2.

2.3 Seasonal Adjustment

As mentioned that due to the difference in demand from month to month and season to season, electric energy demands also demonstrate a cyclic tendency, any model attempts to accomplish the goal of high accurate forecasting performance, it is necessary to estimate this seasonal component. There are several approaches to estimate the seasonal index of data series [56,60,61], including product-model type and non-product-model type. Based on the data series type consideration, this investigation employed Deo and Hurvich's [61] approach to compute the seasonal index, as shown in Eq. (16),

$$season_t = \frac{y_t}{f_t} = \frac{y_t}{\sum_{i=1}^{N} \left(\beta_i^* - \beta_i\right) K(\mathbf{x}_i, \mathbf{x}) + b} . \tag{16}$$

where t=j, 1+j, 2l+j,…, (m-1)l+j only for the same time point in each period. Then, the seasonal index (SI) for each time point j is computed as Eq. (17),

$$SI_j = \frac{1}{m}\left(season_j + season_{l+j} + \cdots + season_{(m-1)l+j}\right). \quad (17)$$

Eventually, the forecasting value of the SSVRCGASA is obtained by Eq. (18),

$$f_{N+k} = \left(\sum_{i=1}^{N}\left(\beta_i^* - \beta_i\right)K(\mathbf{x}_i, \mathbf{x}_{N+k}) + b\right) \times SI_k \,. \quad (18)$$

where k=j, 1+j, 2l+j,…, (m-1)l+j implies the time point in another period (for forecasting period).

3 Numerical Examples

3.1 Data Set

To be based on the same comparison conditions, this study uses historical monthly electric load data of Northeast China to compare the forecasting performance of the proposed SSVRCGASA model with those of ARIMA and TF-ε-SVR-SA models proposed by Wang et al. [56]. In addition, due to verification of performance of Jordan recurrent network, the SVRCGASA model (without seasonal adjustment mechanism) is also involved in comparison. Totally, there are 64 data (from January 2004 to April 2009) of Northeastern China monthly electric load. However, based on Wang et al. [56]'s support vectors computation, only 53 months data (from December 2004 to April 2009) are suggested. Thus, for the same comparison condition, the employed data are divided into three data sets, the training data set (32 months, December 2004 to July 2007), the validation data set (14 months, August 2007 to September 2008), and the testing data set (7 months, from October 2008 to April 2009).

3.2 Parameters Determination of the SVRCGASA Model

In the training stage, the rolling-based forecasting procedure is conducted, which dividing training data into two subsets, namely fed-in subset and fed-out subset, respectively. Meanwhile, the structural risk minimization principle is also employed to minimize the training error, then, obtain one-step ahead forecasting load. Repeat the rolling-based forecasting procedure till the total forecasting load is obtained, and, training error in this training stage is also obtained. While training errors improvement occurs, the three kernel parameters, σ, C, and ε of the SVRCGASA model adjusted by CGASA are employed to calculate the validation error. Then, the adjusted parameters with minimum validation error are selected as the most appropriate parameters. Finally, a one-step-ahead policy is employed to forecast electric load. Note that the testing data set is not used for modeling but for examining the accuracy of the forecasting model. The forecasting results and the suitable parameters for the different SVRCGASA models are illustrated in

Table 1, in which it is indicated that these three models all perform the best when 25 fed-in data are used.

For the monthly electric load in Northeastern China, each month has different electric demand pattern, the seasonal length is verified as 12 [56], thus, there are 12 points (months) in each electric load cyclic year. The seasonal indexes for each point (month) are calculated based on the 46 forecasting values of the SVRCGASA model in training (32 forecasting values) and validation (14 forecasting values) stages, as shown in Table 2. Table 3 shows the actual values and the forecast values obtained using various forecasting models, ARIMA(1,1,1), TF-ε-SVR-SA, SVRCGASA, and SSVRCGASA models. The MAPE values are calculated to compare fairly the proposed models with other alternative models. The proposed SSVRCGASA model has smaller MAPE values than ARIMA, TF-ε-SVR-SA, and SVRCGASA models to capture the electric load cyclic trends on monthly average basis. In Table 3, the seasonal adjustment further revises the forecasting results from the SVRCGASA model (MAPE=3.731%), based on the seasonal indexes (per month) obtained from training and validation stages, to achieve more acceptable forecasting accuracy (1.901%). By comparing SVRCGASA with SSVRCGASA models, it also indicates the significant superiority from seasonal adjustment mechanism, even it is a little time consuming, however, it is deserved to pay some attentions on those cyclic information while modeling.

Table 1. Parameters of the SVRCGASA model

Nos. of fed-in data	Parameters			MAPE of testing (%)
	σ	C	ε	
5	88.1610	457.7400	3.7395	4.9360
10	6.8773	112.5800	3.7047	3.8340
15	6.3793	142.8500	0.0320	4.1970
20	247.8200	3435.2000	2.4109	3.9020
25	**51.2080**	**5045.1000**	**21.6230**	**3.7310**

Table 2. The seasonal indexes for each time point (month) for the SVRCGASA model

Time point (Month)	Seasonal index	Time point (Month)	Seasonal index
January	1.0239	July	1.0775
February	0.9180	August	1.0742
March	1.0234	September	1.0189
April	0.9941	October	0.9906
May	1.0271	November	1.0438
June	1.0321	December	1.0694

Table 3. Forecasting results of ARIMA, TF-ε-SVR-SA, and SSVRCGASA models (unit: hundred million KW/hour)

Time point (Month)	Actual	ARIMA (1,1,1)	TF-ε-SVR-SA	SVRCGASA	SSVRCGASA
Oct. 2008	181.0700	192.9316	184.5035	177.3000	175.6385
Nov. 2008	180.5600	191.1270	190.3608	177.4428	185.2100
Dec. 2008	189.0300	189.9155	202.9795	177.5848	189.9070
Jan. 2009	182.0700	191.9947	195.7532	177.7263	181.9693
Feb. 2009	167.3500	189.9398	167.5795	177.8673	163.2805
Mar. 2009	189.3000	183.9876	185.9358	178.0078	182.1747
Apr. 2009	175.8400	189.3480	180.1648	178.6806	177.6289
MAPE (%)		6.0440	3.7990	3.7310	1.9010

4 Conclusion

For a developing country, like China, or an energy-limited economy, like Taiwan, accurate electric load forecasting is a quite important guide for effective implementations/actions of energy policies. In this paper, we hybridize several novel forecasting techniques, seasonal adjustment, chaotic sequence, genetic algorithm and simulated annealing algorithm, namely SSVRCGASA model, to examine its potentiality in forecasting monthly electric loads. Three other alternative models, ARIMA, TF-ε-SVR-SA, and SVRCGASA models are used to compare the forecasting performance. Experiment results indicate that the proposed SSVRCGASA model has significant superiority among other alternatives in terms of forecasting accuracy. Meanwhile, this research also indicates the hybrid of seasonal adjustment and chaotic sequence is useful to improve the forecasting accuracy. Furthermore, other advanced optimization algorithms for parameters selection and other hybrid mechanism still have their potentials to be employed for a SVR-base load forecasting model to satisfy the requirement of accurate electric load forecasting.

Acknowledgments. This research was conducted with the support of National Science Council, Taiwan, ROC (NSC 100-2628-H-161-001-MY4, NSC 100-2811-H-161-001) and National Natural Science Foundation of China (No. 71171160).

References

1. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1976)
2. Chen, J.F., Wang, W.M., Huang, C.M.: Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. Electric Power Systems Research 34, 187–196 (1995)
3. Vemuri, S., Hill, D., Balasubramanian, R.: Load forecasting using stochastic models. In: Proceeding of the 8th Power Industrial Computing Application Conference, pp. 31–37 (1973)
4. Wang, H., Schulz, N.N.: Using AMR data for load estimation for distribution system analysis. Electric Power Systems Research 76, 336–342 (2006)

5. Christianse, W.R.: Short term load forecasting using general exponential smoothing. IEEE Transactions on Power Apparatus and System PAS-90, 900–911 (1971)
6. Park, J.H., Park, Y.M., Lee, K.Y.: Composite modeling for adaptive short-term load forecasting. IEEE Transactions on Power Systems 6, 450–457 (1991)
7. Mbamalu, G.A.N., El-Hawary, M.E.: Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation. IEEE Transactions on Power Systems 8, 343–348 (1993)
8. Douglas, A.P., Breipohl, A.M., Lee, F.N., Adapa, R.: The impact of temperature forecast uncertainty on Bayesian load forecasting. IEEE Transactions on Power Systems 13, 1507–1513 (1998)
9. Trudnowski, D.J., McReynolds, W.L., Johnson, J.M.: Real-time very short-term load prediction for power-system automatic generation control. IEEE Transactions on Control Systems Technology 9, 254–260 (2001)
10. Park, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E., Damborg, M.J.: Electric load forecasting using an artificial neural network. IEEE Transactions on Power Systems 6, 442–449 (1991)
11. Mirasgedis, S., Safaridis, Y., Georgopoulou, E., Lalas, D.P., Moschovits, M., Karagiannis, F., Papakonstantinou, D.: Models for mid-term electricity demand forecasting incorporating weather influences. Energy 31, 208–227 (2006)
12. Mohamed, Z., Bodger, P.: Forecasting electricity consumption in New Zealand using economic and demographic variables. Energy 30, 1833–1843 (2005)
13. Rahman, S., Bhatnagar, R.: An expert system based algorithm for short-term load forecasting. IEEE Transactions on Power Systems 3, 392–399 (1988)
14. Chiu, C.C., Kao, L.J., Cook, D.F.: Combining a Neural Network with a rule-based expert system approach for short-term power load forecasting in Taiwan. Expert Systems with Applications 13, 299–305 (1997)
15. Rahman, S., Hazim, O.: A generalized knowledge-based short-term load- forecasting technique. IEEE Transaction on Power Systems 8, 508–514 (1993)
16. Abdel-Aal, R.E.: Short-term hourly load forecasting using abductive networks. IEEE Transactions on Power Systems 19, 164–173 (2004)
17. Darbellay, G.A., Slama, M.: Forecasting the short-term demand for electricity—do neural networks stand a better chance? International Journal of Forecasting 16, 71–83 (2000)
18. Hsu, C.C., Chen, C.Y.: Regional load forecasting in Taiwan—application of artificial neural networks. Energy Conversion and Management 44, 1941–1949 (2003)
19. Suykens, J.A.K.: Nonlinear modelling and support vector machines. In: Proceeding of IEEE Instrumentation and Measurement Technology Conference, pp. 287–294 (2001)
20. Vapnik, V., Golowich, S., Smola, A.: Support vector machine for function approximation, regression estimation, and signal processing. In: Advances in Neural Information Processing Systems, vol. 9, pp. 281–287 (1996)
21. Cao, L.: Support vector machines experts for time series forecasting. Neurocomputing 51, 321–339 (2003)
22. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Computers and Operations Research 32, 2513–2522 (2005)
23. Hung, W.M., Hong, W.C.: Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting. Control and Cybernetics 38, 863–891 (2009)
24. Pai, P.F., Lin, C.S.: A hybrid ARIMA and support vector machines model in stock price forecasting. Omega 33, 497–505 (2005)

25. Pai, P.F., Lin, C.S., Hong, W.C., Chen, C.T.: A hybrid support vector machine regression for exchange rate prediction. International Journal of Information and Management Sciences 17, 19–32 (2006)
26. Hong, W.C., Dong, Y., Chen, L.Y., Wei, S.Y.: SVR with hybrid chaotic genetic algorithms for tourism demand forecasting. Applied Soft Computing 11, 1881–1890 (2011)
27. Pai, P.F., Hong, W.C.: An improved neural network model in forecasting arrivals. Annals of Tourism Research 32, 1138–1141 (2005)
28. Hong, W.C.: Rainfall forecasting by technological machine learning models. Applied Mathematics and Computation 200, 41–57 (2008)
29. Hong, W.C., Pai, P.F.: Potential assessment of the support vector regression technique in rainfall forecasting. Water Resources Management 21, 495–513 (2007)
30. Pai, P.F., Hong, W.C.: A recurrent support vector regression model in rainfall forecasting. Hydrological Processes 21, 819–827 (2007)
31. Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, A.A.: Support vector machines for wind speed prediction. Renewable Energy 29, 939–947 (2004)
32. Hong, W.C., Dong, Y., Zheng, F., Wei, S.Y.: Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. Applied Mathematics and Computation 217, 6733–6747 (2011)
33. Hong, W.C., Dong, Y., Zheng, F., Lai, C.Y.: Forecasting urban traffic flow by SVR with continuous ACO. Applied Mathematical Modelling 35, 1282–1291 (2011)
34. Hong, W.C.: Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. Neurocomputing 74, 2096–2107 (2011)
35. Hong, W.C.: Hybrid evolutionary algorithms in a SVR-based electric load forecasting model. International Journal of Electrical Power and Energy Systems 31, 409–417 (2009)
36. Hong, W.C.: Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. Energy Conversion and Management 50, 105–117 (2009)
37. Hong, W.C.: Electric load forecasting by support vector model. Applied Mathematical Modelling 33, 2444–2454 (2009)
38. Hong, W.C.: Application of chaotic ant swarm optimization in electric load forecasting. Energy Policy 38, 5830–5839 (2010)
39. Hong, W.C.: Electric load forecasting by seasonal recurrent SVR with chaotic artificial bee colony algorithm. Energy 36, 5568–5578 (2011)
40. Pai, P.F., Hong, W.C.: Forecasting regional electric load based on recurrent support vector machines with genetic algorithms. Electric Power Systems Research 74, 417–425 (2005)
41. Pai, P.F., Hong, W.C.: Support vector machines with simulated annealing algorithms in electricity load forecasting. Energy Conversion and Management 46, 2669–2688 (2005)
42. Yang, X.-S., Deb, S., Fong, S.: Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications. In: Fong, S. (ed.) NDT 2011. Communications in Computer and Information Science, vol. 136, pp. 53–66. Springer, Heidelberg (2011)
43. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17, 113–126 (2004)
44. Holland, J.: Adaptation in Natural and Artificial System. University of Michigan Press, Ann Arbor (1975)

45. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
46. Lee, J., Johnson, G.E.: Optimal tolerance allotment using a genetic algorithm and truncated Monte Carlo simulation. Computer Aided Design 25, 601–611 (1983)
47. Shieh, H.J., Peralta, R.C.: Optimal in situ bioremediation design by hybrid genetic algorithm-simulated annealing. Journal of Water Resources Planning and Management 131, 67–78 (2005)
48. Ponnambalam, S.G., Reddy, M.M.: A GA-SA multi-objective hybrid search algorithm for integrating lot sizing and sequencing in flow-line scheduling. International Journal of Advanced Manufacturing Technology 21, 126–137 (2003)
49. Zhao, F., Zeng, X.: Simulated annealing—genetic algorithm for transit network optimization. Journal of Computing in Civil Engineering 20, 57–68 (2006)
50. Cordón, O., Moya, F., Zarco, C.: A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. Soft Computing 6, 308–319 (2002)
51. Ganesh, K., Punniyamoorthy, M.: Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing. International Journal of Advanced Manufacturing Technology 26, 148–154 (2005)
52. Wang, Z.G., Wong, Y.S., Rahman, M.: Optimisation of multi-pass milling using genetic algorithm and genetic simulated annealing. International Journal of Advanced Manufacturing Technology 24, 727–732 (2004)
53. Bergey, P.K., Ragsdale, C.T., Hoskote, M.: A simulated annealing genetic algorithm for the electrical power districting problem. Annals of Operations Research 121, 33–55 (2003)
54. Dagum, E.B.: Modelling, forecasting and seasonally adjusting economic time series with the X-11 ARIMA method. Journal of the Royal Statistical Society, Series D (The Statistician) 27, 203–216 (1978)
55. Kenny, P.B., Durbin, J.: Local trend estimation and seasonal adjustment of economic and social time series. Journal of the Royal Statistical Society, Series A (General) 145, 1–41 (1982)
56. Wang, J., Zhu, W., Zhang, W., Sun, D.: A trend fixed on firstly and seasonal adjustment model combined with the ε-SVR for short-term forecasting of electricity demand. Energy Policy 37, 4901–4909 (2009)
57. Xiao, Z., Ye, S.J., Zhong, B., Sun, C.X.: BP neural network with rough set for short term load forecasting. Expert Systems with Applications 36, 273–279 (2009)
58. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H.: Equations of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
59. Dekkers, A., Aarts, E.H.L.: Global optimization and simulated annealing. Mathematical Programming 50, 367–393 (1991)
60. Azadeh, A., Ghaderi, S.F.: Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors. Energy Conversion and Management 49, 2272–2278 (2008)
61. Deo, R., Hurvich, C.: Forecasting realized volatility using a long- memory stochastic volatility model: estimation, prediction and seasonal adjustment. Journal of Econometrics 131, 29–58 (2006)

A Comparative Study of Different Approaches of Noise Removal for Document Images

Brijmohan Singh[1], Mridula[3], Vivek Chand[1], Ankush Mittal[2], and D. Ghosh[1]

[1] Research Cell, College of Engineering Roorkee, Roorkee
{bmsingh1981,vknight078}@gmail.com, ghoshfec@iitr.ernet.in
[2] Graphic Era University, Dehradun-248002
dr.ankush.mittal@gmail.com
[3] IIT Roorkee, Roorkee
mridulaiitr@gmail.com

Abstract. There has been intensive research carried out in the field of OCR (Optical Character Recognition). Lots of work has been done and articles have been published. Noise is one of the important factors which have to be handled at the stage of preprocessing before applying other steps of OCR systems. Noise is undesirable signal because it obscures the subject of the image. This paper presents the comparative study of the five noise removal approaches: Weiner, Median, Wavelet, Contourlet, and Curvelet for document images. The different approaches of noise removal were compared visually and by employing Peak Signal to Noise Ratio (PSNR), F-measure and NRM evaluation measures.

Keywords: OCR, Noise, Curvelet, Wavelet, Contourlet, Weiner filter, Median filter.

1 Introduction

Noise removal is necessary step during pre-processing of the document images before applying OCR to it. The output of this step is clean or smoothed document images which can be used efficently as the input to OCR systems. Noise removal is one of the ongoing resarch area in the field of the textual image processing and OCR. The era of OCR research is four decades old. Lots of research work has been done and different methodologies has been developed for OCR systems. But still noise removal is challenging due to availability of different types of noises in real document images. Real life documents contain many defects such as holes, dots and spots due to several reasons which range from the acquisition source type to environmental conditions [1]. In the process of digitization of documents, scanner and camera are the important hardware to capture the document image. Noise is also found in documents printed on low grade paper, printed by old printers and photocopying machines. Restoration of morphological information presented on

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 847–854.
springerlink.com

the document image is the important step for enhancing performance of OCR systems while keeping the fine details of the image intact [2].

Most approaches of document smoothing are based on human noise detection and filter selection. This process of smoothing may also be done automatically. Filtering approaches cannot be applied blindly because different filters cannot remove all type of noise and it leads to inefficiency in filtering process. There may be a difficult situation if the type of noise being removed does not exist in the document image or does not match with the strength of the filter [3].

There exist several types of noise models such as Gaussian noise, Rayleigh noise, Gamma noise, Exponential noise, Salt and pepper noise, uniform noise, and sinusoidal noise [4]. So, it is important to identify noise type and accordingly select filtering approach. It is really difficult to characterize and classify noise type and filter selection automatically. Recognition of noise type, classification and understanding of filters strength is very important for suitable noise removal, enhancing efficiency and quality of document image [5]. But in real document images, it is very difficult to identify the noise type because the document may contain so many types of noises.

2 Literature Review

There are two approaches of image filtering to clean noise in document images; human classification based and blindly applies a batch of image filters. Few works are reported in [6-9] based on classification of the noise available in a document image for identifying a filter to use. Most of the works reported are based on the human classification in the identification of noise and filter selection. Classification of noise may also be based on the properties like shape, gray values, position and periodicity of occurrence in the document. Noise types such as ink blobs, complex background binarized patterns, marginal noise [10] and salt-n-pepper [11- 12] often lack any consistent property. This irregular noise has typically been classified with simple rule based features. If behaviour of noise is consistent in terms of these properties, it is called regular noise.

Some other works on noise removal are reported in [13- 21]. In [22-23] noise blocks with large size, which are unlikely to be text, are removed using heuristic rules. Chinnasarn et al. [24] proposed an algorithm based on activity detection is proposed to remove salt and pepper noise from document images that contain graphics, characters, and dithered areas. Cheriet [25] proposed a multi-scale approach to extract the full shape of handwritten data from noisy gray level images, in order to overcome the intensity variation of strokes over a broad range of spatial scales. The method presented by Don [26] proposed a method based on noise spot model with Gaussian spatial distributions and Gaussian gray value distributions. Nishiwaki et al. [27] proposed a method of robust frame extraction and take care of noisy document.

3 Methods Employed for Noise Removal

In the off-line OCR, handwritten as well as printed document image to be recognized is captured by a sensor, for example, a scanner or a camera. Pre-processing is always a necessity whenever the document to be processed is noisy, inconsistent or incomplete. Pre-processing significantly improves the effectiveness of document analysis techniques. This paper attempts to presents a comparative study for noise removal from the grey document images. We applied the following methods on the sample data taken:

3.1 Median Filter

Median filter [28] is a well known approach that can remove salt and pepper noise from images but it fails in preserving corners and thin lines in a document image. A median filter is a non linear spatial filter.

3.2 Weiner Filter

Wiener filter [28] is an adaptive filter and often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high-frequency parts of an image. The wiener filter exploits the statistical properties of the image and can be used to restore image in the presence of blur as well as noise.

3.3 Contourlet Transform

The human visual system is so trained that it requires least number of visual active cells to capture the essential information of a natural scene. Also the receptive fields in the visual cortex are characterized as being localized, oriented and band pass [29]. Considering this we can say that the image representation should fulfill following conditions:

1. **Multi-resolution:** The representation should allow images to be successively approximated, from coarse to fine resolutions.
2. **Localization:** The basis elements in the representation should be localized in both the spatial and the frequency domains.
3. **Critical sampling:** For some applications (e.g., compression), the representation should form a basis, or a frame with small redundancy.
4. **Directionality:** The representation should contain basis elements oriented at a variety of directions, much more than the few directions that are offered by separable wavelets.
5. **Anisotropy:** To capture smooth contours in images, the representation should contain basis elements using a variety of elongated shapes with different aspect ratios.

The first three conditions are fulfilled by separable wavelets, while the last two requires a new construction, which gave birth to the concept of contourlet [30].

Natural images do not simply consist of smooth scan lines rather they consist of discontinuous curves which are located along the smooth contours, hence contourlet. Thus, a contourlet is a flexible multi-resolution, local and directional image representation using contour segments. The contourlets have elongated supports at various scales, directions and aspect ratios [29]. The above properties allow contourlet to efficiently represent an image with smooth curves at multiple resolutions.

3.4 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is the sampled version of Continuous Wavelet Transform (CWT). The Discrete Wavelet Transform (DWT) is based on sub-band coding and is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required. The DWT requires less space utilizing the space-saving coding based on the fact that wavelet families are orthogonal or bi-orthogonal bases, and thus do not produce redundant analysis [31]. In CWT, the signals are analyzed using a set of basis functions which relate to each other by scaling and translation. In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques.

3.5 Curvelet Transform

The Curvelet frame preserves the important properties, such as parabolic scaling, tightness and sparse representation for surface-like singularities of co-dimension one. Since many of the characters not only consist of edge discontinuities but also of curve discontinuities, the most widely used Wavelet transform works well with edge discontinuities but a curve discontinuity affects all the Wavelet coefficients . On the other hand, the curve discontinuities in any character are well handled with Curvelet transform with very few numbers of coefficients. Candes and Donoho [32-33] showed that curvelets are better than wavelets at representing edges and de-noising.

4 Evaluation Measures

The denoising approaches are evalated using three well known measures: PSNR, NRM and F-measure. The first evaluation measure, PSNR (Peak signal-to-noise ratio): the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The second measure, F-measure: the weighted harmonic mean of precision and recall. The third evaluation measure, NRM (Negative Rate Metric): based on pixel-wise mismatches between the ground truth and observations in a frame.

5 Results and Discussions

For this experiment, a dataset of 20 handwritten as well as printed English documents were collected. We collected two types of documents: noise free and noised

scanned documents. In noise free document images, we introduced Gaussian noise so that we can compare our results with the noise less images by calculating different evaluation approach such as PSNR, NRM, F- measure and determined the efficiency or effectiveness of approaches in context to noise removal. The results of denoising approaches are shown in fig.1. Fig. 2 shows the performance of different approaches of noise removal using PSNR evaluation measure. We have also compared denoising techniques on different variance of Gaussian noise based on F-measure and NRM evaluation measures as shown in table 1.

Fig. 1. Results of denoising approaches on grey document images

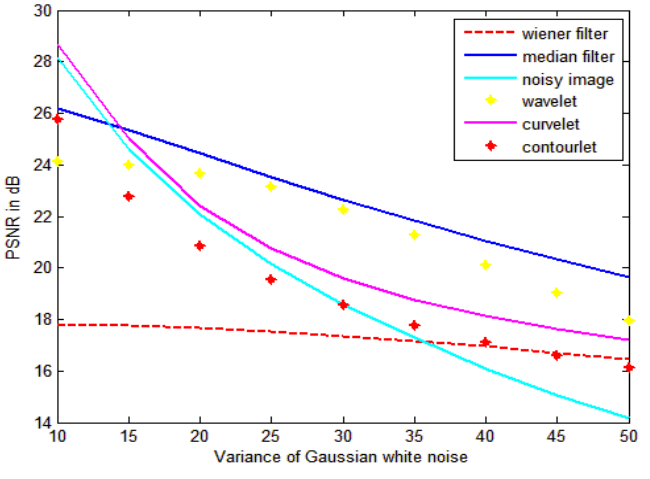


Fig. 2. Graph of PSNR of all above stated de-noising techniques

Table 1. Evaluation of denoising techniques for different variance of Gaussian noise

	Results when N=5		Results when N=10		Results when N=15	
De-noising Techniques	F-measure	NRM	F-measure	NRM	F-measure	NRM
Contourlet	96.79	1.75	94.81	4.00237	94.81	4.00237
Curvelet	96.79	1.75	80.0502	8.87643	94.81	4.00237
Median	80.1853	8.9177	97.1524	1.28143	80.0502	8.87643
Noisy image	98.11	0.7561	94.8619	2.88676	97.1524	1.28143
Wavelet	94.95	2.811	96.0826	2.75101	94.8619	2.88676
Weiner	96.86	2.33	94.81	4.00237	96.0826	2.75101

6 Conclusions

Printing, photocopying, and scanning processes degrade the image quality of any documents. In this research work, we presented a comparitive study of Weiner, Median, Wavelet, Contourlet, and Curvelet denoising techniques based on three evaluation measures: PSNR, F-measure, and NRM. On the basis of results, Curvelet, Wavelet and Contourlet preserves morphological structre of characters but not effective in the denoiseing of document images. This study is useful in the designing and development of new denoising algorithms for denoising.

References

[1] Gatos, B., Mantzaris, S.L., Perantonis, S.J., Tsigris, A.: Automatic page analysis for the creation of a digital library from newspaper archives. Int. J. Digit. Libr. 3, 77–84 (2000)

[2] Peerawit, W., Kawtrakul, A.: Marginal noise removal from document images using edge density. In: Proceeding of 4th Information and Computer Engineering Postgraduate Workshop, Phuket, Thailand (January 2004)

[3] Ye, X., Cheriet, M., Suen, C.Y.: A generic method of cleaning and enhancing handwritten data from business forms. Int. J. Doc. Anal. Recog. 4, 84–96 (2001)

[4] Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989)

[5] Kavallieratou, E., Stamatatos, E.: Improving the quality of degraded document images. In: Proceedings of the Second International Conference on Document Image Analysis for Libraries, pp. 330–339. IEEE (2006)

[6] Cao, H., Govindaraju, V.: Handwritten carbon form pre-processing based on markov random field. In: Proceeding of Computer Vision and Pattern Recognition, pp. 1–7. IEEE (2007)

[7] Lins, R.D., Silva, G.F.P., Simske, S.J., Fan, J., Shaw, M., Sá, P., Thielo, M.: Image classification to improve printing quality of mixed type documents. In: Proceeding of International Conference on Document Analysis and Recognition, pp. 1106–1110. IEEE Press, Barcelona (2009)

[8] Lins, R.D.: A Taxonomy for Noise in Images of Paper Documents - the Physical Noises. In: Kamel, M., Campilho, A. (eds.) ICIAR 2009. LNCS, vol. 5627, pp. 844–854. Springer, Heidelberg (2009)
[9] Lins, R.D., Banerjee, S., Thielo, M.: Automatically detecting and classifying noises in document images. In: Proceeding of ACM Symposium on Applied Computing, vol. 3, pp. 33–39 (2010)
[10] Fan, K.C., Wang, Y.K., Lay, T.R.: Marginal noise removal of document images. Pattern Recognition 35(11), 2593–2611 (2002)
[11] Zheng, Y., Liu, C., Ding, X., Pan, S.: Form frame line detection with directional single-connected chain. In: Proceeding of Sixth International Conference on Document Analysis and Recognition, pp. 699–703 (2001)
[12] Ali, M.: Background noise detection and cleaning in document images. In: Proceeding of 13th International Conference on Pattern Recognition, vol. 3, pp. 758–762 (1996)
[13] Bernsen, J.: Dynamic thresholding of grey-level images. In: Proceeding of 8th International Conference on Pattern Recognition, pp. 1251–1255 (1986)
[14] Niblack, W.: An Introduction to Digital Image Processing, pp. 115–116. Prentice Hall (1986)
[15] Schilling, R.J.: Fundamentals of Robotics Analysis and Control. Prentice-Hall, Englewood Cliffs (1990)
[16] O'Gorman, L.: Image and document processing techniques for the right pages electronic library system. In: Proceeding of 11th International Conference on Pattern Recognition, pp. 260–263 (1992)
[17] Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
[18] Story, G.A., O'Gorman, L., Fox, D., Schaper, L.L., Jagadish, H.V.: The right pages image-based electronic library for alerting and browsing. Computer 25(9), 17–26 (1992)
[19] Ali, M.B.J.: Background noise detection and cleaning in document images. In: Proceeding of International Conference on Pattern Recognition, Vienna, Austria, pp. 758–762 (1996)
[20] Liang, J., Haralick, R.: Document image restoration using binary morphological filters. In: Proceeding of SPIE Document Recognition III, San Jose, CA, vol. 2660, pp. 274–285 (1996)
[21] Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms. SIAM-Multiscale Modeling and Simulation 4, 490–530 (2005)
[22] Loce, R.P., Dougherty, E.R.: Enhancement and restoration of digital documents – Statistical Design of Nonlinear Algorithms. SPIE Optical Engineering Press (1997)
[23] Jain, A.K., Yu, B.: Document representation and its application to page decomposition. IEEE Transaction on Pattern Analysis and Machine Intelligence 20(3), 294–308 (1998)
[24] Chinnasarn, K., Rangsanseri, Y., Thitimajshima, P.: Removing salt-and-pepper noise in text/graphics images. In: Proceeding of IEEE Asia-Pacific Conference on Circuits and Systems, Chiangmai, pp. 459–462 (1998)
[25] Cheriet, M.: Extraction of handwritten data from noisy gray-level images using a multi-scale approach. In: Proceeding of Vision Interface, Vancouver, BC, Canada, vol. 1, pp. 389–396 (1998)

[26] Don, H.S.: A noise attributes thresholding method for document image binarization. International Journal on Document Image Analysis and Recognition 4(2), 131–138 (2000)
[27] Nishiwaki, D., Hayashi, M., Sato, A.: Robust Frame Extraction and Removal for Processing form Documents. In: Blostein, D., Kwon, Y.-B. (eds.) GREC 2001. LNCS, vol. 2390, pp. 36–45. Springer, Heidelberg (2002)
[28] Gonzalez, R.C., Woods, R.E.: Digital image processing (DIP/3e), 3rd edn. Pearson Education, Asia
[29] Siyuan, C., Xiangpeng, C.: The Second-generation Wavelet Transform and its Application in denoising of Seismic Data. Applied Geophysics 2(2), 70–74 (2005)
[30] Do, M.N., Vetterli, M.: The contourlet transform: An Efficient Directional Multiresolution Image Representation. IEEE Transactions on Image Processing 14, 2091–2106
[31] Hostalkova, E., Prochazka, A.: Wavelet Signal and Image Denoising. Institute of Chemical Technology. Department of Computing and Control Engineering
[32] Candès, E.J., Donoho, D.L.: Curvelets- A Surprisingly Adaptive Representation for Object with Edges, pp. 105–120. Vanderbilt University Press, Nashville (2000)
[33] Starck, J.L., Candès, E.J., Donoho, D.L.: The curvelet transform for image Denoising. IEEE Transactions on Image Processing 11(6), 670–684 (2002)

Improving Wireless Local Area Networks Performance Using Particle Swarm Optimization

Leena Arya[1], S.C. Sharma[1], and Millie Pant[2]

[1] Electronics & Computer Engg. Discipline, DPT, IIT Roorkee
[2] Mathematics Discipline, DPT, IIT Roorkee
{leenadpt,scs60fpt,millifpt}@iitr.ernet.in

Abstract. WLAN hold the promise of increasing employee productivity by providing them ubiquitous connectivity and mobility across enterprise. It is a flexible data communication system implemented as an extension to or as an alternative for, a wired LAN [1]. Prediction of the signal strength for indoor propagation environments can be faced with the analysis methods that include many obstacles and the effects of multipath propagation such as signal attenuation, reflection and diffraction. The placement of access points (AP) can be modelled as a nonlinear optimization problem. In the present study, an application of particle swarm optimization (PSO) is shown to determine the optimal placement of AP.

Keywords: Wireless LAN, access point, path loss model, particle swarm optimization.

1 Introduction

Wireless LANs enable users to access network resources and applications securely anytime and anywhere a wireless network is deployed [2]. Access points can nowadays be found in our daily environment, e.g. in many office buildings, public spaces and in urban areas [1]. A wireless LAN is a data transmission system designed to provide location-independent network access between computing devices by using radio waves rather than a wired infrastructure. In the corporate enterprise, wireless LANs are usually employed as the final link between the existing wired network and a group of client computers, giving these users wireless access to the full resources and services of the corporate network across a building or campus [3]. WLAN networks have become very popular means for providing a wireless networking facility for home users, educational institutions, companies etc. due to their ease of installation and their high data rate provision, apart from providing, although limited, mobility to users. [2].

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 855–865.
springerlink.com

If the APs are placed too far apart, they will generate a coverage gap, but if they are too close to each other, this will lead to excessive co-channel interferences and increases the cost unnecessary [5]. In this paper, we use PSO to determine location in such a WLAN. In the indoor environment the propagated electromagnetic signal can undergo necessary three mechanisms of electro-magnetic wave propagation- reflection, diffraction and scattering.

The basic structure of a WLAN is called a Basic Service Set (BSS) which comes in two categories: Infrastructure BSSs and Independent BSSs. In infrastructure mode, the wireless network consists of at least one access point connected to the wired network infrastructure and a set of wireless end stations. This configuration is called a Basic Service Set (BSS) [3]. An Extended Service Set (ESS) is a set of two or more BSSs forming a single subnet work as shown in fig.1. In an independent BSSs (IBSS) stations communicate directly with each other and are usually composed of a small number of stations set up for a short period of time[3]. IBSSs are often referred to as ad hoc networks as shown in fig.2.

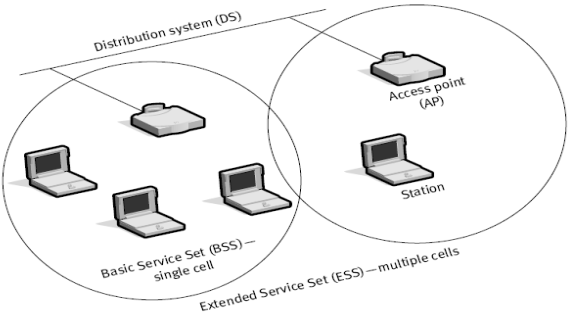


Fig. 1. Infrastructure Basic Service Set

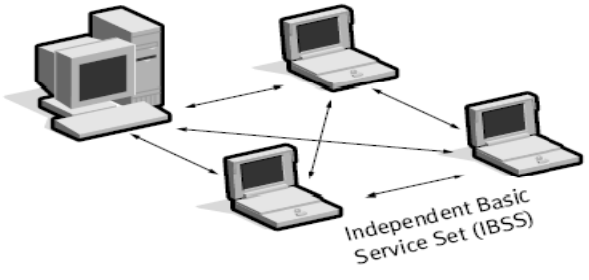


Fig. 2. Independent Basic Service Set

Mathematically, the optimal placement of AP can modelled as a nonlinear optimization problem, for which a suitable method is needed to obtain the solution. In the past few years, much attention is laid on to the nature inspired

optimization algorithms which can easily deal with nonlinear/ non-convex type functions. The focus of the present study is on particle swarm optimization (PSO), which has emerged as a powerful global optimization tool. The concept of PSO was first suggested by Kennedy and Eberhart (1995). It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution [4].

The rest of the paper is organized as follows: Notations are given in section 2. Section 3 provides the mathematical model description and path loss model. Section 4 shows the solution of the model and working of PSO. Section 5 describes the method of testing, setup, parameter settings of PSO and methodology. The measurement results are presented in Section 6. Finally, section 7 concludes the paper.

2 General Notations

Throughout this paper the following notations are used:

a_j	$j = 1 \ldots N$	Access point (AP)
r_i	$i = 1 \ldots M$	Receiver/user
$d(a_j, r_i)$		Distance between AP and receiver
$g(a_j, r_i)$		Path loss from i_{th} user to access point j
g max		Maximum tolerable path loss
P_t		Transmit power
P_r		Received power
R_{th}		Receive threshold
Ap		Position of AP

It should be noted that a_j represents the unknown coordinates of APs. Their number N is not known either. The coordinates of users r_i are assumed to be known and these users can be distributed in design area according to the design specifications.

In the present analysis the distance function assumed to be Euclidean, hence on the plane, the distance (d) between an AP a_j and a receiver r_i is given by [5]:

$$d(a_j, r_i) = \sqrt{(r_i^1 - a_j^1)^2 + (r_i^2 - a_j^2)^2}$$

where $a_j = a_j(a_j^1, a_j^2)$, and $r_i = r_i(r_i^1, r_i^2)$

3 Model Description

The aforementioned problem can be modelled as an optimization problem for which the objective function is to minimize the path loss. Mathematically it may be given as:

$$\min g(a_j, r_i) \leq g\max \forall i = 1,...,M \tag{1}$$

Constraint (1) states that path loss is evaluated against the maximum tolerable path loss $g\max$. This ensures that the quality of coverage at each receiver location is above the given threshold. This given value, $g\max$ can be calculated by subtracting receiver threshold (R_{th}) from transmitter power (P_t).

$$g\max = P_t - R_{th} \tag{2}$$

The above inequality (1) can be expressed in the equality form as:

$$(\min_j g(a_j, r_i) - g\max)^+ = 0, \tag{3}$$

$$\text{Where } (\alpha)^+ = \max(\alpha, 0)$$

3.1 Path Loss Model

The propagation of radio waves is characterized by several factors: (a) free space loss. (b) Attenuated by the objects on the propagation path such as windows, walls, table, chair and floors of building. (c) The signal is scattered and can interfere with itself [7].The basic propagation model is based on free space propagation. In general the power received by an antenna that is separated from the transmitting antenna by the distance d in free space is given by [5-6]:

$$\mathrm{P_r}(a_j, r_i) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d(a_j, r_i)^2} \tag{4}$$

where P_t is the transmitted power, G_t and G_r are the transmitter and receiver antenna gain, d is the distance between transmitter and receiver, and $\lambda = c/f$ is the wavelength of the carrier frequency, c is the speed of light (3×10^8 meter per second) and f is the frequency of radio carrier in hertz. The path loss, which represents signal attenuation between the transmitted and the received power and is measured in dB (decibels), in free space environments, is given by [5-6]:

$$g(a_j, r_i)[dB] = -10\log\left[\frac{G_t G r \lambda^2}{(4\pi)^2 d(a_j, r_i)^2}\right]$$

The above equation does not hold when points a_j and r_i are very close to each other. Therefore, large scale propagation models use a close–in distance, d_0 which is known as the received power reference distance point. Therefore, path losses at reference distance assuming transmit and receive antenna with unity gain as described in [5-6] can be calculated from:

$$g(a_j, r_i) = g(d_o)[dB] = 20\log\frac{4\pi d_o f}{c} \tag{5}$$

Therefore, path loss function in free space at a distance greater than d_0 is given by

$$g(a_j, r_i)[dB] = g(d_o)[dB] + 10\log\left(\frac{d(a_j, r_i)}{d_o}\right)^2 \tag{6}$$

4 Solution of the Model

The standard powerful optimization techniques (Newton based, quasi-Newton methods, conjugate gradient search method, steepest descent method are suitable for continuous functions only and cannot be applied to the given problem. The objective function is nonsmooth and discontinuous due to the tiny change in the position of users or APs that can happen. Since its development is 1995, PSO has become one of the most promising optimizing techniques for solving global optimization problems. Its mechanism is inspired by the social and cooperative behaviour displayed by various species like birds, fish etc including human beings.

4.1 Working of PSO

For a D-dimensional search space the position of the i[th] particle is represented as $X_i = (x_{i1}, x_{i2}, ..., x_{id}, ..., x_{iD})$. Each particle maintains a memory of its previous best position $P_i = (p_{i1}, p_{i2}, ..., p_{id}, ..., p_{iD})$. The best one among all the particles in the population is represented as $P_g = (p_{g1}, p_{g2}, ..., p_{gd}, ..., p_{gD})$. The velocity of each particle is represented as $V_i = (v_{i1}, v_{i2}, ..., v_{id}, ..., v_{iD})$, is clamped to a maximum velocity $V_{\max} = (v_{\max,1}, v_{\max,2}, ..., v_{\max,d}, ..., v_{\max,D})$ which is specified by the user. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a number of times or until a minimum error is achieved. The two basic

equations which govern the working of PSO are that of velocity vector and position vector given by:

$$v_{id} = \omega * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (7)$$

$$x_{id} = x_{id} + v_{id} \quad (8)$$

Here c_1 and c_2 are acceleration constants. They represent the weighting of the stochastic acceleration terms that pull each particle toward personal best and global best positions. Therefore, adjustment of these constants changes the amount of tension in the system. Low of these constants allow particles to roam far from the target regions before tugged back, while high values result in abrupt movement toward, or past, target regions. The constants r_1, r_2 are the uniformly generated random numbers in the range of [0, 1]. Inertia weight, ω is employed to control the impact of the previous history of velocities on the current velocity, thereby influencing the trade-off between global and local exploration abilities of the particles. It can be a positive constant or even a positive linear or nonlinear function of time.

5 Method of Testing

5.1 Setup for AP

We performed our experiment in first case without an obstacle in the design area has dimensions of 64m x 60m and have 400 users. It has 100 rooms and part of corridors. The layout of the floor is shown in Fig 3. The entire wing of the first floor is covered by 10 access points installed at the locations indicated by red symbols in fig.3. Ten locations of measurement are chosen on the first floor of the Malviya Bhavan building as shown in Fig.3.denoted as A,B,C,D,E,F,G,H,I and J. The specification of the model of Access point is LINK (DWL-3200AP) and IEEE 802.11b standard are used to test the model. We conducted our experiments at the first and second floor of the Malviya Bhavan, IIT Roorkee, Saharanpur Campus. Our data collection system comprised of a laptop, running Windows 2007, MSA 338 handheld spectrum analyser, M304 omnidirectional dipole antenna.

5.2 Parameter Settings for PSO

PSO has few control parameters which are to be carefully fine tuned to get the appropriate results. In the present study, we have considered the following parameters; population size is kept 20. Acceleration constants c1 and c2 are fixed as 2.0. The inertia *w* is taken to be linearly decreasing (0.9 – 0.4)

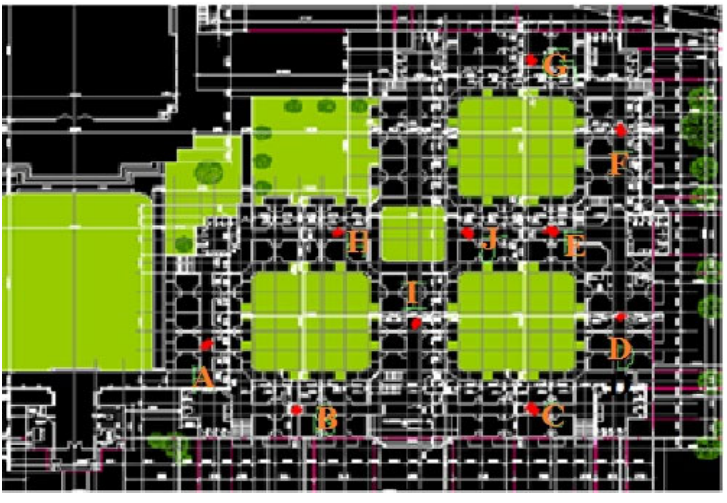


Fig. 3. Plan of the floor where the experiment was conducted. Readings were collected in the corridors.

5.3 Methodology

Once the priority areas have been identified, look for places nearby where it will be easy to install an access point. Access point will require a connection to the wired LAN and also a source of power. The signal strength has been measured using 3.3GHz Spectrum analyzer with Omni directional dipole antennas has been chosen, at a number of points around the access point. The coverage has been checked in those priority areas that are within range. While in other places the aim was to identify the points where the available bandwidth is likely to drop below the theoretical maximum: typically where the signal strength falls below -70dBm [8].

6 Results

The received signal strength calculated using the spectrum analyser is shown in fig.4. and fig.5.The optimal placement of access points for 100 users on first floor with transmitter power of 21dBm is shown in Table 1. The actual placement of access point coordinates is shown in fig.6and the distribution of access point coordinates by using particle swarm optimization is shown in fig.7.

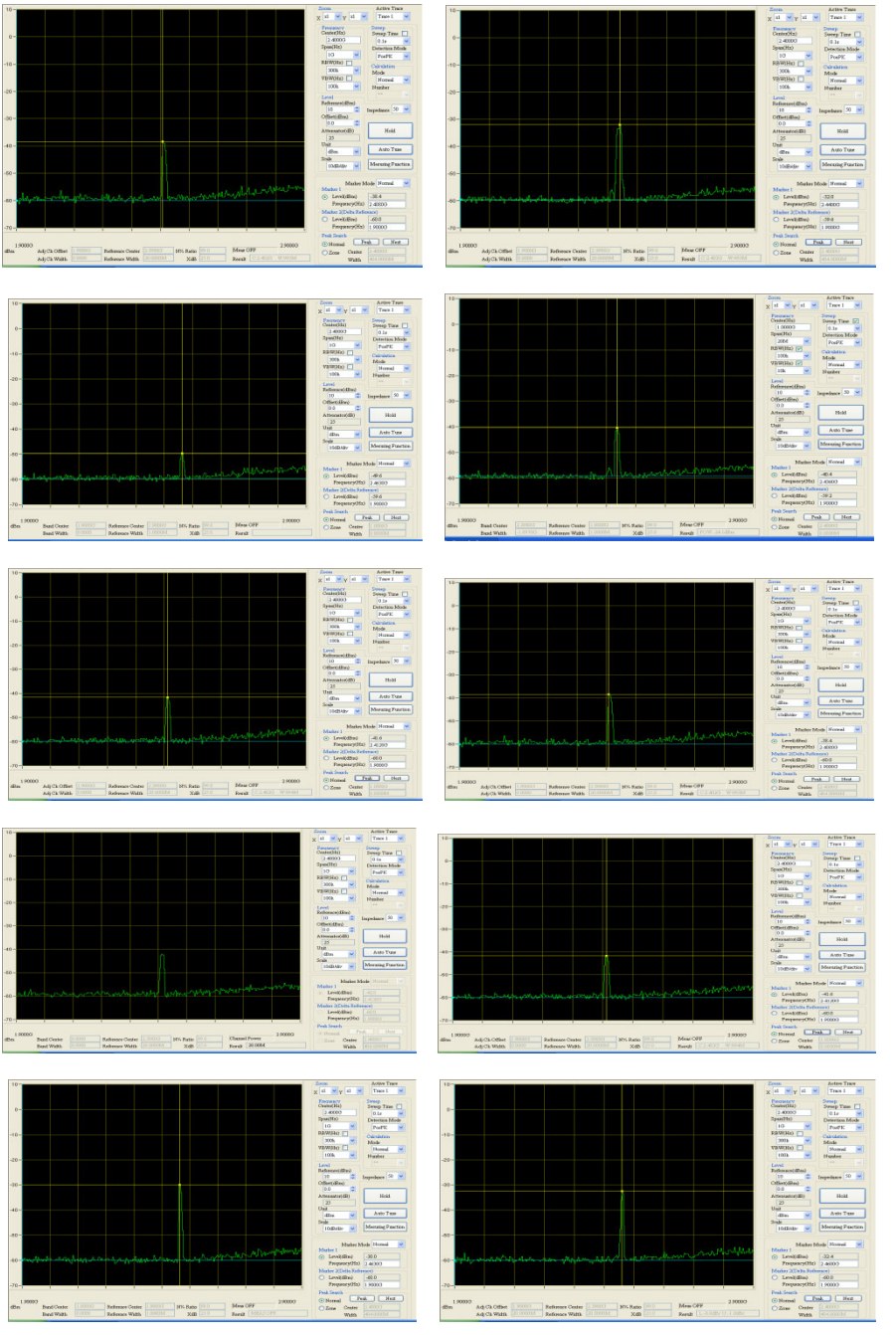


Fig. 4. Received Signal Strength calculated using spectrum analyser

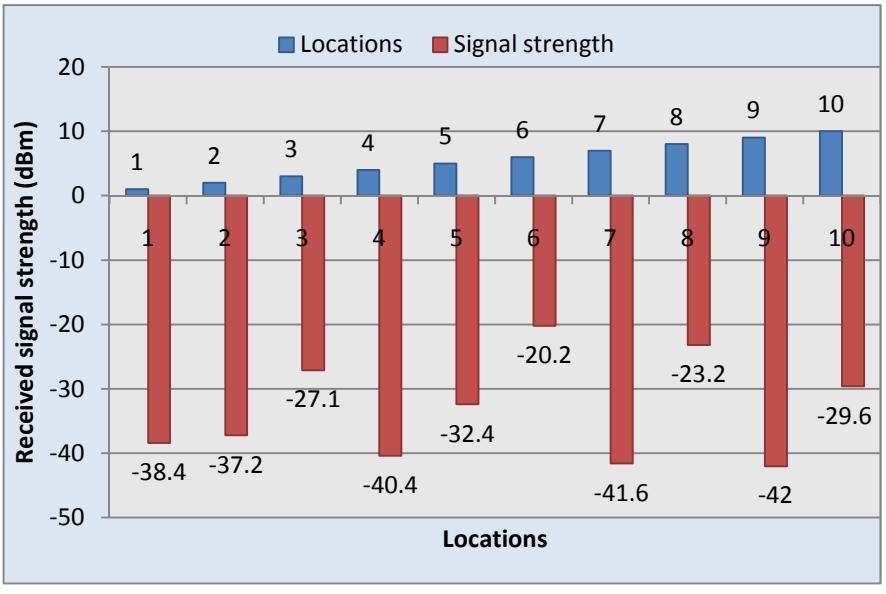


Fig. 5. Signal Strength calculated using spectrum analyzer versus locations

Table 1. Optimal placement of APs for 100 users on first floor with *Pt* =21dBm

Locations of Access points	Received signal strength Rth (dBm)	Actual placement of Access point coordinates	Access point coordinates by PSO
Location A	-38.4	(18.906, 10.934)	(22.16, 11.32)
Location B	-37.2	(24.725, 2.511)	(27.34, 4.54)
Location C	-27.1	(35.905, 1.592)	(38.25, 1.09)
Location D	-40.4	(42.183, 12.925)	(38.98, 10.25)
Location E	-32.4	(38.967, 26.248)	(40.54, 24.25)
Location F	-20.2	(41.877, 43.705)	(40.98, 44.72)
Location G	-41.6	(36.364, 52.434)	(32.22, 50.35)
Location H	-23.2	(26.104, 25.941)	(25.12, 24.23)
Location I	-42.0	(30.545, 12.006)	(32.45, 10.26)
Location J	-29.6	(33.607, 25.941)	(30.25, 24.31)

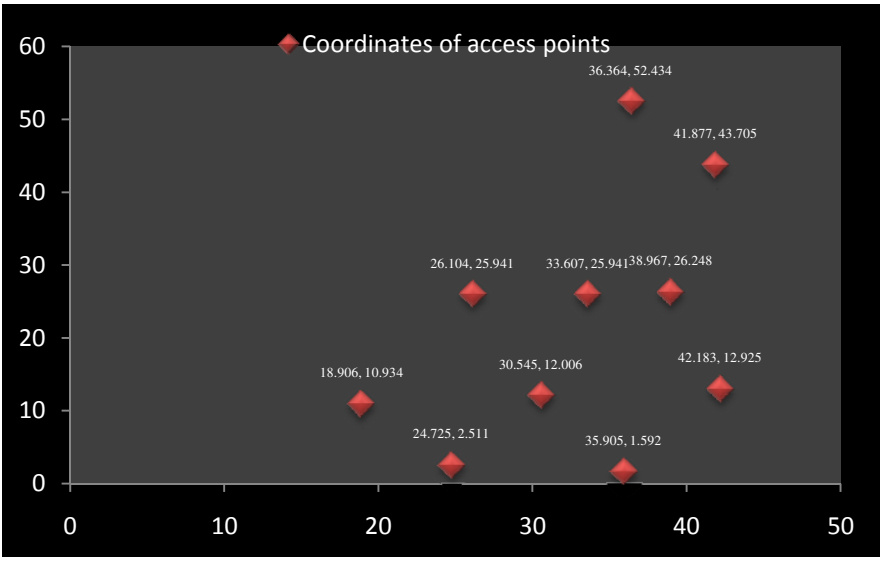


Fig. 6. Actual placement of access points

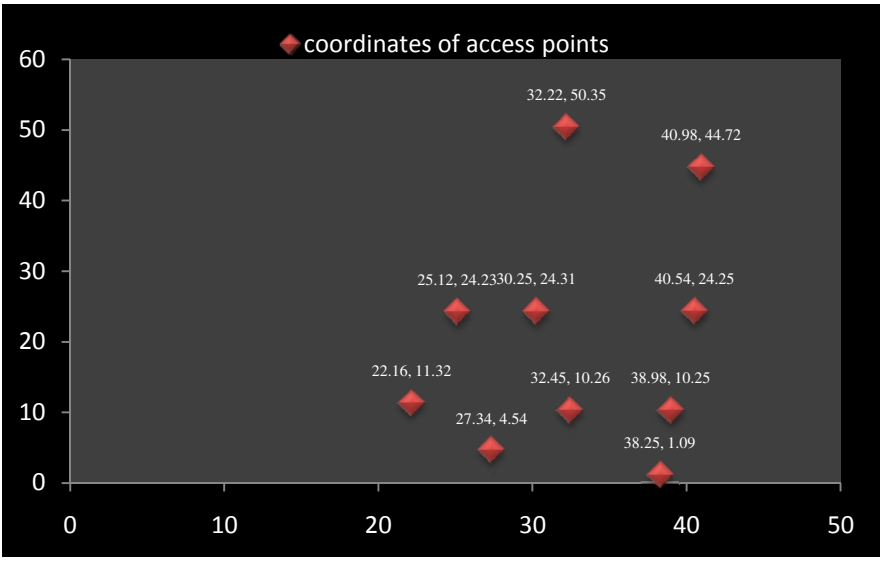


Fig. 7. Distribution of access points by using PSO

7 Conclusion

In this paper we have presented particle swarm optimization to predict the signal strength in indoor environments. This helps in optimizing the Wi-Fi network and reduces the cost of implementation. It is observed that the size of the design area and the number of users and their locations have an effect on the location and the number of APs needed to cover users and path loss increases as a function of distance between the transmitter and user. This shows that the method confirms the expected behaviour of attenuation. Further work will be extended to include obstacles in the mathematical model presented in this paper and test will be conducted.

Acknowledgement. I would like to thanks the Ministry of Science, Department of Science and Technology, New Delhi for sponsoring this work under Women Scientist Scheme WOS-A, Grant No: DST-498-DPT.

References

[1] Soungalo, T., Renfa, L., Fanzi, Z.: Evaluating and Improving Wireless Local Area Networks Performance. International Journal of Advancements in Computing Technology 3(2), 156–164 (2011)

[2] Wireless LAN Solutions At-A-Glance - Cisco, United States, http://www.cisco.com/web/AP/wireless/pdf/WirelessLAN.pdf

[3] IEEE 802.11b Wireless LANs, http://www.3com.com/other/pdfs/infra/corpinfo/en_US/50307201.pdf

[4] Particle Swarm Optimization (PSO), http://www23.homepage.villanova.edu/varadarajan.../PSO_meander-line.p...

[5] Kouhbor, S., Ugon, J., Kruger, A., Rubinov, A.: Optimal Placement of Access Point in WLAN Based on a New Algorithm. In: Proc. Of the International Conference on Mobile Business-IEEE/ICMB, pp. 592–598 (July 2005)

[6] Kouhbor, S.: Physical Security Enhancement in WLAN Systems. In: ISSNIP-IEEE, pp. 733–738 (2007)

[7] Li, B., Dempster, A., Rizos, C., Barnes, J.: Hybrid Method for Localization Using WLAN. In: Spatial Sciences Conference (2005), http://www.gmat.unsw.edu.au/snap/publications/lib_etal2005c.pdf

[8] GD/JANET/TECH:Surveying Wireless Networks (2008), http://www.ukema.ac.uk/documents/.../technical../

Mathematical Modeling of Environmental Optimization of Urban Green Systems: With an Emphasis on Biodiversity and Environmental Conservation

Meenakshi Dhote[1] and Kalpana Shankar Khurana[2]

[1] Department of Environment Planning, School of Planning and Architecture, New Delhi
[2] Deputy Director, Delhi Development Authority, New Delhi
On Deputation; Deputy Director, All India Council of Technical Education, New Delhi
meenakshidhote@gmail.com, kalpana@anikalp.com

Abstract. The objective of this paper is to evolve a Mathematical Model which can be applied to any Urban Green System to achieve maximum environmental benefits from the limited amount of urban green space. Most of the urban green spaces being designed and developed do not have their functional use, ecological stability, social benefit and economic sustainability at their optimal. To overcome this fragmented development, holistic multidisciplinary approach towards functional, ecological, social and economical aspect needs to be developed and adopted. Six major parameters have been identified that can influence the indicative environmental health of any urban green system. These are: Fragmentation of urban green, Biodiversity, Naturalness, Pollutant and Air quality, Hydrology and Noise. The model proposed to be developed, would be a multi-objective optimization model, wherein the objective is to maximize the environmental benefits with optimizing the fragmentation, biodiversity and naturalness keeping the minimizing results of pollutants in the air quality as indicators. The overview of the model is intended with a minimization of the economic factor. Further, an attempt is made to provide suitable numerical values to the mathematical model based on real life data from typical sites from the metropolitan area of Delhi.

Keywords: environmental optimization, biodiversity, environmental conservation urban green spaces.

1 Introduction

Urban Green Spaces are essentially public spaces located in Urban areas, mainly covered by vegetation which are directly used for active or passive influence on the urban environment, accessible to citizens, serving the diverse needs of citizens and thus enhancing the quality of life in cities. They are expected to have an

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 867–879.
springerlink.com

impact on effects of diverse factors like traffic flow and emissions, air quality, microclimate, noise, and social well-being. Urban greens have a role to play in the fight against climate change. Green space in cities offers a significant environmental benefit, including pollution control, water management, wildlife havens and biodiversity. Besides economic benefits are anticipated from the environmental and locational aspect of greens.

1.1 Benefits of Urban Green Spaces

Though the benefits of greens are immense and cannot be listed down in toto, a short description can be classified under environmental, social, and economical benefits. Environmental benefits cover issues like Biodiversity Bank, offers biodiversity habitats, prevention of soil erosion, absorption of rainwater; absorption of pollutants by trees; microclimate benefits; reduction of noise pollution and most importantly from the point of view of a scholar, indicator of overall ecological health. Social benefits are the most difficult to quantify. These include provision of recreational facility; a sense of social place; visual asset to the public; opportunity for social congregation, bringing natural near to humans in an urban environment. Economic benefits are the most neglected in Indian conditions, primarily because the greens are not as maintained as they should be and secondly, these are not emphasized enough. One of the major economic benefits recognized in the western countries is that proximity to property raises its value.

1.2 Elements of Urban Green System

The requirement of urban green spaces has been well established through the ages. Man has a very basic need to stay near nature. In towns and cities, this desire has to be satisfied by the various urban parks. Ancient people attested a high value to the public and private urban greens. Public participation in organized games, athletic exercise and public bathing proved the emphasis. The urban greens got constructed by Kings or rulers, as a product of economic stability, prosperity availability of leisure time. Initially the open spaces had restricted access to public. Later they became the property of the rich. And with the advent of liberalization, they were open for public.

Now it is seen as a responsibility of the government to provide space for passive recreation and active sports activity. In many a cities, by the time the need for the urban greens was realized and manifested, it was already densely built and the built-up land had become so valuable that the cost of acquisition for conversion into parklands had become prohibitive. This resulted in uneven distribution patterns with availability of green recreational areas being very haphazard.

In the Indian scenario, the hierarchical division of the urban greens or “recreational” landuse has been initiated in the various Master Plans of the cities. These urban greens are primarily identifiable by virtue of their size and subsequently by the kind of recreational requirements that they fulfill. This is an established fact that the budget allocated to the development of these urban greens, unlike the

developmental charges in any other Landuse type, is normally considered as a non- refundable expense. No government ever gets any monetary returns on the development of an urban green, thus the 'unsaid' priority is never too high.

1.3 Problem Identification

First hand observations by virtue of being a part of a Government agency that owns, manages and develops large areas of greens in Delhi, has led to certain conclusions and hypothesis which have initiated this research. Most of the urban green spaces being designed and developed do not have their functional use, ecological stability, social benefit and economic sustainability at their optimal. To overcome this fragmented development, holistic multidisciplinary approach towards functional, ecological, social and economic aspect needs to be developed and adopted. This leads the authors to many logical questions assuming that, within an urban area, the available greens should be developed with such a strategy that minimum developmental effort should yield maximum environmental benefits. In other words Environmental Optimization of the Urban Green System is required.

1.4 Objective

Therefore, this paper is aimed at exploring the possibility of optimizing the six identified criterions that provides an optimal solution to develop and maintain an urban green system by way of maximizing the environmental benefits within the given constraints of limited urban metropolitan area and budget. A special attempt has been made on the study of Biodiversity criteria, assuming it as the most noticeable indicator of environmental health.

2 Environmental Criterion of Urban Green System and Their Optimization

In a study leading to this paper, an analysis was performed to identify various possible criterions that could explain all urban green system typology of an Indian metropolitan. These were further represented in terms of quantifiable indicators. Attempt has been made to represent these criterions as a relationship of quantifiable entities so that principles of Operation Research may be applied to them for Environmental Optimization of the urban green system. Verification of this optimized relation is further attempted through real time examples from the Capital city of Delhi. Criterion to explain various urban green system aspects have been identified and constricted to the following six- Fragmentation of urban green, Biodiversity, Naturalness, Pollutant and Air quality, Hydrology and Noise. Criterions are further represented as quantifiable indicators. This is then processed for mathematical modeling.

2.1 Fragmentation of Urban Green

The land patches that are developed under green land use in an urban situation are more than often not continuous and are of various sizes and shapes. At times the area planning is based on principles that make a green space focal. Sometimes they are incidental land area that arises from the 'thoughtful' planning of the zone where logical emphasis is given to almost all other use than green, until and unless the green intentionally planted there. Situation of natural features like rivers, ridges, drains provide a much needed continuous green, though not always maintains to urban use standards. Fragmentation is the assessment of the effectiveness of the green by virtue of its physical constitution. It has been classified under four subheadings- size, shape, isolation and continuity.

2.1.1 Size

Smaller patches of green are considered as less environmentally beneficial than larger patches. There measure in a basic format is area in square meters

$a \times b$, where a=length of patch and b= width of patch

Mathematically it can be indicated as

Maximize Environmental Benefits = $f(A) = a \times b$,

Where f is a function denoting "no. of patches of green"

2.1.2 Shape Index

Narrow patches of green are subject to edge effect and thus are considered environmentally rich. Shape index consists of the breadth of the patch divided by its circumference. If the shape index is high, the edge effect is small. If patches have more edge and less breadth, the effective size of the patch is much less. A high shape index is indicative of poorer ecological quality.

Mathematically it can be indicated as $b \div 2\,(a+b)$, where a=length of patchand b= width of patch

Maximize ecological quality = $g(SI) = b \div 2(a+b)$

where a = length of patch and b = width of patch

Where g is a function denoting "shape index"

2.1.3 Isolatedness of Green Areas

This is a measure of interconnectivity of urban green. The optimal connectivity condition would be continuous. To estimate this each patch is given a single value according to its proximity to the nearest adjacent green, irrespective of its quality or size. Inter- patch distance is measured as the shortest intervening distance between the boarder of the case study site and the borders of neighboring patches; l, where l=shortest distance between the boarders of the case study site and neighboring patches

Mathematically the optimization can be depicted as

Minimize inter-patch distance = Minimize d(CSS, NPi)

Where CSS = case study site and

NPi for i=1, 2, …… n are the "n" number of neighbouring patches
where d is a function which denotes distance between CSS and NPi .

2.1.4 Connectivity of Urban Green Areas

The isolation effects can be alleviated by the presence of connectivity elements. The more the elements, and more diverse they are, the greater is the potential environmental benefit. For each urban green space, the number of green connectivity elements by which it is connected to another urban green space are counted and the number of different types of such corridors is assessed. These two values can be multiplied by each other to indicate the connectivity value for the site.

$\sum n \times q$, where n=number of connective elements
q= assessed value given to each type of connection on a scale of 1 to 5

Mathematically the optimization can be denoted as
Maximize connectivity value of CSS = $h(\sum n \times q)$
Where h is a function which denotes the connectivity value
$\sum n \times q$, where n=number of connective elements and
q= assessed value given to each type of connection on a scale of 1 to 5

2.2 Biodiversity

Biodiversity is defined by the Convention on Biological Diversity (CBD) as "The variability among living organisms from all sources including, inter alia, terrestrial, marine and other aquatic ecosystems and all the ecological complexes of which they are part; this includes diversity within species, between species and of ecosystems."A preview to certain established concepts and theories regarding biodiversity doubly emphasizes the need for Optimization of biodiversity for the required results. There are many theories about how the number of species affects ecosystem functions. One of these is the redundancy hypothesis, shown below, which assumes that the rate of ecosystem functions increases, as more species are present, but only up to a point. After this point, more species are redundant and do not have any additional effect on ecosystem functions. In this theory the loss of species has no initial effect, but after a certain point functions begin to suffer.

Another theory, the rivet hypothesis, shown below, claims that each species added to an ecosystem increases ecosystem functions, although the increase in function may increase more slowly as more species are included. In this model any loss of diversity should be noticeable. Opposed to theories that assume a definite relationship between diversity and ecosystem functions is the idea that there is no fixed relationship, and that the functions of an ecosystem are the result of what the interactions between species are. In this case what is important is not how many species are present but which species are present together and what environment they are in. According to this hypothesis which of these theories is most accurate is not certain, given the problems of scale and the complexity of the measurements. Thus optimization becomes essential.

Diversity of ecosystems is not an easy property to define or measure. There are many methods and ways of estimating Biodiversity. The scale at which the assessment is carried out decides the method employed. Also, the purpose for which the assessment is done decides its methodology. At the global level, continent level, national level, assessments could limit to understanding the types of biomes, ecosystems and aspects related to species within them. Remote sensing data is often used at this scale. Ground truthing, in form of few surveys are carried out in critical areas to validate the macro level data. At the regional level and local level detailed floristic and faunal surveys are carried out. In fact, many of them are used in routine botanical and zoological surveys. Quadrat methods, Transact method etc. are some standard surveys to record type and frequency of species. Application of Remote Sensing and GIS packages are recent development in this area. Methods for assessing biodiversity can be broadly grouped under two categories: Qualitative methods and Quantitative methods. Principles of Operation research are better applied to quantitative methods.

Quantitative methods to assess diversity need a large amount of data on species type and numbers. Hence they are generally attempted at small scales. However, at landscape level with the help of satellite data some quantitative assessments have been attempted. Following is a brief description of the quantitative methods to assess biodiversity.

2.2.1 Species Richness

It is the total number of species types in an area. In other words, the variety. Mathematically it can be indicated as

$$R = \sum_{i=1}^{n} Si$$

Where, S is species types, i to n.

2.2.2 Biodiversity Index

Popularly known as Shannon Diversity after the ecologist, not only indicates the type of species but also their even distribution. The following illustration explains the concept. The index is calculated by;

$$H = -\sum_{i=1}^{n} Pi \log Pi$$

Where n = number of species, Pi = proportion of the ith species as a proportion of total number.

2.2.3 Fragmentation

Fragmentation is an analysis of the disposition of greens. Mathematically it can be expressed as, the number of green patches within an area i.e.,

$$F=\sum_{i=1}^{n} Pgi / A$$

Where, Pgi are green patches i to n and A is study area.

2.2.4 Patchiness (Patch Density)

Patchiness is a measure of the density of patches of all types, be it green or non green. Mathematically it can be expressed as

$$PA=\sum_{i=1}^{n} Ti/N$$

Where, Ti are types of patches i to n and N total number of patches.

2.2.5 Porosity

Porosity is a measure of the number of patches of a type, regardless of patch size. Mathematically, it can be expressed as:

$$PO=\sum_{i=1}^{n} Pi$$

Where, PO is porosity and P is patch type, with numbers I to n.There are others landscape metrics contagion, interspersion, fractal dimension. These require extensive field data and logarithmic analysis and can be used for further detailed analysis. 16

2.2.6 Species Area Curve

This is a model that establishes the relation between the area of habitats and the numbers of species living in them. Quantitatively it is expressed as

$$S=CA^{z}$$

Where S is the number of species, A is the area of the place where the species live, and C and z are constants, and vary from one group of organisms to another.

For purposes of calculating the rate of species extinction, C can be ignored; z is what counts. In the great majority of cases z falls between 0.15 and 0.35. When species are able to disperse easily from one place to another, z is low. Birds have low z value; land snails and orchids a high z value. The higher the z value, the more the species numbers will eventually fall after the area is reduced. The rule of thumb is that when an area is reduced to one tenth of its original size, the number of species eventually drops to one half. This corresponds to a z value of 0.30 and is actually close to the number often encountered in nature 17.

2.3 Naturalness

This criterion considers natural, unmodified habitat as the ecological benchmark. Landscaping and the implementation of radical management strategies of green areas will generally reduce ecological naturalness. Criteria considered are erosion due to recreational use, loss of rare species and the presence of exotic species.

2.3.1 Degree of Disturbance

The amount of disturbance is indicative of the intensity of disturbance directly resulting from human impact on the site. The more obvious effect is the field- layer vegetation. Estimation of the proportion of the surface which is heavily worn out is indicative of the disturbance caused. Erosion of field- layer vegetation is a direct indicator of anthropogenic impact at the site. Mathematically it can be indicated as a ÷ A, where a= worn out area and A= total area of site. For Optimization, this disturbance needs to be minimized.

2.3.2 Proportion of Exotic and Rare Species

A characteristic feature of urban habitats is the presence of exotic species. The availability of altered and novel habitats in urban regions, as well as obstacles to dispersal, provide the opportunity for novel communities including exotic species to become established. This aspect may be mathematically indicated as E + R, where E is number of various exotic species, R is number of various rare species

Minimize exotic species and rare species = E + R

2.4 Pollutant and Air Quality

Urban green features have two beneficial effects on the urban atmosphere. They are capable of absorbing pollutants from the atmosphere and also release humidity. Air quality and capacity of urban green to absorb pollutants and to release humidity are considered.

2.4.1 Air Quality

Simply put, the quality of air is measurement of known air pollutant. Since air pollution is known to be different in different seasons, data for four different seasons is to be considered. Pollutants considered can be -NO2, SO2, CO, O3, PM10and humidity. This study proposes to consider Particulate Matter PM10 as the key indicator.

2.4.2 Capacity of Urban Green to Improve Air Quality

The volume of vegetation and the surface area of water and unsealed soil are indicative of the capacity for removing pollutants from and releasing humidity to the atmosphere. Each biotope is given a value which is indicative of the volume of

vegetation. These are multiplied by the proportions of each biotope present. The surface area of water and unsealed soil are recorded separately. Mathematically

$(V \times b) \div A$

$w \div A,$

$u \div A$

where V=estimated volume of leaf surface

b= value given to each type of biotope

A= total area of site

w= surface area of water body

u= Area of uncovered surface

$F(iaq) = \sum(V \times b)/A + w/A + u/A$

This function needs to be maximized.

2.5 Hydrology

In a natural ecosystem, water input occurs in the form of precipitation, most of which is absorbed by the surface soil layer or flows into water bodies. In urban areas, most precipitation falls onto hard, virtually impermeable surfaces, from which it rapidly flows into the municipal drainage system, with the result that the acquisition of water is often a major difficulty for urban vegetation and other living organisms. Patches with unsealed soil surfaces benefit from normal water dynamics. Artificial drainage is another aspect of drainage.

2.5.1 Soil Sealing

The amount of soil sealing is estimated by assessing the amount of surface which is affected by the different levels of sealing. Soil sealing also affects other soil functions. Amount of soil surface in each of the soil sealing categories which lies within the boundaries of the case study site is measured as Constructed (covered with building); covered (hard surface); compact (grit surface or sports field); non-sealed. Mathematically it is denoted as an × c, where an=area of various soil sealing categories as above and c is comparative value assigned to each type of category. It should be minimized for optimum results.

2.6 Noise

Noise is recognized as an important means of anthropogenic disturbance to the environment. Large urban green features such as trees are able to reduce the effect of this disturbance. Measurement of noise is taken at least six different locations at peak hours. N may be the measured where it is the diversion from standard value of noise pollution standards. This measure also needs to be minimized.

3 Verification by Real Time Values

For the purpose of this study, three sites in the urban city of Delhi have been identified in such a manner that they provide variety in environmental situation.

3.1 Salvage Park, Mayapuri

This is a Master plan recreational area in the Western side of Delhi. The reasons for its selection are primarily its popularity and extensive use by public. There are large grown trees and a much worn out play field. Its proximity to an adjacent recreational green is also significant.

3.2 Sanjay Lake, Trilokpuri

This park is the only consolidated large green in whole of East Delhi. It has a large lake as a landscape feature. Attempts have been made to rejuvenate the water level of in the lake by bringing in the STP overflow of nearbyKondli sewage plant. Also adjacent drain inflow is to be treated before addition to the lake. The proximity to the National highway and nearby highly densely populated residential area is characteristic to the site. Also the kind of existing vegetation is primarily Eucalyptus, which is not considered of high environmental value.

Table 1. Data for the three case studies

Sr. No.	Value	Symbol	Salvage Park Mayapuri	Sanjay Lake Park Trilokpuri	SwarnJayanti Park Rohini
1	Total Area	A	89304 m^2	799766 m^2	731025 m^2
2	Length of patch	a	19	2051 m	1031 m
3	Average Breadth of patch	b	117 m	224 m	269 m
4	Circumference	∑(a+b)	89304 m	7817 m	4809 m
5	Shortest distance between the boarders of patch and neighbouring patch	L	22 m	16 m	20 m
6	Number of connectivity elements	n	19 galis 1 across road	2 drains 1 cross road 4 along road	1 cross road
7	Assessed value given to each type of connection on a scale of 1 to 5	q	2 1	4 1 2	1
8	Number of different types of habitat present	T	4	5	5
9	Worn out area	a_w	9215 m^2	111619	102357
10	Surface area of water body	W	0	128152	28759
11	Value given to each types of biotope on a scale of 1 to 5	u	1,2,3,4	1,2,3,4,5	1,2,3,4,5

Table 2. Data under assimilation

Number of various exotic species	E
Number of species counted	α
Number of various rare species	R
Number of specimen counted	β
Diversion from standard value of PM_{10}	P
Estimated volume of leaf surface	V
Area of various soil sealing categories	a_n
Comparative value assigned to each category	C

Table 3. Values for the three case studies

Sr. No	Criteria No.	Criteria	Description	Formula	Salvage Park Mayapuri	Sanjay Lake Park Trilokpuri	Swarn Jayanti Rohini
1	1.1	Size	Area in square meter	ax b=A	89304 m^2	799766 m^2	731025 m^2
2	1.2	Shape Index	Breadth of the patch divided by its circumfe-rence	b÷2(a+ b)	117 ÷ 89304	224÷ 7817	269 ÷ 4809
3	1.3	Isolatio-ness of Green Areas	Shortest dis-tance be-tween the boarders of the case study site and neighboring patches	L	22	16	20
4	1.4	Connec-tivity of Urban Green Area	Number and value of green con-nectivity elements	∑ n x q	(3×2) + (1×1) =7	(2×4)+ (1×1)+ (4×2) =17	(1×1)=1
5	2.1	Species diversity	Number of species found, with distinction between rare and exotic species	E α + R β			
6	2.2	Biotope diversity	Number of different types of habi-tatpresent	T	4	5	5
7	3.1	Degree of Distur-bance	Amount of disturbance	$\sum(a_w u) \div A$	(9215×2)÷ 89304 =	(111619×2) ÷799766 =	(102357 ×2)÷ 731025 =

3.3 Swarn Jayanti Park, Rohini

This is also a large Master Plan green located in the suburbs of Delhi. Though it is the only large green in Rohini, it has number of greens nearby. Also adjacent is a Commercial water amusement park. Besides there are three lakes of varied sizes that provide opportunity for kinds of biotopes.

Proportion of exotic and rare species, Air Quality, Capacity of urban green to improve air quality, Soil Sealing and noise are some of values that are under assimilation.

4 Discussion

The three different case studies provide large range of environmental biotope situation. Attempt has been made to optimize the benefits of the greens with an extensive input of the physical characters of each site. The data from the different site in terms of the biodiversity and air quality is difficult to get. The aspect that also needs to be analyzed is the economic input. This is presently being considered and is an ongoing process. Though the model that would be the most effective would have to include aspects of economics as well.

5 Conclusion

It is possible tomaximize the environmental benefits of an Urban Green System with optimizing the landscape inputs and lead to developing a real time Optimization Model.

References

[1] Ali, S., Fatehally, L.: Common Birds. National Book Trust or India
[2] Canter, L.: Environmental Impact Assessment, pp. 390–396
[3] Donley, R., Bowman, R., Marzluff, J.M. (eds.): Avian Ecology and Conservation in an Urbanizing World, pp. 92–93. Kluwer Academic publishers, Dordrecht (2001)
[4] Farina, A.: Principles and Methods in Landscape Ecology. Chapman and Hall Ltd., London (1998)
[5] Gardener, R.H., Klopatch, M.J. (eds.): Landscape Ecological Analysis-Issues and Applications. Springer, New York (1999)
[6] Gilbert, O.L.: The Ecology of Urban Habitats. Chapman and Hall (1989)
[7] Gutzwiller, K.J. (ed.): Applying Landscape Ecology in biological Conservation, p. 48. Springer, New York (2000)
[8] Hawksworth, D.L., Kirk, P.M., Clarke, S.D. (eds.): Biodiversity Information: Needs and Options. Unep, Iubs, Iucn, Cab International (1997)
[9] http://www.CanadianBiodiversity.com
[10] http://www.environmentalbiologysequence-ecosystems.html
[11] Hugget, R.J.: Fundamentals of Biogeography. Routledge, New York (1998)

[12] Iirs, T.C.: Biodiversity and Environment. Geographic Information System. Perspectives; IIRS, 35–41 (2000)
[13] Lyndsay, H., Watts, S. (eds.): Essential Environmental Science-methods and techniques. Routledge, London (1996)
[14] Mani, M.S.: Insects. National Book Trust of India, New Delhi (1997)
[15] Nair, S. C.: The Southern Western ghats-a biodiversity Plan. INTACH Series in Ecology and Sustainable (1991)
[16] Philip, W.C.: Urban Habitats, London and New York (1999)
[17] Roberts, M., Chapman, J.: Biodiversity the Abundance of Life. Cambridge University Press, London (1997)
[18] Sundaram, A.M.: City Green Plan as a means to achieve sustainable development (Background paper), Map India 2002, New Delhi (2002)
[19] Tivy, J.: Bio Geography: A study of Plants in the Ecosphere, 3rd edn. Longman Scientific & Technical (1993)
[20] UNEP, Global Biodiversity Assessment. Cambridge University Press, London (1995)
[21] Wilson, E.O.: The Diversity of Life, pp. 264–265. Penguin Books

Heuristics to Find Maximum Independent Set: An Overview

Kedar Nath Das[1] and Biplab Chaudhuri[2]

[1] Department of Mathematics, NIT Silchar, Assam
[2] Department of Mathematics, G.C. College, Silchar, Assam
kedar.iitr@gmail.com, biplabs2003@yahoo.com

Abstract. The maximum independent set problem (MISP) is to find the largest subset of vertices of a graph such that none of these vertices are connected by an edge (i.e., all vertices are independent of each other). It has many real life applications. In solving MISP, especially where the complexity of the graph increases, exact methods become impractical or sometimes become too slow to provide a solution. In those cases, the evolutionary heuristics are being successfully used by many researchers. In this paper, a critical review of different existing approaches in evolutionary methods to solve MISP, has been made. Later, some of the future scopes are being presented.

Keywords: Maximum independent set, Evolutionary algorithms, Combinatorial optimization, Hybrid algorithms.

1 Introduction

Consider an undirected graph $G=(V,E)$. The problem of determining a set $V^* \subseteq V$ such that $\forall i, j \in V^*$, the edge $(i,j) \notin E$ and $|V^*|$ is maximum, is known as maximum independent set problem (MISP). The MISP and related problems are formulated as nonlinear problems. It has the variety of applications. To cite a few, it could be modified for predicting the secondary structure in RNA (Y. Takefuji et. al., [19]). It is also useful in coding theory, geometry, VLSI design, classification theory, etc (A. Sakamoto, et.al [16]).

MISP is NP-complete (M. R. Garey and D. S. Johnson, [10]). Once the NP-hardness of a combinatorial optimization problem is established, the search for an optimal solution is abandoned. Especially, for a graph of some hundreds of vertices with computational complexity, the exact methods fail to give a solution. At this juncture, approximation approaches play an important rule at least to provide a near optimal solution. Therefore, since last couple of years, it became a challenge among the researchers to find a good heuristic, i.e. a polynomial running time algorithm to solve MISP.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 881–892.
springerlink.com

S. Butenko [6] listed out the heuristic approaches available in the literature to find MISP. The heuristics like simulated Annealing, Artificial neural network, genetic algorithm, greedy randomized adaptive search procedures, tabu search and the heuristic based on continuous optimization; have been discussed in the paper, with various areas of applications like (a) matching molecular structures by clique detection, (b) macromolecular docking, (c) integration of genome mapping data, (d) comparative modeling of protein structure, (e) Covering Location Using Clique Partition. In this paper a novel approach has been developed to solving computationally difficult combinatorial optimization problems on graphs, namely maximum independent set, maximum clique, graph coloring, minimum dominating sets and related problems. The application areas of the considered problems include information retrieval, classification theory, economics, scheduling, experimental design, and computer vision among many others.

In this paper, a critical review is carried out for the approaches of evolutionary heuristics that have been used to solve MISP. The review outputs are presented in section 2. The conclusion and future scopes have been discussed in section 3. Section 4 includes the list of references.

2 Review Outputs

The hierarchy of development of concepts and ideas implemented on solving the MISP using evolutionary methods are discussed in this section. Of course there are a number of approaches available in the literature to solve MISP. However, this paper concentrates only on some mostly used heuristics. The different heuristics approaches used in literature are categorized in different subsections as follows.

2.1 Genetic Algorithm

Genetic algorithm (GA) is a directed search technique based on natural selection. While under goes with its operators namely selection, crossover and mutation, it has been empowered by the Darwinian law of 'survival of the fittest'. Since last few years, it has been successfully used to solve NP-hard combinatorial optimization problems.

To solve MISP, an NP-hard problem, it is T. Back and S. Khuri [3], pioneered the work. They presented an alternative approach that uses GA as a generalized heuristic. The robustness of their approach based on a graded penalty function for infeasible string is demonstrated. As a conclusion they recommended to use the GA for solving MISP, irrespective of formulation of new fitness function.

The genetic algorithm uses crossover as an operator in its generation cycle. In crossover, two parents participate in each step to share their knowledge and strength to provide two new children. C. C. Aggarwal, et.al [2] tried to realize the effect of power of knowledge based mechanism in genetic algorithm. They proposed a new idea of knowledge based crossover to find the independent set of a graph. Out of the two parents considered for crossover, they assume one as that

string having the best objective function value from the feasible set of the population whereas the other is so constructed to maintain the diversity in the search space.

A. Sakamoto, et.al [16] presented a genetic algorithm for MISP. A permutation encoding with a greedy decoding have been used to solve the MISP. The DIMACS benchmark graphs are used to test the algorithm. For most graphs solutions found by the proposed algorithm are optimal, and there are also a few exceptions that solutions found by the algorithm are almost as large as maximum clique sizes. The simulation results are compared with hybrid GA (called GMCA), and one of the best existing maximum clique algorithms (called CBH). The proposed GA beats the both in the final solution.

A. Taranenko and A. Vesel [20] designed a genetic algorithm based on the elitist strategy. In the algorithmic procedure, they have initialized the population that undergoes the process of **selection** (Roulette Wheel), **crossover** (Partially mapped, Order, Cycle, Modified Partially mapped) and **mutation**. The first three crossovers are well known. The authors mean the modified partially mapped crossover differs to partially mapped crossover, where the parent sequences between the crossover points are not necessarily exchanged. The central attraction of their paper is to implement the elitist strategy (called elitism) in the GA cycle, just after mutation. Elitism preserves the most-fit chromosomes of a particular generation to the next, just by combining both populations obtained in two conjugative generations and taking the best half. This way the convergence becomes faster. The algorithm is tested on DIMACS benchmark graphs. Elitist genetic algorithm yields satisfactory results. Authors claimed that it outperforms the pre-quoted results obtained by heuristic-based GA (HGA) and permutation encoding based GA (PGA).

Many researchers have contributed their various heuristics to solve the MISP. Lee and Cho [12] have already counted the number of MISs of binary trees and planted plane trees. But there is an instance of finding the weighted maximum independent set problem (WMISP). WMISP initiates with the weighted graph, where weights 'w_i' is assigned to each of the vertices 'v_i'. The WMISP is to find an independent set $S^* \subset S$ (the set of all vertices of the given graph) such that $W\left(S^*\right) = \sum_{v_i \in s^*} w_i$ is maximum over all independent sets of S. Let there are n tasks and some of those are sharing resources, weights are given to the tasks according to their priority. When it is required to find the set of largest total weight which can be performed at same time, WMISP is coming to the picture. This problem is NP- complete (M. R. Garey and D. S. Johnson, [10]) for general graphs. For the first time S. M. A. Nayeem and M. Pal [15] provided a genetic algorithm approach to solve such a problem as follows for a general graph of 'n' vertices.

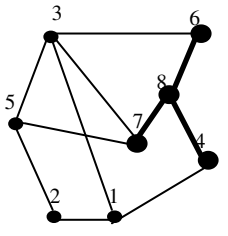
i. Fitness function:

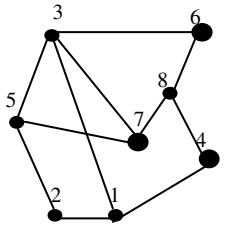
The general optimization problem of the form $Max\ g(x),\ s.t.\ h_i(x) \le 0,\ i = 1,2,...,n.$ can be transferred to

$Max\ g(x) - r\sum_{i=1}^{n}\phi\left[h_i(x)\right]$, where ϕ is the penalty function and r ($r = nW$, very large) is the penalty coefficient.

ii. Initial Population:

Random initial population is implemented by generating binary random numbers '0' or '1' for n times. Where there is '1', only those vertices of the graph and the edges joining them are selected for a chromosome. It is noted here that all chromosomes may not be feasible as shown in figure 1. The first graph here is infeasible where as the second one is feasible in the sense of an independent set.





0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0

Fig. 1. Infeasible and Feasible Chromosomes

iii. Selection:

Uniform binary tournament selection has been used with a probability of 0.5. Fitter individual has been selected in each pair of individuals, in the selection process.

iv. Crossover:

Authors used the single point crossover, where there is a crossover site need to be chosen randomly between each individual and then the right side of the site are exchanged. The probability of crossover is taken as $P_c = 0.7$.

v. Mutation:

The bitwise mutation has been considered for the GA cycle with a probability $P_m = 0.01$. Sometimes it may reduce an infeasible chromosome to a feasible one. For example, in figure 1, the second graph has the feasible set of bolded vertices, just after applying the mutation to the vertex number '8', where the bit changes from '1' to '0'.

Prior to the above paper, researchers shown that, the time required in solving MWISP, depends on the density of the graphs. But after the computational

experiments carried out in this paper, the authors left a conclusion that the density of the graph does not play an important role in solving MWISP. Also they clarified that especially for larger graphs, the proposed GA outperforms the then existing algorithms.

For a graph G, independent number is the cardinality of the largest independent set. An orientation of G is an assignment of a direction to each edge $\{u, v\}$ denoted by $u \to v$ or $v \to u$, as the case may be. An orientation of G is said to be acyclic if it has no directed cycles. There exist a relation between the independence number and its acyclic orientations. Based on this relationship, V.C. Barbosa and L. C. D. Campos [4] proposed a new formulation for solving MISP. They named it as WAO (Widest Acyclic Orientation). It helps in searching an acyclic orientation that is widest over the set of all orientations of the graph. Based on the graph theoretic notions, they focused on the design of the individuals' representation and evolutionary operators. After testing the proposed algorithm on the DIMACS benchmark graphs, they concluded that it is likely with a narrow better margin to other competitors. Authors also suggested for the future scope of investigation on better set of parameters for the evolutionary search.

It is more interesting to find the maximum independent set of a circle graph. For example consider a circle graph with 14 vertices and 7 edges as shown in figure 2(a) below. The maximum independent set can be found out with the concept of taking the edges, as vertices (see Figure 2(b)) and join the vertices if there is an intersection of corresponding edges in figure 2(a). Therefore, 'finding maximum independent set of a circle graph' is equivalent to 'finding the maximum planar subgraph in it'.

S. Wang et. al. [21], proposed a genetic algorithm to solve MISP for circle graph. The following ideas have been implemented.

i. Problem formulation:

Authors consider the binary coding to represent chromosome. To define the fitness function, they merged the constraints with the objective function value with a penalty parameter $A.h(t)$, where A is a constant and $h(t)$ is the sigmoid function defined as $h(t) = 1/(1+e^{-t})$.

ii. Genetic operators:

The two-point crossover and simple-random mutation have been implemented in GA cycle. But it is worth to note here that the operators are performed based on *conditions* instead of *probability*.

The proposed GA has been tested on randomly generated circle graphs. Simulation results have been compared with Y. Takefuji et. al.[19]. Authors concluded that the result is better than the best existing algorithm.

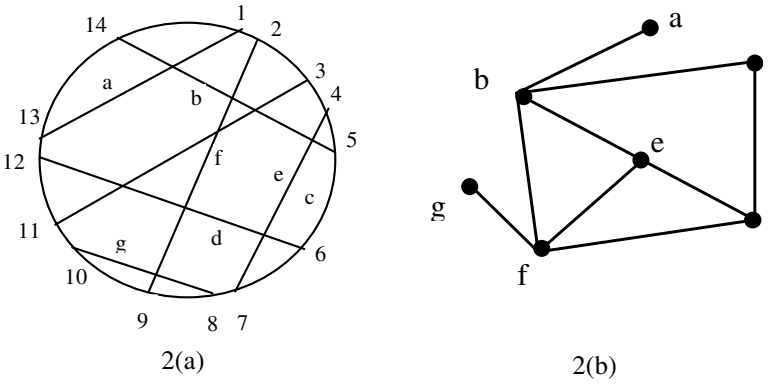


Fig. 2(a). Circle graph, **Fig. 2(b)** Maximum independent set of the circle graph.

In order to obtain better result for MISP, some instances in the standard GA operators became essential to improve by some hybridization techniques. S. Mehrabi et. al. [14] proposed a hybrid genetic algorithm for MISP. They considered the fitness of an individual in the population as the number of vertices in the solution. The steady state like selection method is adopted since in the successive generations only 50% of the current population changes. To name it as hybrid genetic algorithm, authors introduced a new crossover operator called *heuristic crossover operator* (HIX) to improve the offsprings reproductively and merged with a *repair operator* concept. After the mutation, because of the independency of the obtained vertex set, the feasibility of the child solution is not always expected. The beauty of the repair operator algorithm is that it successively removes vertices (starting with the largest degree) from the child solution, until a feasible solution is achieved. Hence this guarantees to have feasible solution in the population.

Through the simulation results obtained, authors concluded that their proposed hybrid GA to solve MISP, outperforms not only to the standard GA, but also the existing evolutionary heuristic called GENEsYs, in terms of solution quality. Also it is realized that with an appropriate mutation rate setting, the performance is far better than the other classical meta-heuristics.

M. M. Javidi and S. Mehrabi [11] presented the same algorithm and data set described above. No new output came out to present. However, they included some of the review on previous research work, in their paper.

2.2 Ant Colony Optimization

The ant colony optimization (ACO) heuristic has been inspired by the observation on real ant colony's foraging behavior and on that, ants can often find the shortest path between food source and their nest. It is inspired by colonies of real ants that deposit a chemical substance on the ground, called *pheromone*. The larger the pheromone is, the larger the probability that an ant selects the right path. ACO meta-heuristics basically a multi-agent system where low level interactions

between single agents (called artificial ants) results in a complex behavior of the whole system.

At the end of 20th century, ACO has been successfully used in solving hard combinatorial optimization problems. M. Schütz et. al. [17] used the ACO for solving MISP. The only variable component is the knowledge about the problem, i.e., the *local heuristic* involved in the probability of item selection. It is explained by an example as follows. Consider the following graph (Fig.3).

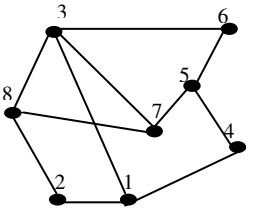


Fig. 3. Instance of MISP

In Figure 3, $|V| = 8$. Let $S'_k(t) = \{2\}$, then $F_k(t) = V - \{2\} - \{1,8\}$ $= \{3,7,4,5,6\}$. Here the set $\{1,8\}$ represents the subset of infeasible items due to the inclusion of item $\{2\}$ in the partial solution. Now the subset $\{3,7,4,5,6\}$ is the set of current feasible items and the corresponding heuristic values are as follows:

$$\tau_3(S_k(t)) = |F_3| = |\{4,5\}| = 2,\ \tau_4(S_k(t)) = |F_4| = |\{3,6,7\}| = 3,$$
$$\tau_5(S_k(t)) = |F_5| = |\{3\}| = 1,\ \tau_6(S_k(t)) = |F_6| = |\{4,7\}| = 2,$$
$$\tau_7(S_k(t)) = |F_7| = |\{4,6\}| = 2$$

Therefore, the highest score is obtained by item $i = 4$ possessing the biggest set of feasible items for the next selection step.

The designed algorithms were tested either on randomly generated graphs or taken from DIMACS benchmark graphs. For the random graphs, they claimed that the best performance was achieved for those graphs generated with higher probability values — highly connected graphs — except for the graphs with the low density. Nevertheless, the performance of their algorithm for this kind of instances was as good as the performance of greedy randomized adaptive search procedures (GRASP) according to the reported results of the method.

Later, Y. Li and Z. Xul [13] designed a modified ACO for solving MISP, based on the following three key points.

i. The solution construction process is modified by introducing a complement construction process to scatter search point in search space, as follows.

Consider $\tilde{S}_k = V - S_k$, S_k be the independent set found by ant k. Therefore $\tilde{S}_k$ can't be independent. Find the number of vertices adjacent to V_i in $\tilde{S}_k$ i.e. $Adj(V_i)$, $\forall V_i \in \tilde{S}_k$. Delete vertices from $\tilde{S}_k$ in the decreasing order of $Adj(V_i)$ until $\tilde{S}_k$ changes into an independent set. Then, like solution construction process, augment $\tilde{S}_k$ into a maximal independent set.

ii. A new computational method for heuristic information is proposed as follows. In order to produce good solution while solving MISP, $\tilde{S}_k$ should contain as many vertices as possible, and as less edges as possible. This can be realized from the following example.
Consider the graph in figure 4 that contains a total of 6 vertices. Clearly, the vertices 1 and 4 have the same degree. If the vertex 1 is selected, then $\tilde{S}_1 = \{3,4,5\}$. It contains 3 edges and we can construct an independent set of size 2. In the other hand, if the vertex 4 is selected, then $\tilde{S}_4 = \{1,2,6\}$. It contains 2 edges and we can construct an independent set of size 3.

iii. The pheromone update rule is modified in ACO, by the following

$$\tau_i(t+1) = \tau_i(t+1) - \min_i \tau_i(t+1) + \min_i \eta_i$$

where $\tau_i(t)$ and $\eta_i(t)$ represent the pheromone trail and heuristic value respectively at time t for the i[th] vertex. This is used to avoid the value τ_i increasing too large and have too big difference to η_i.

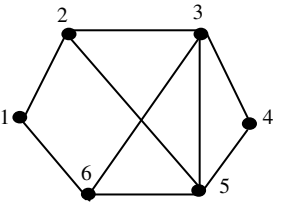


Fig. 4. Schematic figure

Authors tested their so designed modified ACO on 7 test problems and compared the result with Hofield neural network (HNN) and Canonical genetic algorithm (CGA). The comparative simulation result indicates that the proposed ACO is effective and efficient than others.

2.3 Artificial Neural Network

For artificial neural network (ANN), there exist no universally accepted definitions. However, it is generally accepted that an ANN is a network of many simple processing elements ('units' or 'nodes'), each having a small amount of local memory. The units, often organized in layers are connected by communication channels ('connections') and operate only on their local data and on the inputs they receive through the connections.

L. I. Burke [5] first focused on generating maximal independent sets by introducing a new artificial neural structure. Authors claimed through the empirical results that

i) The proposed neural approach can easily generate several maximal independent sets and it works very fast.
ii) In finding the maximum independent set, the approach finds larger sets than the heuristic approach does.

The dynamic behavior of the transiently chaotic neural network and greedy heuristic for MISP, have been observed and analyzed by X. Su et. al. [18]. They presented a transiently chaotic neural network for MISP. It is concluded through extensive simulation results that the proposed method outperforms other existing algorithms in solving p-random graph.

2.4 Greedy Randomized Adaptive Search Procedure

The greedy randomized adaptive search procedure (GRASP) is an iterative sampling method for finding approximate solutions to combinatorial optimization problem, at the end of each iteration. The overall best solution found by GRASP iteration, is treated as the GRASP solution. The GRASP possesses 4 basic components. (a) a greedy function, (b) an adaptive search strategy, (c) a probabilistic selection procedure and (d) a local search procedure.

T. A. Feo, et. al. [7] proposed a GRASP and is tested on randomly generated graphs having 400 to 3500 vertices and edge probability 0.2 to 0.9. The results are compared with Simulated Annealing, Tabu Search methods. Authors concluded that the proposed GRASP provides high quality and fast solution over others for the larger instances with 3500 or more vertices.

2.5 Tabu Search

Tabu search (TS) is a modified local search algorithm, in which a prohibition-based strategy is employed to avoid cycles in the search trajectories and to explore new regions in the search space. At each step of the algorithm, the next solution visited is always chosen to be the best 'legal' neighbor of the current state, even if its cost is worse than the current solution. The set of legal neighbors is restricted by one or more tabu lists which prevent the algorithm to go back to recently visited solutions. These lists are used to store historical information on the path

followed by the search procedure. Sometimes the tabu restriction is relaxed and tabu solutions are accepted if they satisfy some 'aspiration level condition'.

C. Friden et. al. [8] proposed a tabu search for MISP. The size of the independent set to search for is fixed and the algorithm tries to minimize the number of edges in the current subset of vertices. Three tabu lists were being used. One of storing the last visited solutions and the other two to contain the last introduced / deleted vertices. They concluded that a best neighbor may be found in almost constant time, by using hashing for implementing the first list and choosing a small value for the dimensions of the other two lists.

The same authors in [9], introduced a tabu-search-based branch and bound algorithm presented to solve MISP. However, no much work has been done to find MISP using Tabu search, till date.

2.6 Simulated Annealing

According to condensed-matter physics, the term 'annealing' refers to a physical process to obtain a pure lattice structure, where a solid is first heated up in a heat bath until it melts, and next cooled down slowly until it solidifies into a low-energy state. During the process, the free energy of the system is minimized. The solutions of a combinatorial optimization problem, generally refers to the states of the physical system, and the cost of a solution is equivalent to the energy of the state.

Without presenting any experimental result, E. Aarts and J. Korst [1] suggested the use of simulated annealing (SA) for sloving the independent set problem, using a penalty function approach. In this case, the solution space is the set of all possible subsets of vertices of the graph G, and the problem is to maximize the cost function $f\left(V^{'}\right)=\left|V^{'}\right|-\lambda\left|E^{'}\right|$, where $\left|E^{'}\right|$ is the number of edges in $G\left(V^{'}\right)$ and λ is a weighting factor exceeding 1.

Sometimes, in SA, a neighbor $s^{'}$ to an initial solution s, is generated as the next solution. Then the change in cost $\Delta F\left(s,s^{'}\right)$ is evaluated. X. Xu et. al. [22] presented a criterion to accept a solution. If a reduction in cost is found, the current solution is replaced by the generated neighbor, otherwise with a certain probability, either s or $s^{'}$ becomes the current solution. Authors defined a new acceptance function, which is the probability of accepting a transition that causes an increase in the cost ΔF, as $p=e^{\frac{\Delta F}{T}}$, where T represents the control parameter that corresponds to temperature in the analogy with the physical annealing process. Through simulation results it is claimed that with implementing this new acceptance function, the proposed algorithm can find an optimal solution to a given graph with higher probability. Moreover, it has higher convergence rate than the original SA, in finding the optimal solution on benchmark graphs.

3 Conclusion and Future Scopes

Since last 4 decades, in solving MISP, a continuous struggle and gradual improvement on developing algorithms and concepts are carried out in the literature. When the complexity of the graphs increases, the exact methods fails to provide an optimal solution, where as the heuristics can provide at least a near optimal solution to it. This paper contains a critical up-to-date review only on concepts and ideas based on heuristic techniques used for solving MISP. It is observed that heuristic methods yield many interesting and profound results. But due to the problem dependent nature of the heuristics, they all can't be compared to each other at a time. However it is concluded from the review that the hybrid versions on GA provides faster and better optimal solution then the simple GA. The heuristic crossover with repair operator, improves the solution quality. Like GA and ACO, no much works have been done in the heuristics like SA and TS. Very fewer papers are found application oriented.

Moreover, there are instances to develop new efficient operators and to design some hybrid versions of heuristic algorithms to solve the NP-complete problem MISP. It also needed to apply the so designed algorithm in solving the real life problems like MAX-CUT problem, biomedical engineering problems, etc. In ACO, a modified and efficient pheromone update rules need to be investigated.

References

[1] Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines. J. Wiley & Sons, Chichester (1989)

[2] Aggarwal, C.C., Orlin, J.B., Tai, R.P.: Optimized crossover for the independent set problem. Operations Research 45(2), 226–234 (1997)

[3] Back, T., Khuri, S.: An evolutionary heuristic for the maximum independent set problem. In: Proceedings of the IEEE Conference on Evolutionary Computation, vol. 2, pp. 531–535 (1994)

[4] Barbosa, V.C., Campos, L.C.D.: A Novel Evolutionary Formulation of the Maximum Independent Set Problem. Journal of Combinatorial Optimization 8(4), 419–437 (2004)

[5] Burke, L.I.: A neural design for solution of the maximal independent set problem. European Journal of Operational Research 62(2), 186–193 (1992)

[6] Butenko, S.: Maximum independent set and related problems, with applications. Doctoral Dissertation, University of Florida Gainesville, FL, USA (2003)

[7] Feo, T.A., Resende, M.G.C., Smith, S.H.: A greedy randomized adaptive search procedure for maximum independent set. Operations Research 42(5), 860–878 (1994)

[8] Friden, C., Hertz, A., Werra, D.D.: STABULUS: a technique for finding stable sets in large graphs with tabu search. Jr. of Computing 42, 35–44 (1989)

[9] Friden, C., Hertz, A., Werra, D.D.: TABARIS: An exact algorithm based on tabu search for finding a maximum independent set in a graph. Compu. Oper. Res. 17, 437–445 (1990)

[10] Garey, M.R., Johnson, D.S.: Computers and Intractability: A guide to the theory of NP-completeness. W. H. Freeman, San Francisco (1979)
[11] Javidi, M.M., Mehrabi, S.: On Evolutionary Algorithms for Maximum Independent Set Problem. Journal of Artificial Intelligence: Theory and Application 1(2), 54–59 (2010)
[12] Lee, C., Cho, S.: The number of maximal independent sets of unlabeled trees. Journal of Appl. Math. & Computing 18(1-2), 639–647 (2005)
[13] Li, Y., Xul, Z.: An ant colony optimization heuristic for solving maximum independent set problems. In: Proceedings of the Fifth Int. Conf. on Computational Intelligence and Multimedia Applications (ICCIMA 2003), pp. 206–211. IEEE Computer Society, Washington, DC (2003)
[14] Mehrabi, S., Mehrabi, A., Mehrabi, A.D.: A New Hybrid Genetic Algorithm for Maximum Independent Set Problem. In: Proceedings of 4th International Conference on Software and Data Technologies (ICSOFT 2009), Sofia, Bulgaria (2009)
[15] Nayeem, S.M.A., Pal, M.: Genetic Algorithmic approach to find the maximum weight independent set of a graph. Journal of Applied Mathematics and Computing 25(1-2), 217–229 (2007)
[16] Sakamoto, A., Liu, X., Shimamoto, T.: A genetic approach for maximum independent set problem. IEEE Trans. Fundamentals E80-A(3), 551–556 (1997)
[17] Schütz, M., Leguizamón, G., Michalewicz, Z.: An Ant System for the Maximum Independent Set Problem. In: Proceedings of VII Congreso Argentino de Ciencias de la Computación, vol. II, p. 1027 (2001)
[18] Su, X., Tang, Z., Wang, J.: An improved transiently chaotic neural network for the maximum independent set problem. Int. Jr. of Neural System 16(6), 381–392 (2004)
[19] Takefuji, Y., Chen, L., Lee, K., Hulffman, J.: Parallel algorithms for finding a near maximum independent set of a circle graph. IEEE Trans., Neural Networks 1(3) (1990)
[20] Taranenko, A., Vesel, A.: An elitist genetic algorithm for the maximum independent set problem. In: 23rd Int. Conf., Information Technology Interfaces ITI 2001, vol. 1, pp. 373–378 (2001)
[21] Wang, S., Wang, R., Chen, Z., Okazaki, K.: Finding a Near-Maximum independent Set of a Circle Graph by using Genetic Algorithm with Conditional Genetic Operators. In: ICNC 2008 Proceedings of the 2008 Fourth International Conference on Natural Computation, vol. 1, pp. 597–600. IEEE Computer Society, Washington, DC (2008)
[22] Xu, X., Ma, J., Wang, H.: An Improved Simulated Annealing Algorithm for the Maximum Independent Set Problem. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS, vol. 4113, pp. 822–831. Springer, Heidelberg (2006)

A Dichotomy of the Application of Genetic Algorithms in the Optimal Design of Multirate Filter Banks

Gurvinder S. Baicher

University of Wales Newport, City Campus, Usk Way, Newport, Gwent, NP20 2BP, UK
g.singh-baicher@newport.ac.uk

Abstract. This paper considers theoretical issues regarding reconstruction errors and conditions for near perfect reconstruction (PR) of the input signal for a uniform and non-uniform multiband multirate filter banks. The main emphasis is on the application of genetic algorithms in the optimization process of the design of the two forms of filter banks with further consideration of a dichotomy of design issues and optimization of the uniform and non-uniform filter banks. The issues of design of uniform multiband filter banks is based on the use of a single prototype square root raised cosine low pass filter from which the multiband filters are derived using cosine modulation technique. On the other hand, non-uniform multiband filter banks are designed using multiple square root raised cosine prototype filters from which the analysis and synthesis filters are derived using the closed form approximation. The optimization process using genetic algorithms is thus based on perturbing the bandwidth and roll-off factor of the prototype filters to generate near-optimal results.

Keywords: Optimization, uniform multirate filter bank, non-uniform multirate filter bank, genetic algorithms.

1 Introduction

Multirate processing of digital signals involves the use of variable rate sampling at different sections of a signal processing system with the key aim for efficient processing of communication signals. Multirate signal processing has been used in various applications such as digital audio systems, speech and image processing, trans-multiplexers, sub-band coding and signal data compression. Often in digital signal processing applications, there is a need for splitting a band limited signal into an upper and a lower band of frequencies at the transmitting end and then recombining these at the receiving end. Such a split of frequency bands can be usefully used for improving the coding efficiency of the transmitted signal, thereby improving the available bandwidth [1, 2]. For practical signals it is

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 893–905.
springerlink.com

sometimes more effective to have a large number of bands into which the signals are split. This multi-band split in frequency domain can lead to further refinements and improvements to the overall process. Further variations are in terms of bands that are split in equal widths of the frequency bands, commonly known as uniform filter banks or the splits that are of unequal widths of the frequency bands but derived using specific procedures to optimise the reconstruction errors at the receiving end. The latter form is commonly known as non-uniform filter banks. The problem, however, is to be able to reconstruct the received signal as close as possible to the original input signal. Where this can be achieved exactly then we say that the filter bank has a 'perfect reconstruction' (PR) property. In practice, there are several approximations used such as; the filters are not ideal and there may also be signal and filter coefficient quantization that may affect the overall performance of the system. In this respect, a simple genetic algorithm has been used to optimise the performance of the filter bank. An example of a specific form of a non-uniform multi-band multirate filter bank structure is shown in Fig. 1, where M is the number of bands into which the input signal is split. Note that for uniform filter banks r_0, r_1,....r_{M-1} will each be M.

The maximally decimated M-channel uniform filter banks have been extensively studied and the conditions for PR have been well established [2]. However, the non-uniform filter banks suffer from extensive design constraints and conditions for PR are particularly difficult to resolve [3-7]. In this case therefore, the compromise is to relax the design constraints and use some optimisation technique to optimise the network towards minimal errors. In either case, we will assume that there is no quantisation or coding errors introduced during transmission or receiving of the signals. The optimisation process is based on considering the overall network as shown in Fig. 1. An input impulse (in the form of a dirac-delta function) is applied at the input stage of the simulated network and a Fourier transform is worked out for the impulse response. This then represents the transfer function of the overall network from which the frequency response is derived. This process of optimisation thus leads to minimisation of distortion errors in a combined manner.

2 Theory and Design Issues

For a M-channel uniform filter (UF) bank (as shown in Fig.1 for which r_0, r_1,.....r_{M-1} will each be M), the input signal will be split equally in frequency bands. For an input signal of bandwidth π radians then the bandwidth of each filter will be π/M radians, also the sampling rate of each sub-band signal is reduced by a factor of M. Such a structure is referred to as maximally decimated filter bank. The M-channel non-uniform filter bank (NUF) with integer decimation factors r_k, k=0,....,M-1 as shown in Fig. 1 will be constrained to the case of maximally decimated integer values for which the following applies

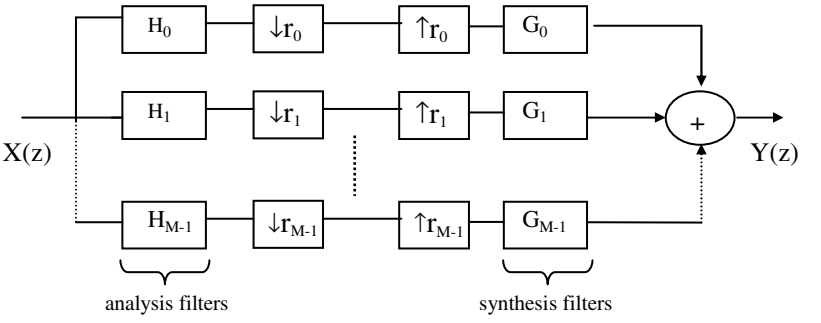


Fig. 1. Non-uniform multirate filter bank with integer decimators and integer expanders

$$\sum_{k=0}^{M-1} \frac{1}{r_k} = 1 \tag{1}$$

Most examples of NUF banks fall into the category of 'incompatible sets' of r_k for which complete elimination of aliasing errors is virtually impossible for non-ideal filters. Such filter banks may, however, be resolved into a tree-structured form representing uniform filter banks [2]. This property is useful in working out the overall aliasing error [6]. The reconstructed signal for a UF bank is given by [3]

$$Y(z)= \frac{1}{M} \sum_{\ell=0}^{M-1} \left[\sum_{k=0}^{M-1} G_k(z)\, H_k(z\, W^{\ell}) \right] X(z\, W^{\ell}) \tag{2}$$

Where $H_k(z)$ and $G_k(z)$ are the transfer functions of the analysis and synthesis filters respectively.

From eqn. 2 we may represent the following

$$A_{\ell}(z) = \frac{1}{M} \sum_{k=0}^{M-1} G_k(z)\, H_k(z\, W^{\ell}) \tag{3}$$

for $0 \leq \ell \leq M\text{-}1$

The term $X(zW^{\ell})$ for $\ell \neq 0$ represents the shifted version of the spectrum X(z). The reconstructed output is, therefore, a linear combination of the input signal and its M-1 uniformly shifted aliasing components. For $\ell=0$, equ.1 may be written as

$$Y(z)= \frac{1}{M} \sum_{k=0}^{M-1} G_k(z)\, H_k(z\,).X(z) \tag{4}$$

This represents the transfer function of the filter bank. Also since, for perfect reconstruction no distortions are expected i.e. Y(z) = X(z), then for a practical system the distortion function is given by the linear distortions of the filter bank and represented by

$$A_0(z) = \frac{1}{M}\sum_{k=0}^{M-1} G_k(z)\, H_k(z) \tag{5}$$

Note that when $|A_0(z)|$ is not an all pass condition then there will be amplitude distortion and if $A_0(z)$ has non-linear phase, then there will be phase distortion.

The aliasing components of the input signal must normally be worked out individually for ℓ=1,2,…,M-1 for which the components do not cancel each other. However, in practice an aliasing function is defined that describes the overall summing of uncorrelated aliasing components in the filter bank. This is given by

$$A_{alias}(z) = \sqrt{\sum_{\ell=1}^{M-1}\left|\frac{1}{M}\sum_{k=0}^{M-1} G_k(z)\, H_k(z\, W^{\ell})\right|^2} \tag{6}$$

For perfect reconstruction, the choice of analysis/synthesis filters must be such that aliasing is completely cancelled and $A_0(z)$ is a pure delay. The system is then free from aliasing, amplitude and phase distortions.

The input-output relationship for an M-channel NUF bank with integer decimation factors is given by

$$Y(z)=\sum_{k=0}^{M-1} G_k(z)\frac{1}{r_k}\sum_{\ell=0}^{r_k-1} H_k(z\, W_{r_k}^{\ell}).X(z\, W_{r_k}^{\ell}) \tag{7}$$

where $W_{r_k} = e^{-j2\pi/r_k}$

For such filter banks, the study here is limited to the case of maximally decimated integer values for which the condition of eqn. 1 applies.

Assuming L is the least common multiple of the r_k's and R_k are integers such that $L= r_k R_k$. Then, the non-uniform bank is resolved into an L-channel uniform bank with the new transfer functions of analysis filters given by $z^{tr_k} H_k(z)$, t=0,…., R_k-1 and $z^{-tr_k} G_k(z)$, t=0,…., R_k-1 for synthesis filters. The reconstructed output signal is then given by

$$Y(z)=\sum_{\ell=0}^{L-1} X(z\, W_L^{\ell})\; A_{\ell}(z) \tag{8}$$

where $A_{\ell}(z)$ denotes the components given by

$$A_{\ell}(z) = \frac{1}{L}\sum_{k=0}^{M-1} H_k(z\, W_L^{\ell})\, G_k(z)\sum_{t=0}^{R_k-1} W_L^{\ell t r_k} \tag{9}$$

For ℓ =0, the linear distortion function is obtained and is given by

$$A_0(z) = \frac{1}{r_k}\sum_{k=0}^{M-1} H_k(z)\, G_k(z) \tag{10}$$

In the absence of aliasing, $A_0(z)$ is the transfer function and the NUF bank is a linear time invariant system. The various aliasing components $X(z\,W^{l}{}_{L})$ of the input signal, where l=1,….L-1, that do not cancel each other out should therefore, be considered separately. An overall aliasing error function is defined as

$$A_{alias}(z) = \left[\sum_{\ell=1}^{L-1} \left| A_{\ell}(z) \right|^2 \right]^{1/2} \tag{11}$$

3 Design Implementation and Optimization

The preferred option for designing both the M-channel UF and NUF banks is based on deriving the individual analysis and synthesis filters through cosine modulation of a low pass prototype filter. While it is feasible to derive all of the filters with a single low-pass prototype filter for the UF bank, for the NUF bank several low-pass prototype filters have to be used to derive the appropriate analysis and synthesis filters by means of cosine modulation. The key feature of this method of design is to obtain an approximate transfer function of the prototype filter that will generate approximately power complementary frequency responses of frequency-shifted replicas about the centre frequencies. Also, the stop-band attenuation of the prototype filter is sufficiently large so that all alias spectra, apart from the directly adjacent one, are suppressed. The next choice is the appropriate selection of a low pass filter that may lend itself well for optimal derivation of the transfer function of the filter banks. For this option we have used the square root raised-cosine low pass filter [8].

The frequency and impulse response for such a low pass filter is given by eqns. 12 and 13 respectively.

$$P(j\omega) = \begin{cases} 1 & \text{for } \dfrac{|\omega|}{\omega_c} \le 1 - r \\ \cos\left[\dfrac{\pi}{4r} \left(\dfrac{\omega}{\omega_c} - (1 - r) \right) \right] & \text{for } 1 - r \le \dfrac{|\omega|}{\omega_c} \le 1 + r \\ 0 & \text{for } \dfrac{|\omega|}{\omega_c} \ge 1 + r \end{cases} \tag{12}$$

where ω_c is the cut-off radial frequency and 'r' is the roll-off factor.

$$p(n) = \frac{4rn.\cos\left[n\pi \dfrac{(1+r)}{M} \right] + M.\sin\left[n\pi \dfrac{(1-r)}{M} \right]}{M\left[1 - \left(\dfrac{4rn}{M} \right)^2 \right].n\pi} \tag{13}$$

where M is the number of channels of the filter bank and $\pi/2M$ is the bandwidth of the prototype filter.

The special condition for $n \rightarrow 0$ is given by

$$p(0) = \frac{1}{M} + \frac{r}{M}\left(\frac{4}{\pi} - 1\right) \tag{14}$$

and for $n = \pm M/4r$ when this is an integer value then the corresponding limits are

$$p(\pm M/4r) = -\frac{r}{M}\left[\frac{2}{\pi}\cos\left(\frac{\pi}{4r}(1+r)\right) - \cos\left(\frac{\pi}{4r}(1-r)\right)\right] \tag{15}$$

For a realizable causal filter, the impulse response must be symmetrically truncated and appropriately time-shifted. The causal impulse response that is obtained by convolving with itself results in the M^{th} band filter. The resulting raised-cosine characteristic is then approximately obtained. The overlapping of alias components of the spectra in the design of analysis and synthesis filter banks can be minimized by properly selecting the weighting factors that cause a small shift in the corresponding frequency responses. The closed form expressions for the impulse response of the analysis and synthesis filter banks [8] are given by eqns. 16 and 17 respectively.

for analysis filter bank

$$h_k(n) = 2\,p(n)\cos[(k+1/2)(n-(N-1)/2)\pi/M + (-1)^k\pi/4] \tag{16}$$

for synthesis filter bank

$$g_k(n) = 2\,p(n)\cos[(k+1/2)(n-(N-1)/2)\pi/M - (-1)^k\pi/4] \tag{17}$$

where $k = 0,1,2,\ldots,M-1$ and N is the number of coefficients used for the prototype filter.

For the design of a UF bank and based on the approximately complementary frequency responses of the frequency-shifted replicas of the prototype filter for the cosine modulated design, an optimization process based on marginally changing the specifications of the prototype filter can be expected to generate improved results. This conjecture was applied to test by using two variables i.e. the bandwidth of the prototype and the roll off factor 'r'. The bandwidth of the prototype filter was assumed to be some value π/M_p, where M_p is a real valued number approximately equal to 2M (for an 8 channel system $M_p \cong 16$). Due to the variation in the value of M_p, the restriction on 'r' i.e. $0<r\leq 1$ no longer applies. The roll off value 'r' is thus allowed to vary in the range 0 to 2. The characteristic of the square root raised cosine prototype filter with $r \cong 0.5$ could offer sufficient stop band separation for higher order filters but for lower order filters 'r' may be closer to 1 to allow for adequate aliasing error cancellation.

The NUF bank is derived from the cosine modulated uniform filter bank by using multiple prototypes p(n) and deriving the appropriate analysis and synthesis filters represented by Equations 16 and 17. The optimisation process is then based upon using the prototype variables i.e. bandwidth $\pi/2M = \pi/M_p$ and the roll-off

factor 'r' for each of the low pass finite impulse response (FIR) prototype filters to optimise the overall distortion of the filter bank.

Both the UF and NUF bank networks are implemented and tested using the Matlab platform and Simulink. A number of design examples are considered based on the design method mentioned above. The metrics derived for comparison purpose is based on the following

i) The maximum peak to peak ripple of the amplitude distortion E_{pp} is given by

$$E_{pp} = \max[|A_0(z)|] - \min[|A_0(z)|] \tag{18}$$

ii) Aliasing distortion that is derived by taking the maximum value of $A_{alias}(z)$ over all ω. This gives the worst possible peak aliasing distortion i.e.

$$E_A = \max[A_{alias}(z)] \tag{19}$$

iii) Matlab Simulink tests based on a test circuit of the form shown in Fig. 2 using a random signal at the input with uniform distribution in the range –1 to +1. The error signal v_e is then obtained from the difference between the output signal and an appropriately delayed version of the input signal. This error signal is used to calculate the root mean square value given by v_{rms} and the maximum peak to peak error voltage given by out_{pp}.

The main inspiration for optimal design of UF and NUF banks using raised-cosine prototype low pass filters was the application of real-valued genetic algorithms (GA). For this we used a Matlab compatible toolbox developed by Chipperfield et al [9]. Specific GA codes were developed both for the UF and NUF bank networks. The main task was to minimise the overall distortion of the reconstructed signal in a combined manner for amplitude and aliasing errors. It was observed that while GAs performed quite well, there was further improvement feasible through the use of some standard gradient and non-gradient based optimization algorithms. To this end, a hybrid format was used as shown in Fig. 3. The standard non-gradient based algorithm tested was the downhill Simplex algorithm and the second method is a gradient based unconstrained quasi-Newton method.

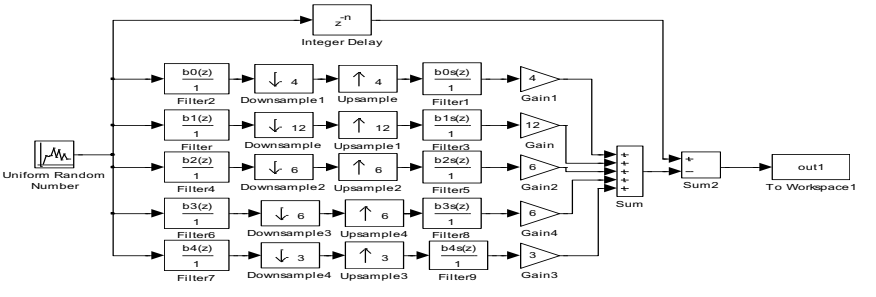


Fig. 2. Matlab Simulink test circuit for a 5-band non-uniform filter bank

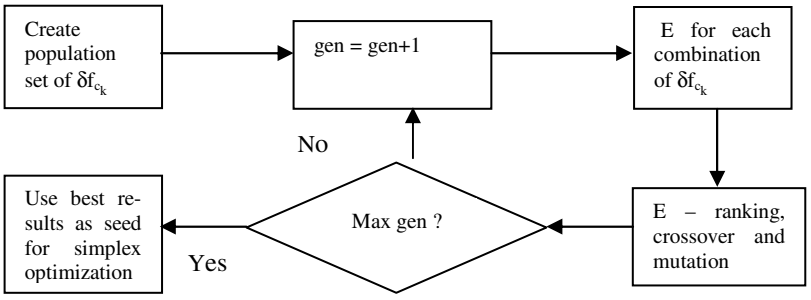


Fig. 3. Flow chart showing the GA optimisation procedure

4 Some Results

4.1 Case for UF Bank

The main purpose of the optimisation process here is to minimise the objective function with the specific aim of minimising the overall magnitude and aliasing error so that a perfect reconstruction characteristic is closely met [10]. The error objective function to be minimised and used in the optimisation process is based on the multiple objectives of linear distortion (E_{pp}) and maximum aliasing distortion (E_A) and is given by

$$\text{Obj\_err} = \alpha E_{pp} + (1-\alpha) E_A \tag{20}$$

Where α is the trade-off parameter given by $0 \le \alpha \le 1$. Note that E_{pp} and E_A are defined by eqns. 18 and 19 respectively.

Due to the dependency of the optimisation process on the trade-off parameter α, the GA search method was further enhanced for the design example by including α as the third variable. The search for the minima value, therefore, includes a range for prototype filter bandwidth π/Mp where Mp varies from 13 to 19 (for an 8 band UF bank), the roll-off factor 'r' varies in the range 0 to 2 and parameter α varies in the range 0.1 to 0.9. The GA optimised results are shown in shown in Table 1. These results were then used as starting 'seed' values for the Simplex method and the final optimised results obtained are shown in Table 2. The hybrid optimised results show a significant improvement over the directly optimised results for the Simplex and the quasi-Newton methods and a marginal improvement over the GA optimised results. The best optimised results taken from Table 2 generate coefficients of the prototype filter. These values were used to plot the frequency responses of the prototype filter and the eight analysis filters of the pseudo-QMF bank as shown in Figs. 4(a) and 4 (b) respectively. The linear and aliasing distortion responses are shown in Figs. 5 (a) and (b) respectively. The Simulink test error signal is shown in Fig. 6.

Table 1. Optimised results using GA with Mp range=13 to 19, 'r' range = 0 to 2 and alpha range= 0.1 to 0.9 for N=141.

α	E_{pp}	E_A	v_{rms}	out_{pp}
0.1045	0.0022	0.0002	0.0005	0.0029

Table 2. Optimised results using hybrid downhill Simplex method with seed values taken from GA results of Table 1 for N=141.

α	E_{pp}	E_A	v_{rms}	out_{pp}
0.1045	0.0015	0.0002	0.0004	0.0027

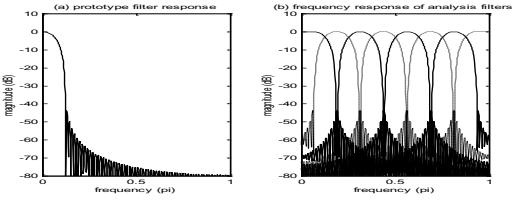


Fig. 4. Magnitude frequency response for N=141 (Table 2 optimized results): (a) prototype filter and (b) all channels

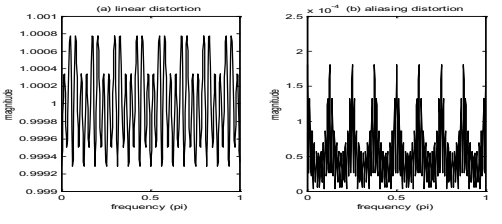


Fig. 5. Overall distortion for N=141 (Table 2 optimized results):(a)linear and (b)aliasing

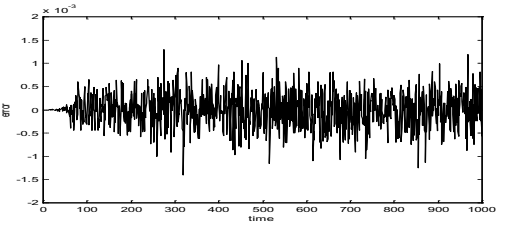


Fig. 6. Simulink test error signal for a random input with a uniform distribution [-1,1] for N=141 (Table 2 optimised results)

4.2 Case for NUF Bank

The NUF bank is derived from the cosine modulated uniform filter bank by using multiple prototypes p(n) and deriving the appropriate analysis and synthesis filters represented eqns 16 and 17. The optimisation process is then based upon using the prototype variables i.e. bandwidth $\pi/2M = \pi/Mp$ and the roll-off factor 'r' for each of the low pass FIR prototype filters to optimise the overall distortion of the filter bank. The main purpose of the optimisation process here is to minimise the objective function with the specific aim of minimising the overall magnitude and aliasing error so that a perfect reconstruction characteristic is closely met. The error objective function to be minimised and used in the optimisation process is given by

$$E=\max[20\log_{10}|FFT(y(n))|]-\min[20\log_{10}|FFT(y(n))|] \qquad (21)$$

where y(n) is the impulse response of the network of Fig. 1 when the input impulse applied is given by x(n)=1,0,0…..0.

The optimization process is identical to that used for the UF bank as shown in Fig. 3. Three design examples are considered here and all of these fall in the category of maximally decimated NUF banks. These are:

Ex. 1: A 3-band structure with $r_0=2$, $r_1=6$, $r_2=3$ and N=51.
Ex. 2: A 5-band structure with $r_0=4$, $r_1=12$, $r_2=6$, $r_3=6$, $r_4=3$ and N=45 – case 1.
Ex. 3: A 5-band structure with $r_0=8$, $r_1=8$, $r_2=4$, $r_3=4$, $r_4=4$ and N=45 – case 2.

All of the above design examples fall in the category of 'incompatible sets' i.e. at least one shifted copy of $X(e^{j\omega})$ does not have a compatible pair at the output of another expander. In this case, complete elimination of aliasing is impossible for non-ideal filters. However, these examples can be resolved into a tree-structured form representing the uniform filter banks [4]. This property is useful in working out the overall aliasing error function given by eqn. 11.

The 3-band and 5-band NUF bank structures based on the values of r_k as shown above were optimized to derive the results as shown in Table 3. The corresponding frequency response of analysis filters, the distortions and the Simulink test error signals for design examples 1 and 2 are shown in Figs. 7 to 10.

Table 3. Optimized results for the 3-band NUF bank

Design Example	N	E_{pp}	E_A	v_{rms}	out_{pp}
3-band	51	0.0116	0.2568	0.0025	0.0193
5-band (case 1)	45	0.1051	0.1826	0.0172	0.1231
5-band (case 2)	45	0.0337	0.2717	0.0062	0.0446

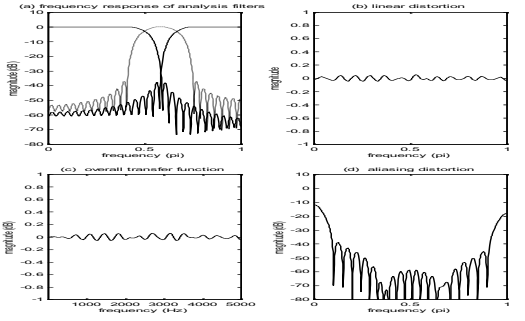


Fig. 7. For a 3-band NUF bank and N=51 (a) magnitude frequency response of all analysis filters (b) transfer function of the network assuming aliasing is zero (c) overall transfer function of the network of Fig. 1 (d) aliasing distortion.

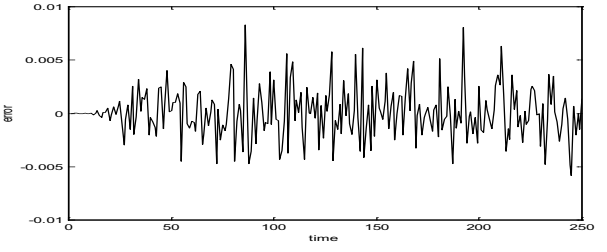


Fig. 8. Simulink test error signal for a random input with uniform distribution [-1,1] for the 3-band NUF bank using N=51

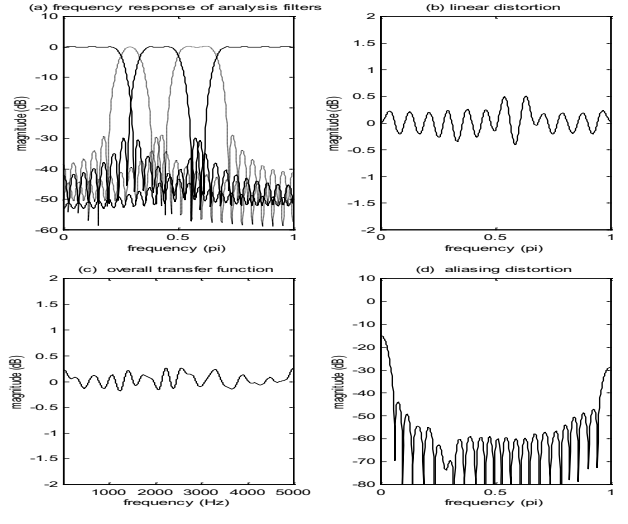


Fig. 9. For a 5-band NUF bank – case 1 and N=45 (a) magnitude frequency response of all analysis filters (b) transfer function of the network assuming aliasing is zero (c) overall transfer function of the network of Fig. 1 (d) aliasing distortion

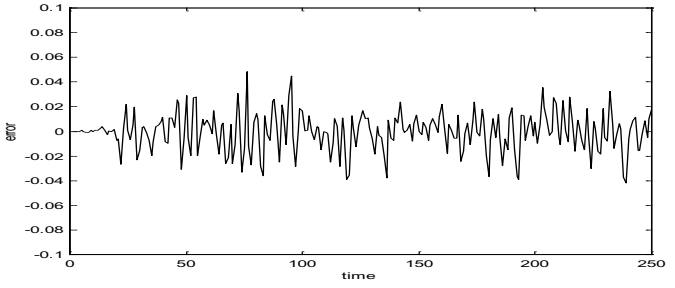


Fig. 10. Simulink test error signal for a random input with uniform distribution [-1,1] for the 5-band NUF bank – case 1 and N=45

5 Conclusions

A novel approach to the optimisation technique is proposed and investigated in this paper where an impulse (in the form of a dirac-delta function) is applied to the entire network of Fig. 1 and the Fourier Transform of the response is derived, giving the overall transfer function of the network. The optimisation process then generates an optimal overall transfer function. This method leads to the minimisation of the magnitude and the aliasing distortion in a combined manner. Two forms of maximally decimated M-channel multirate filter banks are considered here, these are; the uniform and non-uniform filter banks that are both maximally decimated.

For the case of NUF bank, a number of severe theoretical constraints for their design are relaxed in favour of a direct design approach based on the use of hybrid optimised (using GAs and a standard method) low pass FIR prototype filters. For both the UF and NUF banks, the design method is based on using a square root raised cosine FIR low pass prototype filter. The analysis and synthesis filters are derived from the prototype filters by the cosine modulation technique. The optimisation process is based on independently perturbing the roll-off factor and the bandwidth parameters of the prototype filters. Although the NUF banks considered in this study are restricted to the type using integer-valued decimators, this study can be easily extended to the case of NUF banks using rational valued decimators.

Several design examples have been considered to demonstrate the potency of the design and optimisation techniques developed in this study. The specific and some comparative results further substantiate this. The use of a hybrid optimisation process has demonstrated a strong synergy between the regulated random 'chance' based technique of genetic algorithms that work well over a wide landscape of possible discontinuous functions and a standard minimisation algorithm such as down-hill Simplex, that works well for largely continuous functions but on its own is rather susceptible to converging fairly rapidly to a local minima point depending on the selection of the initial 'seed' values.

References

[1] Baicher, G.S.: Real-time implementation of a class of quadrature mirror filter bank using Genetic Algorithms. In: IEEE International Conference on Bio-inspired Computing–Theory and Practice BIC-TA, Liverpool (2010)
[2] Vaidyanathan, P.P.: Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property. IEEE Trans. on Acoustics, Speech and Signal Processing ASSP-35, 476–492 (1987)
[3] Vaidyanathan, P.P.: Multirate systems and filter banks. Prentice-Hall (1993)
[4] Hoang, P.-Q., Vaidyanathan, P.P.: Non-uniform multirate filter banks: theory and design. In: Proc. IEEE Int. Symp. Circuits Syst., Portland, OR, pp. 371–374 (1989)
[5] Wada, S.: Design of non-uniform division multirate FIR filter banks. IEEE Trans. Ccts. and Sys. – II 42(2) (1995)
[6] Jeong-jin, L., Byeong, G.L.: A design of nonuniform co-sine modulated filter banks. IEEE Trans. on Ccts. and Sys. 42(11), 732–737 (1995)
[7] Argenti, F., Del Re, E.: Non-uniform filter banks based on a multi-prototype cosine modulation. In: Proc. IEEE ICASSP, Atlanta, GA, pp. 1511–1514 (1996)
[8] Fliege, N.J.: Multirate Digital Signal Processing. John Wiley and Sons, Chichester (1993)
[9] Chipperfield, A., Fleming, P., Pohlheim, H., Fonseca, C.: Genetic Algorithm Toolbox for use with MATLAB, ver. 1.2. Department of Automatic Control Systems Engineering, University of Sheffield, UK (1993)
[10] Baicher, G.S.: Optimal Design of a class of M-Channel Uniform Filter Bank using Genetic Algorithms. In: IEEE International Conference on Signal Processing and Communications, Dubai (2007)

Topology Control in Wireless Ad Hoc Networks

Anil Yadav[1], Raghuraj Singh[2], and Rama Shankar[1]

[1] Deptt. of Computer Science & Engg., Motilal Nehru National Institute of Technology, India
[2] Deptt. of Computer Science & Engg., Harcourt Butler Technological Institute, Kanpur, India
yadanil@gmail.com, rscse@rediffmail.com, rsy@mnnit.ac.in

Abstract. W*ireless ad hoc network* enable new and exciting applications, such as decision making in the battlefield, emergency, search-and-rescue operations, but also pose significant technical challenges. Topology control problems are concerned with the assignment of power values to the nodes of an ad hoc network so that the power assignment leads to a graph topology satisfying some specified properties. This paper considers such problems under several optimization objectives, including minimizing the maximum power and minimizing the total power. In this paper we give a brief overview of topology (arrangement of devices) control problem in wireless ad hoc networks, up to some extent. Given a set of nodes in a 2-D plane, end-to-end traffic load (i.e amount of data to be transmitted between mobile hosts) and delay between a pairs of node, the problem is to find a network topology that can meet the optimal QoS requirements. We can solve the problem, by formulating as a linear programming problem for the traffic loads. and an optimal solution has been proposed to solve the problem.

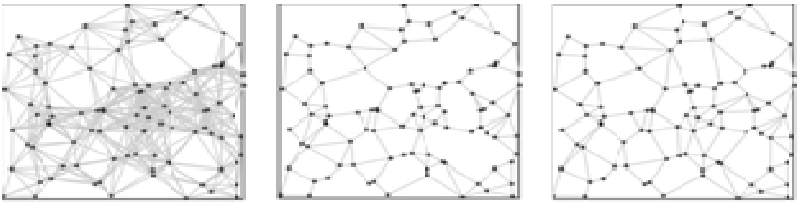
Keywords: Wireless Ad hoc network, QoS, topology, Energy management.

1 Introduction

What is a topology of mobile ad hoc network (MANET)?
A topology consists of a set of nodes and a set of links, describes the connectivity information of a wireless network. Unlike the links in a wire-network topology, which are defined by modems and cables, the links in a MANET topology are usually the result of some link determination protocol subject to a transmission power control. A basic requirement for an edge e*(u, v)* to be in the network topology of graph G is that node u and node v are within each other's transmission range[1].Topology may be defined as: *The art of coordinating nodes decisions regarding their transmitting ranges inorder to gurantee a network lifetime with desired properties.*

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 907–918.
springerlink.com

Most of the network topologies for wireless ad hoc networks are defined by connecting all neighbors within a fixed transmission range. Figure1 shows an example of a 100 node network topology with poission distributed nodes within a 100 x 100 square and with Poisson parameter X = 0.01 nodes/unit area, and a fixed transmission radius *R* of 30, with homogenous in Figure(a) and non-homogeneous in Figure(b), and (c). In order to maintain good network reliability, the fixed transmission radius must be adjusted to keep good connectivity even in a sparse region. Therefore, nodes in dense regions have high nodal degree, and more traffic per unit area is generated and delivered in these regions than in sparse regions. But there is much less throughput available for each node and more interference to packet transmission. These problems associated with dense regions can easily cause performance degradation even congestion in the network.



Figure(a) Figure(b) Figure(c)

Fig. 1. A simple topologies with 100 node, 2-D, poission-distribution random network in 100 x 100 square, R = 30,and X=0.01

One of the central challenges, ad hoc network topology control, is how to adapt to the dynamics of mobile nodes. Any centralized algorithm is vulnerable at the central controller and takes too much time to disseminate update information to nodes and their neighbors. Wireless ad hoc network has applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in inhospitable terrain, etc. It is featured by dynamic topology (infrastructure less), multi-hop communication, limited resources (bandwidth, CPU, battery, etc) and limited security. These characteristics put special challenges in routing protocol design. In multi-hop ad hoc networks, communication between two nodes that are not direct neighbors requires the relay of messages by the intermediate nodes to reach the final destination. Each node acts as a router, as well as a communication end-point. There are many modern network applications that require QoS provisions in ad hoc networks, such as transmission of multimedia data, and interactive distributed applications. Ad-hoc network is dynamic in the sense that new nodes may be added, existing nodes may be deleted or nodes moves around at any speed freely, whenever necessary. Thus the network topology changes dynamically, i.e. the routing protocols in multi hoc ad-hoc a network need to adapt quickly to the topology changes.

The topology of an ad hoc network depends on both an uncontrollable factors such as mobility, terrain and fading etc. and controllable parameters such as, transmit power, processing gain and the directions of antenna [2, 3, 4, 5, 26]. Topology control is to allow each node in the network to adjust its transmitting

power (i.e., to determine its neighbors) so that a more optimal network topology can be formed. Wireless devices/nodes are normally operated by batteries only and they have limited memory. A transmission by a wireless device is often received by many nodes within its vicinity, which possibly causes signal interferences at these neighboring nodes. An issue associated with topology control is often energy management. Since the capacity of battery power is very much limited, energy consumption is a major concern in topology control. To increase the lifetime of such networks and maintain network connectivity, it is more challenging to design a network topology for wireless ad hoc network that is suitable for designing an efficient routing scheme to save power and storage memory consumption than for the traditional wired networks [6].The primary motivation for studying topology control problems is to make efficient use of the available power at each node. In this paper, we address the need for energy efficient topology control problem. Accordingly, the main contributions of this paper are as follows:

1. For a given a set of wireless nodes in a 2-D plane, our objective is to find a network topology that can minimize the maximum transmitting power of nodes through linear programming problem(LPP).
2. A general framework for mobility-sensitive topology control in MANETs.
3. Simulations were performed and results with different parameters are presented.

Rest of the paper is organized as follows. Section 2 describes the previous works on topology control in ad hoc network. Formulation of topology and system model is illustrated in Section 3. Topology control for both splittable and non splittable are describes in section 4. In section 5 simulation setup & results are presented and in section 6 ends this paper with conclusion and some future research work.

2 Previous Work

In this section of the paper we, reviews existing topology control schemes, especially localized schemes.

Hu in [1] describes a distributed, Delaunay triangulation-based algorithm for choosing logical links and as a consequence carrying out reliable topology control. In choosing these links he follows a few heuristic guidelines such as not exceeding an upper bound on the degree of each node and choosing links that create a regular and uniform graph structure. He does not take advantage of adaptive transmission power control. The earlier work for topology control can be found in [5][6]. According to Hou et al. in [6] discuss the relationship between transmission range assignment and throughput. An analytic model was development to allow each node to adjust its transmitting power to reduce interference and hence achieve high throughput. Lloyd *et al. in* [7] suggested on the construction and maintenance of a network topology by three parameters namely G(M, P, O) where “M” represents the graph model either directed or undirected, “P” represents the graph property either 1-connected or 2-connected, and “O” represents the minimization objective

i.e min max power or min total power. Ramanthan and Rosales-Hain [8] describe a centralized spanning tree based algorithm for creating connected and bi-connected static networks with the objective of minimizing the maximum transmitting power for each node. Additionally, they describe two distributed heuristics, LINT (local information no topology) and LILT (local information link-state topology), adjust the node transmit power to maintain network connectivity in response to topological changes. Their reasoning and algorithms are based on simple heuristics and consequently do not guarantee network connectivity in all cases. In [9]Li *et al.* proposed building a local minimum spanning tree(MST) at each node to include its 1-hop neighbors and selecting neighbors in the MST as logical neighbors. This scheme guarantees topology control algorithm that achieves network connectivity with minimal power consumption. The above schemes can be enhanced to achieve multiple desirable properties such as low message overhead, constant stretch ratio, low weight, and minimal interference. A cone-based distributed topology (CBTC) control algorithm was developed in [10, 11], that increases network life time while maintaining global connectivity with reasonable throughput in a multi hop wireless ad hoc network. Network lifetime is increased by determining the minimal operational power requirement for each node in the network while guarantying the same maximum connected node set as when all nodes are transmitting with full power. Several optimizations are also proposed to further reduce the number of logical neighbors and transmission range. Furthermore, CBTC requires only direction information instead of accurate location information. Huang *et al.* extended this work in [12] by using directional antennas. Marsan *et al.* presented a technique in [13] to optimize the topology of Bluetooth, by minimizing the maximum traffic load of nodes.

Rodoplu and Meng in [14] proposed another method of reducing the number of edges while maintaining network connectivity and in addition, preserving all minimum energy paths. A minimum-energy path between two nodes u and v is defined as the shortest path between u and v, using transmission power as edge cost. An edge (u,v) can be removed if there exists another node w such that 2-hop path (u,w,v) consumes less energy than direct transmission. In [15] Singh *et al.* focused on various metrics of energy efficient routing, such as minimizing energy consumed per packet, minimizing cost per packet, and so on. Kawadia *et al. in* [16] proposed a clustering method for routing in ad hoc networks.The goal is to choose the transmit power level, so that low power levels can be used for intra-cluster communication and high power levels for inter-clusters. In [17],Wieselthier *et al.* studied the problem to make efficient use of available power of each node, such that the total energy cost of a broadcast/multicast tree is minimized. Some heuristic algorithms were proposed, namely, MST (minimum spanning tree), and SPT (shortest-path tree). The proposed algorithms were evaluated through simulations. Wan *et al.* in [18] presented a quantitative analysis of performances of these three heuristics.

3 Formulation of Topology

Topology control problems have been studied under two graph models. First is directed graph model and second is undirected graph model. For the directed graph model, for any pair of nodes *u* and *v*, whenever the directed edges (*u, v*) and (*v, u*) are present, this pair of directed edges is replaced by a single undirected edge *{u,v}*.

Definition1. The node connectivity of an undirected graph is the smallest number of nodes that must be deleted from the graph so that the resulting graph is disconnected.
Definition2. The edge connectivity of an undirected graph is the smallest number of edges that must be deleted from the graph so that the resulting graph is disconnected.
For example, a tree has node and edge connectivities equal to 1 while a simple cycle has node and edge connectivities equal to 2. When the node (edge) connectivity of a graph is greater than or equal to *k*, the graph is said to be *k*-node connected (*k*-edge connected).

3.1 System Model

In multi hop environment, we consider the following transmitting power model: $tp_{i,j} = d_{i,j}^{\alpha}$, where $tp_{i,j}$ is the transmitting power required for node *i* to reach node *j*, d_{ij}, is the distance between *i* and *j*, and α is a parameter typically having value between 2 and 4.

The wireless ad hoc network can be represented by $G = (V, E)$, where *V* is the set of vertices (nodes) and *E* a set of undirected edges (links). Each node has a bandwidth capacity *B*, and a maximal transmitting power level *P*. Let p_i denote the transmitting power of node *i*. We assume that each node can adjust its power level, but not beyond some maximum power *P* i.e $0 \le tp_i \le P$ for $1 \le i \le n$. The network connectivity between two nodes depends on their transmission range and power. An edge $e(i,j) \in E$ iff $tp_i = d_{i,j}^{\alpha}, and \;\; tp_j = d_{i,j}^{\alpha}$.

Let us consider $\lambda_{s,d}$ and $\Delta_{s,d}$ be the traffic load and the maximally allowed hop-count for node pair (*s*, *d*), respectively. Let $P_{max} = \max\{ p_i \mid 1 \le i \le n \}$. The topology control problem can be defined as: *V is the set of node with their locations,* $\lambda_{s,d}$ *and* $\Delta_{s,d}$ *for node pair (s, d), find transmitting power* p_i *for* $1 \le i \le n$*, such that all the traffic load can be routed within the hop-count bound, and* P_{max} *is minimized.* We consider two cases: First end-to-end traffic load are non-splittable, i.e., $\lambda_{s,d}$ for node pair (*s*, *d*) must be routed on the same path from *s* to *d*; Second end-to end traffic load are splittable, i.e., $\lambda_{s,d}$ can be routed on several different paths from *s* to *d*. We assume that each node can transmit signals to its neighbors in a conflict free manner. There are many medium access control layer protocols [19, 20] or code assignment protocols [21, 22] that have been proposed to reduce signal interference in radio transmissions.

4 Topology Control

In this section, we have consider the variables and equations as suggested by *X.Jia et al.* in [10, 24, 25]. Let us consider graph G(V, E),where
V =Set of nodes and their locations
E = Set of bi-directional edges(links)
B = Bandwidth of each node
$\lambda_{s,d}$ =Traffic demands for each node pair (s,d)
$\Delta_{s,d}$=Maximally allowed hop-count for node pair (s,d)ï
P = Maximally allowed transmitting power of nodes
Now, we consider the two cases of topology control problem one is with load non-splittable and in other load is splittable.

4.1 Topology Control With Traffic Load Non-splittable

According to *X.Jia et al.* in [10, 24], following variable and equations are considered.
Boolean Variables:

$$x_{i,j} = \begin{cases} 1 & if\ there\ there\ is\ a\ link\ \ from\ \ node\ \ i\ to\ node\ \ j \\ 0 & otherwise \end{cases}$$

$$x_{i,j}^{s,d} = \begin{cases} 1 & if\ the\ routes from s to d\ goes through link(i,j) \\ 0 & otherwise \end{cases}$$

P_{max} = The maximum transmitting power of nodes.

Optimization:
Minimize the maximum transmitting power of nodes i.e Min P_{max}
Constraints:
Here, we consider topology constraints, transmitting power constraints, delay constraints, bandwidth constraints are in [24].
The problem of topology control for non-splittable has now been formulated as a linear programming problem (LPP). We has used Matlab8.2 to solve the problem for experimental purpose.

4.2 Topology Control with Traffic Load Splittable

During the flow of the network, the traffics load between a node-pair may take different routes due to congestion or failures in the network. In this part, we consider the case that the traffic load can be split.

(A) Problem Formulation

Additional Variables:

$f_{i,j}^{s,d}$, representing the load of node pair (s,d) that go through link (i,j)

Optimize:

Minimize the maximum power of nodes

$$\text{Min } P_{max} \tag{1}$$

Constraints: Topology constraints and transmitting power remains the same.
Bandwidth constraint:

$$\sum_{(s,d)} \sum_{j} f_{i,j}^{s,d} + \sum_{(s,d)} \sum_{j} f_{j,i}^{s,d} \le B \quad \forall i \in V \tag{2}$$

Route constraints:

$$\sum_{j} f_{i,j}^{s,d} - \sum_{j} f_{j,i}^{s,d} = \begin{cases} \hat{\lambda}_{s,d} & if\ s = i \\ -\hat{\lambda}_{s,d} & if\ d = i \forall i \in V \\ 0 & otherwise \end{cases} \tag{3}$$

$$f_{i,j}^{s,d} \le f_{i,j}^{s,d}{}_{i,j} \quad \forall i, j \in V, (s,d) \tag{4}$$

Variable Constrainst:

$$x_{i,j} = 0, or\ 1, f_{i,j}^{s,d} \ge 0 \quad \forall i, j \in V, (s,d) \tag{5}$$

We consider the following points:
Constraint (3) is for flow conservation along all the routes for node pair (*s*, *d*).There is no delay constraint in the above formulation, because the traffic between a node pair can be routed via several different paths. The topology control problem with traffic splittable has now been formulated as a linear programming problem in equation (1) – (5).

(B) Proposed Solution

The goal is to find the network topology when that all traffic load can be routed and the maximal node power is minimized. In the case where traffic load are splittable, the solution consists of two steps. First Increment the node energy to connect two nodes that have the shortest distance among the unconnected node pairs. Second**:** Check if the traffic load can be routed on the topology generated in step 1. If so, the topology is found otherwise repeat step first and second. Since the traffics load are splittable, the problem in step 2 can be transformed to a variant of

the multi-commodity flow problem, i.e for a given network topology, to route commodities on the network such that the maximal load of nodes is minimized. First, we first tackle the QoS routing issue for a given network topology.

(i)QoS Routing Issue

Given a network graph *G(V,E)* with traffic load between node pairs, route this traffic load in the graph, in such a way that the maximum node-load in the system, is minimized (i.e MinL_{max},). The problem can be formulated as the following:

$$\text{Min } L_{max} \tag{6}$$

$$\sum_j f_{i,j}^{s,d} - \sum_j f_{j,i}^{s,d} = \begin{cases} \lambda_{s,d} & if\ s = i \\ -\lambda_{s,d} & if\ d = i \forall i \in V \\ 0 & otherwise \end{cases} \tag{7}$$

$$\sum_{(s,d)} \sum_j x_{i,j}^{s,d} + \sum_{(s,d)} \sum_j f_{j,i}^{s,d} \le L_{\max} \qquad \forall i \in V \tag{8}$$

$$\begin{aligned} & f_{i,j}^{s,d} \ge 0 \quad \forall i,j \in V,(s,d) \\ & L_{\max} \ge 0 \end{aligned} \tag{9}$$

$$\sum_j f_{i,j}^{s,d} - \sum_j f_{j,i}^{s,d} = 0 \tag{10}$$

$$Note\ that\ \forall (s,d), f_{i,j}^{s,d} = 0, if\ (i,j) \in E(G)$$

Equation (6) is the objective function, which is to minimize the maximum node load. Equation (7) shows receiver flow conservation requires that the difference between the flow coming and leaving a receiver due to a flow between (s, d) should be (i) zero if the node is not the source or the destination.(ii) $\lambda_{s,d}$ if the node is the source.(iii) - $\lambda_{s,d}$ if the node is the destination. Equation (9) obtains the maximum node load in the network. When any request cannot be routed due to the disconnection of the network, it will report an error of disconnection. Equation (10) shows the transmitter flow conservation constraints require that all flow coming and leaving a transmitter due to flow between (s, d) should be equivalent.

This is a linear programming (LPP) problem. The optimal solution can be found in polynomial time $O((|E|t)3.5)$, where $|E|$ is the number of edges in graph G, and t is the number of node pairs which have non-zero traffic[23]. We use Matlab8.2 to compute the LPP.

(ii)Energy Efficient QoS Topology control algorithm

The idea of the algorithm is to sort all node pairs in non-decreasing order according to their Euclidean distance. Each time the pair of nodes that have the shortest distance and have not yet had a link between the two nodes are picked and their power is increased until they can reach each other. Then, the QoS routing algorithm runs on the network to see if the requested traffics can be all routed. This operation is repeated until the QoS topology is found, or all nodes already reached their maximal power.

Input: A set of node V with their locations, $\lambda_{s,d}$ for nodepair (s,d) and bandwidth B
Output: Transmitting power levels p for all nodes in V.

(a)Sort all node-pairs in non-decreasing order with $d_{i,j}^{\infty} \leq P(i < j)$ and $d_{i,j}$.

(b)Pick up the next node-pair with closest distance but not yet connected and increase the power to make them connected to get a new graph G.

(c)Run the QoS routing algorithm on G to obtain $L_{\max}$. If $L_{\max} \leq B$, or there is no left link then stop; otherwise repeat (b) and (c).

In step (b), it stops if all nodes already reach power P and an error of no solution is reported in this case. To reduce the number of times of calling the QoS routing algorithm in step (c), we use the binary search method to find the QoS topology, instead of adding an edge each time and running the routing algorithm. In this algorithm, the node power is gradually increased until the required topology is formed. It is not difficult to see that the maximum node energy needed to form the required topology is minm, provided the number of power-levels of a node is finite.

5 Experiment

Here, we consider two widely adopted traffic model for MANET simulation analysis, the first is *Constant Bit Rate (CBR) Traffic Model* in which every source (sender) of traffic load generates a constant flow of packets and the second is *Variable Bit Rate (VBR) Traffic Model,* in which the traffic load sources generating a variable amount of data as function of time. Here, in this paper we, have used VBR traffic model for simulation purpose.

(A) *Simulation Setup*

The simulations are conducted in a 20×20, 2-D free-space region, in which the nodes are randomly and uniformly distributed. All nodes have the same bandwidth capacity $B = 500$. The value of α in the transmitting power function is set to 2, i.e., $tp_{i,j=} d_{i,j}^{\infty}$ $for \propto = 2$. Request(s) $R = \{s, d, \lambda_{s,d}\}$ are generated by using the Poisson distribution function. For each node, we use the random Poisson function with the mean value $\lambda = 1$ to generate a number k, which is the number of requests originating from this node. The load $\lambda_{s,d}$ for a pair of nodes (s, d) is assigned by a

random function of a normal distribution with variance equal to $0.5\lambda_m$, where λ_m is the mean value of the normal distribution function.

(B) *Simulation Results and analysis*

Fig.1 shows the topology of a network with 4 nodes and 4 requests with traffic non-splittable. The source, destination, and traffic demand of the 4 requests are generated as described in subsection 6 A. The average traffic load per request (i.e., λm) is {(20x20) /4=100}, which is $0.2B$. The details of the requests and the computed routing information are listed in Table1. For comparison, Figure 2 is the topology for the case where traffics are considered splittable and the topology is computed by using the method described in section V. The input for Figure2 is the same as Figure1. Table2 shows the routing and traffic distribution in the topology of Figure2. Comparing Figure1 with Figure2, we can see that the topology for non-splittable case has a long distance edge in Figure1, which results in a much higher $P_{\max}$ than the splittable case. The topology for splittable case has more short edges, which helps splitting the traffic off among multiple routes. Note that: Figure1 and Figure2 are topologies which remove the redundant edges, i.e., no load edges.

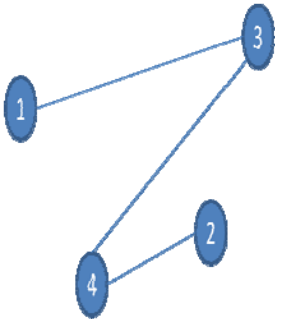


Fig. 2. Topology of four nodes and four fours requests for non-splittable traffic load

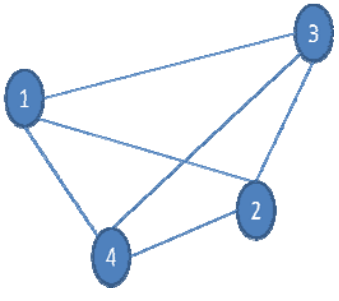


Fig. 3. Topology of four nodes and requests for splittable traffic load

Table 1. The Qos Requests and Their routes for Figure2

Source(S)	Destination(D)	Traffic Demand($\lambda_{s,d)}$	Route
2	4	97.4603	2 → 4
3	4	93.2602	3 → 4
1	3	101.0183	1 → 3
3	2	95.8970	3→ 4→ 2

Table 2. The QoS Requests and Their routes for Figure.3

Source (S)	Destination (D)	Load Demand($\lambda_{s,d)}$	Route
2	4	88.0611 4.9452 1.9232	2 → 4 2→1→4 2→3→4
3	4	93..2602	3 → 4
1	3	60.2726 36.8259	1→ 2→ 3 1→ 4→ 3
3	2	65.8970 27.3726 18.7683	3 →2 3 →1→ 2 3 →4→ 2

6 Conclusion

In this paper we examine the problem of topology control problem. Both cases of traffic load splittable and traffic load non-splittable have been considered. For the former case, we considered both bandwidth and delay parameters as basic requirements. For the latter case, we considered only the single parameter i.e bandwidth requirement. For both the cases the problem has been formulated as a linear programming problem (LPP). A polynomial time algorithm has been proposed to compute the optimal solution. The problem discussed is a static configuration problem. The traffic demands are assumed to be known in prior. By configuring a optimal QoS topology, QoS requests can be best served in the system. However, due to the dynamics and the unpredictability of network traffics load , a QoS request can still be blocked no matter how good the topology is. In a dynamic environment where nodes are mobile up to some extent and traffic load are dynamic, the proposed topology control algorithm can be run periodically to keep the topology optimal in the sense that it balances the node.

References

[1] Hu, L.: Topology control for multihop packet radio networks. IEEE Trans. On Communications 41(10), 1474–1481 (1983)
[2] Zhu, C., Corson, M.S.: QoS routing for mobile ad hoc net-works. In: IEEE INFOCOM (2002)
[3] Lin, C.R., Liu, J.S.: QoS routing in ad hoc wireless networks. IEEE Journal on Selected Areas in Communications 17(8), 1426–1438 (1999)
[4] Yorozu, Y., Hirano, M., Oka, K., Tagawa, Y.: Electron spectroscopy studies on magneto-optical media and plastic substrate interface. IEEE Trans. J. Magn. Japan 2, 740–741 (1987)
[5] Lin, C.R.: Admission control in time-slotted multihop mobile networks. IEEE Journal on Selected Areas in Communications 19(10), 1974–1983 (2001)

[6] Hou, T., Li, V.O.K.: Transmission range control in multihop packet radio net Works. IEEE Trans. on Communications 34(1), 38–44 (1986)
[7] Lloyd, E.L., Liu, R., Marathe, M.V., Ramanathan, R., Ravi, S.S.: Algorithmic aspects of topology control problems for ad hoc networks. In: ACM MOBIHOC (2002)
[8] Ramanathan, R., Rosales-Hain, R.: Topology control of multihop wireless networks using transmit power adjustment. IEEE Transaction on Communications, 404–413 (2000)
[9] Li, N., Hou, J., Sha, L.: Design and analysis of an MST- based topology control Algorithm. In: IEEE INFOCOM (2003)
[10] Jia, X., Li, D., Du, D.: Qos Topology control in Ad Hoc Wireless Network. In: IEEE INFOCOM (2004)
[11] Wattenhofer, R., Li, L., Bahl, P., Wang, Y.M.: Distributed topology control for power efficient operation in multihop wirerless ad hoc networs. In: IEEE INFOCOM, vol. 3, pp. 1388–1397 (2001)
[12] Huang, Z., Shen, C.C., Scrisathapornphat, C., Jaikaeo, C.: Topology control for ad hoc net works with directional antennas. In: IEEE 11th Conf. on Computer Communications and Networks, Miami, pp. 16–21 (2002)
[13] Marsan, M.A., Chiasserini, C.F., Nucci, A., Carello, G., Giovanni, L.D.: Optimizing the topology of Bluetooth wireless personal area networks. In: IEEE INFOCOM (2002)
[14] Rodoplu, V., Meng, T.H.: Minimum energy mobile wireless networks. IEEE J. Selected Areas in Communications 17(8), 1333–1344 (1999)
[15] Singh, S., Woo, M., Raghavendra, C.S.: Power-aware routing in mobile ad Hoc networks. In: ACM MOBICOM, Dallas, pp. 181–190 (1998)
[16] Kawadia, V., Kumar, P.R.: Power control and clustering in ad hoc networks. In: IEEE INFOCOM (2003)
[17] Wieselthier, E., Nguyen, G.D., Ephremides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: IEEE INFOCOM (2000)
[18] Wan, P.J., Calinescu, G., Li, X.Y., Frieder, O.: Minimum-energy broadcast routing in static ad hoc wireless networks. In: IEEE INFOCOM (2001)
[19] Muqattash, A., Krunz, M.: CDMA-based MAC protocol for wireless ad hoc net Works. In: ACM MobiHoc, pp. 153–164 (2003)
[20] Rodoplu, V., Meng, T.: Position based CDMA with multiuser diction (P-CDMA/MUD)for wireless ad hoc networks. In: IEEE 6th Int'l Symp. on Spread Spectrum Techniques and Applications, vol. 1, pp. 336–340 (2000)
[21] Sousa, E., Silvester, J.A.: Spreading code protocols for distributed spread-spectrum packet radio networks. IEEE Transaction on Communications 36(3), 272–281 (1988)
[22] Hu, L.: Distributed code assignments for CDMA packet radio networks. IEEE/ACM Transaction on Networking, 668–677 (December 1993)
[23] Schrijver, A.: Theory of linear and integer programming. John Wiley & Sons (1986)
[24] Jia, X., Li, D., Du, D.: QoS topology control in ad hoc wireless networks. In: Proc. of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Hong Kong, vol. 2, pp. 1264–1272 (March 2004)
[25] Li, D., Jia, X., Du, D.: QoS topology control for non homogeneous Ad Hoc wireless Networks. EURASIP Journal on Wireless Communications and Networking 1, 1–10 (2006)
[26] Cabrera, J.B.D., Ramanathan, R., Gutierrez, C., Mehra, R.K.: Stable Topology control of multihop wireless network using transmit power adjustment. IEEE Communications Letters 11(7) (July 2007)

A Survey of Multi-index Transportation Problems and Its Variants with Crisp and Fuzzy Parameters

Akhilesh Kumar[1] and Shiv Prasad Yadav[2]

[1] Department of Mathematics, ASET, Amity University Haryana, Manesar, Gurgaon, India
akumar@ggn.amity.edu, akhilesh.maths@gmail.com
[2] Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India
yadavfma@iitr.ernet.in

Abstract. The Multi-index transportation problems are of immense use in present scenario. This paper provides a survey of the work done in the area of such problems from 1955 till date. The purpose of this survey is to provide the reader an up to date account of such problems and their current variants, many of which consider the fuzziness in objective function, constraints and/or coefficients. The survey also studies current approaches to solve such problems.

Keywords: Multi-index, Transportation Problem, Multi-commodity, Fuzziness, Genetic Algorithm.

1 Introduction

Classical transportation problem is the task of computing the minimum cost required to transport a certain type of commodity from a set of sources to a set of destinations.

In the industry normally two or more goods are manufactured at a plant and transported to various destinations. This is done to maximize the profit. For this reason the two-dimensional classical transportation problem is extended to multi-index transportation problem.

1.1 Historical Aspect

The Multi-index Transportation Problems (MITPs) were initially considered by Schell [18] and Galler & Dwyer [7]. Later on the multi-commodity transportation problem (also called Solid Transportation Problem), which is closely related with the MITP, was introduced by Haley [9]. Further an algorithm on the lines of Modi-method was proposed and illustrated with an example by him. Haley [10] justified the use of this algorithm with the help of theorems and gave a necessary condition for the existence of a solution.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 919–932.
springerlink.com

1.2 Mathematical Formulation of the Problem with Three Indices (with Three Planer Sums)

Determine $x_{ijk} \geq 0$ for all *i, j, k* which minimize

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} c_{ijk}x_{ijk} \tag{1}$$

subject to the constraints

$$\sum_{i=1}^{m} x_{ijk} = A_{jk}, \qquad \sum_{j=1}^{n} x_{ijk} = B_{ki}, \sum_{k=1}^{p} x_{ijk} = E_{ij} \tag{2}$$

where

$$\sum_{j=1}^{n} A_{jk} = \sum_{i=1}^{m} B_{ki}\,, \qquad \sum_{k=1}^{p} A_{jk} = \sum_{i=1}^{m} E_{ij}\,, \qquad \sum_{k=1}^{p} B_{ki} = \sum_{j=1}^{n} E_{ij} \tag{3}$$

and

$$\sum_{j=1}^{n}\sum_{k=1}^{p} A_{jk} = \sum_{k=1}^{p}\sum_{i=1}^{m} B_{kj} = \sum_{i=1}^{m}\sum_{j=1}^{n} E_{ij} \tag{4}$$

The three-dimensional multi-index transportation problem (MITP) can be visualized as a block of *mnp* cells with each cell representing x_{ijk} for $i = 1, \ldots, n$; $j = 1, \ldots, m$; $k = 1, \ldots, p$ when summed along rows, columns and heights give A_{jk}, B_{ki} and E_{ij} respectively, representing above equations (2), (3) and (4).

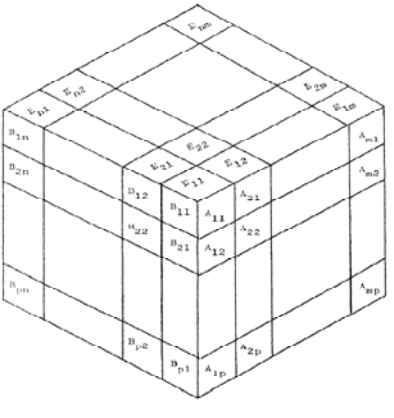


Fig. 1.

For a basic feasible solution of the above problem $mnp - (m - 1)(n - 1)(p - 1)$ values of x_{ijk}'s must be non-zero. This paper overviews the work done on multi-index transportation problems (MITPs) with crisp as well as fuzzy parameters. The survey is divided into two sections. In section II, we provide a survey of recent work done for developing various methods for solving multi-index transportation problems (MITPs) with crisp parameters. In section III, we discuss the work done by various approaches for solving the MITPs with fuzzy objective functions, constraints and coefficients.

2 Algorithms for Solving MITPs and Its Extensions with Crisp Parameters

Korsnikov [14] described a class of planar three-index transportation problems which can be reduced to classical transportation problems and illustrated with an example of a polytope with full dimension having this property.

Vignaux and Michalewicz [20] applied Genetic Algorithm (GA) approach to the classical transportation problem in two patterns. In the first approach, as they expressed the allocation matrix of the TP as single dimensional array (vector), so inversion could also be used along with mutation and crossover as Genetic Operators. In the second approach, as the population of allocation matrices was expressed in its natural way, therefore instead of applying inversion operation mutation and crossover only were used. While generating offspring by mutation submatrices of allocation matrices are altered such that row sums and column sums remain the same. And while generating offspring by crossover from two parent matrices say $V_1 = (\boldsymbol{v^1_{ij}})$ and $V_2 = (\boldsymbol{v^2_{ij}})$ two temporary matrices $DIV = (div_{ij})$ and $REM = (rem_{ij})$ were created as follows:

$$div_{ij} = \left\lfloor \left(v^1_{ij} + v^2_{ij}\right)/2 \right\rfloor \; and \; rem_{ij} = \left(v^1_{ij} + v^2_{ij}\right) mod \; 2 \tag{5}$$

REM matrix was then transformed into two matrices REM_1 and REM_2 such that

$$REM = REM_1 + REM_2, \tag{6}$$

with row sums and column sums of REM_1 and REM_2 as half of those of *REM* respectively. The two offsprings of V_1 and V_2 were then produced as

$$V_3 = DIV + REM_1 \; and \; V_4 = DIV + REM_2 \tag{7}$$

The results of runs for 100 generations were compared for both approaches and was concluded that the matrix approach is more efficient than the vector approach.

Li, Ida, Gen and Kobuchi [16] transformed the original multi-criteria MITP into a single objective problem by global criteria method. Then to solve the single-objective MITP thus obtained, they proposed neural network architecture according to augmented Lagrange multiplier method. Even larger problems can be

efficiently solved for optimal solution, due to the massive computing unit-neurons and parallel mechanism of neural network approach.

Ahuja and Arora [1] defined Multi-index Fixed Charge Bi-criteria Transportation Problem. They considered time & cost as the two criteria. They presented an algorithm to find efficient cost-time trade-off pairs for the problem by first formulating corresponding fixed charge MITP. The procedure was illustrated by working out an example.

Zitouni & Keraghel [21] dealt with the capacitated MITP with four indices by establishing an existence criterion, an optimality condition and an algorithm of resolution.

Bulut & Bulut [4] discussed axial and planer four-index transportation problem of order $1\times m\times n\times 1$ and proved it to be equivalent to a circularization network flow problem with $m+n+2$ nodes and $(m+1)(n+1)$ arcs. By giving algebraic characterization of each of these problems, they proved that they have common characterizations and can be solved in terms of eigenvectors of the matrices J_m and J_n (J_m is an $m\times m$ matrix whose entries are 1).

Zitouni, Keraghel and Benterki [22] concentrated on the numerical testing of the already developed algorithm with computer program written in Delphi 4. The tests carried out on different examples and the solutions obtained concluded that the algorithm is stable, robust and requires few iterations and time to achieve optimal solution.

3 Fuzzy TPs and MITPs and their Extensions and Algorithms for their Solutions

Gen et. al. [8] applied criteria space approach for bicriteria linear program conductive to find out the non dominated extreme points in the criteria space of Bicriteria solid transportation problem. They followed the Michalewicz's idea of representing the allocation matrix as chromosomes and initialization of the initial population. After obtaining initial solution-matrices X_s ($s = 1, 2, \ldots,$ *population size*), they defined the evaluation function as:

$$eval(X_s) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{t} C_{ijk}x_{ijk} \tag{8}$$

where, $C_{ijk} = \frac{\alpha}{\alpha+\beta}C_{\mathbf{1}ijk} + \frac{\beta}{\alpha+\beta}C_{\mathbf{2}ijk}$.

The values of α and β were obtained by $\alpha = |z_1 - z_{\mathbf{1}min}|$ and $\beta = |z_{\mathbf{2}min} - z_2|$, where $z_{\mathbf{1}min}$ and $z_{\mathbf{2}min}$ are taken from those pairs of (z_{1s}, z_{2s}) which give minimum values of their respective objective functions as $(z_{\mathbf{1}min}, z_2)$ and $(z_{\mathbf{1}}, z_{\mathbf{2}min})$. Crossover was also done on the lines of Michalewicz's [20] approach and selection of next generation used the elitist approach.

Cadenas & Jiménez [5] considered multi-objective STP with fuzzy goals. First they solved single objective STPs using each time only one objective (ignoring all others) and denoted their optimal solutions as $X^{1*} = \{x_{ijk}^1\}, X^{2*} = \{x_{ijk}^2\}, \ldots, X^{P*} = \{x_{ijk}^P\}$.

Then the pay off matrix of values of each objective functions at each of above obtaining solutions were calculated as follows.

$$\begin{array}{c} \\ Z_1 \\ Z_1 \\ . \\ . \\ . \\ Z_P \end{array} \begin{array}{c} \begin{array}{cccccc} X^{1*} & X^{2*} & . & . & . & X^{P*} \end{array} \\ \begin{bmatrix} Z_1(X^{1*}) & Z_1(X^{2*}) & . & . & . & Z_1(X^{P*}) \\ Z_2(X^{1*}) & Z_2(X^{2*}) & . & . & . & Z_2(X^{P*}) \\ . & . & . & & & \\ . & . & & . & & \\ . & . & & & . & \\ Z_P(X^{1*}) & Z_P(X^{2*}) & . & . & . & Z_P(X^{P*}) \end{bmatrix} \end{array} \tag{9}$$

For each of P objectives ($p = 1,2, \dots, P$), L_p = aspiration level of achievement & U_p= highest acceptable level of achievement and then the membership functions were calculated as follows.

For each $p = 1,2, \dots, P$:

$$L_p = Z_p(X^{p*}) \text{ and } U_p = \max\{Z_p(X^{1*}), Z_p(X^{2*}), \dots, Z_p(X^{P*})\} \tag{10}$$

$$\text{and} \qquad \mu(Z_p) = \begin{cases} 1 & if\ Z_p \le L_p \\ \frac{U_p - Z_p}{U_p - L_p} & if\ L_p \le Z_p \le U_p \\ 0 & if\ Z_p \ge U_p \end{cases}. \tag{11}$$

Then the fuzzy LPP as per the *additive weighted model* was formulated as

$$\max V(\mu) = \sum_{p=1}^{P} w_p . \mu_p(Z_p) \tag{12}$$

subject to the constraints

$$\sum_{j=1}^{n}\sum_{k=1}^{K} x_{ijk} = a_i\ (i \in I); \sum_{i=1}^{m}\sum_{k=1}^{K} x_{ijk} = b_j\ (j \in J); \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} = e_k\ (k \in K) \tag{13}$$

$$x_{ijk} \ge 0, \qquad for\ all\ i, j, k,$$

with $0 \le w_p \le 1$ $(p = 1, \dots, P)$ and $\sum_{p=1}^{P} w_p = 1$.

This was solves by Genetic Algorithm on the lines of technique proposed by Vignaux and Michalewicz [20].

Alves *et. al.* [2] proposed for the first time an approach to solve linear multi-commodity transportation problem with fuzzy objective function, fuzzy constraints and fuzzy coefficients. This approach was follow up of Zimmermann's method to turn the objective function into an inequality constraint, once the original linear programming problem has been dealt for fuzzy coefficients of constraints & the objective function. Hence the fuzzy linear multi-commodity transportation problem (FLMTP)

$$\widetilde{Min}\ Z = \sum_{k=1}^{K}\sum_{j=1}^{J} \tilde{c}_{kj}x_{kj} + \tilde{c}'_{kj}x'_{kj} \tag{14}$$

subject to

$$\tilde{A}_{x_k} \lesssim \tilde{r}_k \ \ (k = 1, \dots, K); \tag{15 a}$$
$$\tilde{A}'_{x_k} \lesssim \tilde{r}'_k \ \ (k = 1, \dots, K); \tag{15 b}$$

$$\sum_{k=1}^{K}\left(\tilde{D}_{kj}x_{kj}\right) \lesssim \tilde{b}_j \ \ (j = 1, \dots, J); \tag{16}$$

$$\sum_{j=1}^{J}(\tilde{T}_{kj}\tilde{F}_{kj}x_{kj} + \tilde{T}'_{kj}\tilde{F}'_{kj}x'_{kj}) \lesssim \tilde{f}_k \quad (k = 1, \dots, K) \tag{17}$$

$$x_{kj}, x'_{kj} \geq 0$$

can be expressed as

$$min\ \theta \tag{18}$$

subject to

$$\sum_{k=1}^{K}\sum_{j=1}^{J} c^o_{kj}x_{kj} + c'^o_{kj}x'_{kj} \leq r^o_0 + \theta t^o_0 \tag{19}$$

$$A^o{}_{x_k} \leq r^o{}_k + \theta t^o_k \ ; \qquad A^{o\prime}{}_{x_k} \leq r'^o_k + \theta t'^o_k; \ \ (k = 1, \dots, K) \tag{20 a \& b}$$

$$\sum_{k=1}^{K}\left(D^o_{kj}x_{kj}\right) \leq b^o_j + \theta t''^o_j \ \ (j = 1, \dots, J) \tag{21}$$

$$\sum_{j=1}^{J}(T^o_{kj}F^o_{kj}x_{kj} + T'^o{}_{kj}F'^o{}_{kj}x'{}_{kj}) \leq f^o_k + \theta t'''^o_k \quad (k = 1, \dots, K) \tag{22}$$

$$x_{kj}, x'_{kj} \geq 0\,, \qquad \theta \in [0, 1]$$

and

$$min\ \theta \tag{23}$$

subject to

$$\sum_{k=1}^{K}\sum_{j=1}^{J} \overline{c}_{kj}x_{kj} + \overline{c}'_{kj}x'_{kj} \leq \underline{r}_0 + \theta\underline{t}_0 \tag{24}$$

$$\overline{A}_{x_k} \leq \underline{r}_k + \theta\underline{t}_k; \qquad \overline{A}'_{x_k} \leq \underline{r}_k + \theta\underline{t}_k; \quad (k = 1, \dots, K) \tag{25 a \& b}$$

$$\sum_{k=1}^{K}\left(\overline{D}_{kj}x_{kj}\right) \le \underline{b}_j + \theta\underline{t''}_j \quad (j=1,\dots,J) \tag{26}$$

$$\sum_{j=1}^{J}(\overline{T}_{kj}\,\overline{F}_{kj}\,x_{kj} + \overline{T'}_{kj}\overline{F'}_{kj}x'_{kj}) \le \underline{f}_k + \theta\underline{t'''}_k \quad (k=1,\dots,K) \tag{27}$$

$$x_{kj}, x'_{kj} \ge 0\,, \qquad \theta \in [0,1]$$

where,

$$\tilde{c}_k = \left(c_k^o, \underline{c}_k, \overline{c}_k\right);\ \tilde{c'}_k = \left(c'^o_k, \underline{c'}_k, \overline{c'}_k\right);\ \tilde{r}_k = \left(r_k^o, \underline{r}_k, \overline{r}_k\right);\ \tilde{t}_k = \left(t_0^o, \underline{t}_0, \overline{t}_0\right)$$

etc. (other symbols having similar descriptions). $\tilde{t}_0, \tilde{t}_k, \tilde{t'}_k, \tilde{t''}_k, \tilde{t'''}_k$ are maximum tolerances.

Now as these are multi-commodity transportation problems therefore they used two of the specialised & basic techniques for this type of problem *viz,* partitioning and decomposition.

Jiménez & Verdegay [12] studied the interval multi-objective solid transportation problem, with general objective function (linear or non-linear) which can be stated as

$$Minimize\ Z_p = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{t} f_{ijk}^{p}\left(x_{ijk}\right), \qquad for\ p=1,\dots,P \tag{28}$$

subject to

$$\sum_{j=1}^{n}\sum_{k=1}^{l} x_{ijk} \in A_i, for\ i=1,\dots,m\ ; \quad \sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk} \in B_j, for\ j=1,\dots,n;$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} \in E_k, \qquad for\ k=1,\dots,l \tag{29 a, b \& c}$$

$$x_{ijk} \ge 0\ for\ all\ i,j,k.$$

This as per them is feasible if and only if $D=[d^1,d^2]=A\cap B\cap E\ \neq \emptyset$, where *A, B* and *E* are intervals

$$A = \sum_{i=1}^{m} A_i = \left[\sum_{i=1}^{m} a_i^1, \sum_{i=1}^{m} a_i^2\right] = [a^1, a^2] \tag{30}$$

$$B = \sum_{i=1}^{m} B_i = \left[\sum_{i=1}^{m} b_i^1, \sum_{i=1}^{m} b_i^2\right] = [b^1, b^2] \tag{31}$$

$$E = \sum_{i=1}^{m} E_i = \left[\sum_{i=1}^{m} e_i^1, \sum_{i=1}^{m} e_i^2\right] = [e^1, e^2] \tag{32}$$

by interval arithmetic.

Using weighted additive approach, they introduced a weight vector $\overline{w} = \{w_1, w_2, \dots, w_P\}$, $0 \le w_p \le 1,\ p = 1, \dots, P, \sum_{p=1}^{P} w_p = 1$ and hence combined multiple objective functions into one overall objective function to transform the problem into single objective interval STP. Single overall objective function as follows:

$$Minimize\ Z = \sum_{p=1}^{P} w_p \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{t} f_{ijk}^{p}(x_{ijk}) \tag{33}$$

To solve this interval STP they applied a non standard GA, which was proposed by Michalewicz for linear and non linear transportation problems. For this they gave two ways to create an initial solution. First procedure generates a solution which is at a vertex of the feasible polyhedron. The second procedure obtains a solution that can be an edge or a face of the feasible polyhedron. The fitness function is objective function of the chosen auxiliary problem. As genetic operators two types of mutations and an arithmetical crossover were used. In the second type of mutation, while recalculating the contents of the chosen sub-array, the second initialization procedure is used. The offspring of two parent feasible solutions V_1 and V_2 by arithmetical crossover were produced by the convex combinations of these parents as:

$V_3 = c_1V_1 + c_2V_2$ and $V_4 = c_2V_1 + c_1V_2$, where c_1& $c_2$are fixed randomly in each crossover such that $c_1 + c_2 = 1$ and $c_1 \ge 0, c_2 \ge 0$.

Li, *et. al.* [15] presented an evolutionary program to solve the fuzzy multi objective solid transportation problem in which the coefficients of objective function are taken as fuzzy numbers. By ranking fuzzy numbers with integral values and hence ranked values of fuzzy objectives, Pareto optimal solutions were determined with the help of GA. The best compromise solution was then obtained with the combination of Technique of Order Preference by Similarity to Ideal Solution (TOPSIS) to GA.

Tzeng *et. al.* [19] studied the problem of coal allocation system of Taipower which was having two objectives of (a) minimizing the cost of freight for importing coal and (b) maximizing satisfaction level of overall schedule pattern. This problem was having 23 sources of 4 qualities of coal to be transported to 3 destinations using 3 types of vessels available. The annual amount of supply of sources and monthly demand of each of 3 power plants were fuzzy numbers. Further this had a restriction that one of the unloading ports couldn't accept 2 of the 3 types of vessels. The problem was formulated as fuzzy multi-criteria multi-index transportation problem with 5 indices. Using reducing index method and interactive fuzzy multi-objective linear programming methods to locate the solution, a procedure of solving the problem was proposed.

Jimenez & Verdegay [13] considered the Fuzzy Solid Transportation Problem (FSTP) where objective function may be linear or non-linear and supplies, demands & conveyance capacities are trapezoidal fuzzy numbers & a fuzzy solution to the problem is required. The formulation of this problem (which has classical STP as a particular case) was given as follows:

$$Minimize \sum_{i \in 1} \sum_{j \in J} \sum_{k \in K} f_{ijk}(x_{ijk}) \tag{34}$$

subject to

$$\sum_{j \in J} \sum_{k \in K} \tilde{1} x_{ijk} \cong \tilde{a}_i, (i \in I); \sum_{i \in I} \sum_{k \in K} \tilde{1} x_{ijk} \cong \tilde{b}_j, (j \in J);$$

$$\sum_{i \in I} \sum_{j \in J} \tilde{1} x_{ijk} \cong \tilde{e}_k, (k \in K); \tag{35 a, b \& c}$$

$$x_{ijk} \geq 0, \quad for\ all\ i, j, k.$$

where $\tilde{a_\iota} = \left(a_i^1, \underline{a}_i^1, a_i^2, \overline{a}_i^2\right),\ i \in I, \quad \tilde{b_\iota} = \left(b_j^1, \underline{b}_j^1, b_j^2, \overline{b}_j^2\right), j \in J, \quad \tilde{e_\iota} = \left(e_i^1, \underline{e}_i^1, e_i^2, \overline{e}_i^2\right), k \in K$ are fuzzy supplies, demands & conveyance capacities & $\tilde{1} = (1, 0, 1, 0)$.

This problem was converted into interval-parametric problem

$$Minimize \sum_{i \in 1} \sum_{j \in J} \sum_{k \in K} f_{ijk}(x_{ijk}) \tag{36}$$

subject to

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} \in \left[a_i^1(r), a_i^2(r)\right], (i \in I); \sum_{i \in I} \sum_{k \in K} x_{ijk} \in \left[b_i^1(r), b_i^2(r)\right], (j \in J);$$
$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \in [e_i^1(r), e_i^2(r)], (k \in K) \tag{37}$$
$$x_{ijk} \geq 0, \quad for\ all\ i, j, k.$$

Where parameter r = minimal feasibility degree for a solution,

$a_i^1(r) = g(\tilde{a}_i) - g\left(\tilde{d}_i^a\right)(1-r), a_i^2(r) = g(\tilde{a}_i) + g\left(\tilde{d}_i^a\right)(1-r),\ i \in I,$
$b_i^1(r) = g\left(\tilde{b}_j\right) - g\left(\tilde{d}_j^b\right)(1-r), b_j^2(r) = g\left(\tilde{b}_j\right) + g\left(\tilde{d}_j^b\right)(1-r),\ j \in J$
and $e_k^1(r) = g(\tilde{e}_k) - g\left(\tilde{d}_k^e\right)(1-r), e_k^2(r) = g(\tilde{e}_k) + g\left(\tilde{d}_k^e\right)(1-r),\ k \in K$

$\tilde{d}_i^a, \tilde{d}_j^b\ and\ \tilde{d}_j^c$ being violations in the fuzzy equality constraints of supplies, demands and conveyance (respectively).

The nonlinear parametric programming problem hence obtained was then solved by a Genetic Algorithm based solution method for the first time proposed by them, by solving an auxiliary Parametric Solid Transportation Problem (PSTP) as follows:

$$Minimize \sum_{i \in 1} \sum_{j \in J} \sum_{k \in K} f_{ijk}(x_{ijk}) \tag{38}$$

subject to

$$\sum_{j\,\epsilon\,J}\sum_{k\,\epsilon\,K} x_{ijk}\ \epsilon\ a_i+\alpha_i r,(i\,\epsilon\,I);\quad \sum_{i\,\epsilon\,I}\sum_{k\,\epsilon\,K} x_{ijk}\ \epsilon\ b_j+\beta_j r,(j\,\epsilon\,J);$$
$$\sum_{i\,\epsilon\,I}\sum_{j\,\epsilon\,J} x_{ijk}\ \epsilon\ e_k+\gamma_k r,(k\,\epsilon\,K); \qquad (39\ a,b\ \&\ c)$$

$$x_{ijk}\geq 0,\qquad for\ all\ i,j,k.$$

$r\,\epsilon\,[LOW,UPP]$, where $a_i\geq 0, b_j\geq 0, e_k\geq 0$, $for\ all\ i,j,k$, $\sum_{i\,\epsilon\,I} a_i = \sum_{j\,\epsilon\,J} b_j \sum_{k\,\epsilon\,K} e_k$, $\sum_{i\,\epsilon\,I}\alpha_i = \sum_{j\,\epsilon\,J}\beta_j \sum_{k\,\epsilon\,K}\gamma_k$. $a_i+\alpha_i r\geq 0, b_j+\beta_j r\geq 0, e_k+\gamma_k r\geq 0, for\ all\ i,j,k$ and $0\leq LOW\leq 1$.

For this they extended the approach of Chanas *et. al.* [7] to solve the Fuzzy transportation problem (FTP) to the solid case. The solution hence obtained to this problem say $x^*_{ijk}(r),(i\,\epsilon\,I,j\,\epsilon\,J,k\,\epsilon\,K)$ for *r* in its limits, is the family of solutions $\{(x^*_{ijk}(r),r)\},(i\,\epsilon\,I,j\,\epsilon\,J,k\,\epsilon\,K)$ is considered as a fuzzy solution to original FSTP.

Liu & Lin [17] defined Fuzzy Fixed Charge Solid Transportation Problem (FFCSTP) from the Solid Fixed Charge Transportation Problem (FCSTP) having model

$$Minimum\left\{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} c_{ijk}\,x_{ijk}+\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} d_{ijk}\,y(x_{ijk})\right\} \qquad (40)$$

subject to

$$\sum_{j=1}^{n}\sum_{k=1}^{l} x_{ijk}\leq a_i,(i\,\epsilon\,I);\ \sum_{i=1}^{m}\sum_{k=1}^{l} x_{ijk}\geq b_j,(j\,\epsilon\,J);\ \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk}\leq e_k,(k\,\epsilon\,K)$$
$$x_{ijk}\geq 0,\qquad for\ all\ i,j,k. \qquad (41)$$

where all costs and parameters are non-negative.

The following concepts of recent advancement of fuzzy variables were used.

(1) Credibility measure of a fuzzy event as:

$$Cr\{A\}=\frac{1}{2}(Pos\{A\}+Nec\{A\}),$$

where $Pos\{A\}$ & $Nec\{A\}$ denote the possibility measure & necessity measure (respectively) for fuzzy event *A*.

(2) Concept of α-optimistic and α-pessimistic values for purpose of ranking of fuzzy variable ξ as

$\xi_{sup}(\alpha)=\sup\{r: Cr\{\xi\geq r\}\geq\alpha\}$ and $\xi_{inf}(\alpha)=\inf\{r: Cr\{\xi\geq r\}\geq\alpha\}$

respectively, where $\alpha\,\epsilon\,[0,1]$.

For the cost $f(x, \tilde{c}, \tilde{d})$ defined as

$$f(x, \tilde{c}, \tilde{d}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} \tilde{c}_{ijk}\, x_{ijk} + \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} \tilde{d}_{ijk}\, y(x_{ijk}) \tag{42}$$

where $x, \tilde{c}\ and\ \tilde{d}$ are the vectors consisting of decision variables x_{ijk}, direct cost $\tilde{c}_{ijk}$ and fixed cost $\tilde{d}_{ijk}$, $i \in I, j \in J, k \in K$. The α-cost of $f(x, \tilde{c}, \tilde{d})$ was defined as minimal value of $\overline{f}$ such that $Cr\{f(x, \tilde{c}, \tilde{d}) \leq \overline{f}\}$, (where α is a confidence level between 0 & 1).

Then the model of FFCSTP is given by

$$minimize\ \overline{f} \tag{43}$$

subject to

$$Cr\{f(x, \tilde{c}, \tilde{d}) \leq \overline{f}\} \geq \alpha\ ; \tag{44}$$

$$Cr\left\{\sum_{j=1}^{n}\sum_{k=1}^{K} x_{ijk} \leq \tilde{a}_i\right\} \geq \beta_1, i \in I\ ; \tag{45 a}$$

$$Cr\left\{\sum_{i=1}^{m}\sum_{k=1}^{K} x_{ijk} \geq \tilde{b}_j\right\} \geq \beta_2, j \in J;\ \ Cr\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} \leq \tilde{e}_k\right\} \geq \beta_3, i \in K \tag{45 b \& c}$$

$x_{ijk} \geq 0$ for all *i, j, k*, where $\alpha, \beta_1, \beta_2\ and\ \beta_3$ are the predetermined confidence levels.

To solve this model, they presented a hybrid intelligent algorithm in which first a neural network was introduced for approximation of uncertain function, and then a Genetic Algorithm was employed to solve the problem.

Bodkhe *et. al.* [3] applied fuzzy programming technique to solve multi-objective solid transportation problem to obtain a compromise solution by defining hyperbolic and exponential (non-linear) membership functions for each of objectives as follows.

Hyperbolic membership function:

$$\mu^H Z_p(X) = \begin{cases} 1, & if\ Z_p \leq L_p \\ \frac{1}{2}\cdot\dfrac{e^{\left\{\frac{(U_p+L_p)}{2} + Z_p(x)\alpha_p\right\}} - e^{\left\{\frac{(U_p+L_p)}{2} - Z_p(x)\alpha_p\right\}}}{e^{\left\{\frac{(U_p+L_p)}{2} + Z_p(x)\alpha_p\right\}} + e^{\left\{\frac{(U_p+L_p)}{2} - Z_p(x)\alpha_p\right\}}} + \frac{1}{2}, & if\ L_p \leq Z_p \leq U_p \\ 0 & if\ Z_p \geq U_p \end{cases} \tag{46}$$

and exponential membership function:

$$\mu^E Z_p(X) = \begin{cases} 1, & if\ Z_p \leq L_p \\ \dfrac{e^{-s\psi_p(x)} - e^{-s}}{1 - e^{-s}} + \dfrac{1}{2}, & if\ L_p \leq Z_p \leq U_p \\ 0 & if\ Z_p \geq U_p \end{cases} \tag{47}$$

where $\psi_p(x) = \big(Z_p(x) - L_p\big)/\big(U_p - L_p\big)$, $p = 1, \dots, P$ and s is a parameter prescribed by the decision maker.

These fuzzy problems were stated to be equivalent to the following crisp problems (respectively).

In case of using hyperbolic membership function:

$$Maximize \quad \lambda \tag{48}$$

subject to

$$\lambda \leq \frac{1}{2}\cdot\frac{e^{\left\{\frac{(U_p+L_p)}{2}+Z_p(x)\alpha_p\right\}}-e^{\left\{\frac{(U_p+L_p)}{2}-Z_p(x)\alpha_p\right\}}}{e^{\left\{\frac{(U_p+L_p)}{2}+Z_p(x)\alpha_p\right\}}+e^{\left\{\frac{(U_p+L_p)}{2}-Z_p(x)\alpha_p\right\}}}+\frac{1}{2}, \qquad p = 1, \dots, P \tag{49}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{K} X_{ijk} = a_i \ (i = 1, \dots, m); \qquad \sum_{i=1}^{m}\sum_{k=1}^{K} X_{ijk} = b_i \ (i = 1, \dots, n);$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ijk} = e_k \ (k = 1, \dots, K); \tag{50 a, b \& c}$$

$X_{ijk} \geq 0$ for all *i, j, k* and $\lambda \geq 0$.

This as per them can further be simplified as:

$$Maximize \quad X_{mn+1} \tag{51}$$

subject to

$$\alpha_p Z_p(x) + X_{mn+1} \leq \frac{\alpha_p\big(U_p + L_p\big)}{2}, \qquad p = 1, \dots, P \tag{52}$$

and conditions (50 a, b & c) with $X_{ijk} \geq 0$ for all *i, j, k* and $X_{mn+1} \geq 0$. Where $X_{mn+1} = tanh^{-1}(2\lambda - 1)$

Similarly in case of using exponential membership function:

$$maximize \ X_3 \tag{53}$$

subject to

$$s\{1 - \psi_p(x)\} \geq X_3, \qquad p = 1, \dots, P \tag{54}$$

and conditions (50 a, b & c) with $X_{ijk} \geq 0$ for all *i, j, k* and $X_3 \geq 0$. Where $X_3 = \log\{1 + \lambda(e^s - 1)\}$.

They solved the problems hence obtained and illustrated the calculations by Linear Interactive and Discrete optimization (LINDO) software.

5 Conclusion

The above survey would be helpful to researchers working in the field of Multi-index transportation problems in context of advancements in various methods evolved to solve the problems with both crisp and fuzzy environment.

References

[1] Anu, A., Arora, S.R.: Multi-Index fixed charge bicriterion transportation problem. Indian J. Pure Appl. Math. 32(5), 739–746 (2001)

[2] Alves, I.C., Yamakami, A., Gomide, F.: An approach for fuzzy linear multicommodity transportation problems and its application. In: Proceedings of 1995 International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium, Fuzzy Systems, vol. 2, pp. 773–780 (1995)

[3] Bodkhe, S.G., Bajaj, V.H., Dhaigude, D.B.: Fuzzy programming technique to solve multi-objective solid transportation problem with some non-linear membership functions. Advances in Computational Research 2(1), 15–20 (2010)

[4] Aysun, B.S., Hamza, B.: An axial four-index transportation problem and its algebraic characterizations. International Journal of Computer Mathematics 81(6), 765–773 (2004)

[5] Cadenas, J.M., Jiménez, F.: A genetic algorithm for the multiobjective solid transportation problem: a fuzzy approach. In: International Symposium on Automotive Technology and Automation, Proceedings for the Dedicated Conferences on Mechatronics and Supercomputing Applications in the Transportation Industries, Aachen, Germany, pp. 327–334 (1994)

[6] Chanas, S., Delgado, M., Verdegay, J.L., Vila, M.A.: Interval and fuzzy extensions of classical transportation problems. Transportation Planning and Technology 17, 203–218 (1993)

[7] Galler, B., Dwyer, P.S.: Translating the method of reduced matrices to machines. Naval Res. Logistics Q 4, 55–71 (1957)

[8] Gen, M., Ida, K., Li, Y.: Solving bicriteria solid transportation problem by genetic algorithm, Humans, Information and Technology. Systems, Man and Cybernetics 2, 1200–1207 (1994)

[9] Haley, K.B.: The Solid transportation problem. Op. Res. 10, 448–463 (1962)

[10] Haley, K.B.: The multi-index problem. Op. Res. 11, 368–379 (1963)

[11] Hammer, P.L.: Time minimization transportation problem. Naval Research Logistics Quarterly 16, 345–357 (1969)

[12] Jimenez, F., Verdegay, J.L.: Interval multiobjective solid transportation problem via genetic algorithm. In: Proc. of the Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, vol. 2, pp. 787–792 (1996)

[13] Jimenez, F., Verdegay, J.L.: Obtaining fuzzy solutions to the fuzzy solid transportation problem with genetic algorithms. In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1657–1663 (1997)

[14] Korsnikov, A.D.: Planar three index transportation problems with dominating index. Mathematical Methods of Operations Research 32(1), 29–33 (1988)

[15] Li, Y., Ida, K., Gen, M.: Evolution Program for multicriteria solid transportation problem with fuzzy numbers. Proc. of IEEE International Conference on Systems, Man and Cybernetics 3, 1960–1965 (1995)
[16] Li, Y., Ida, K., Gen, M., Kobuchi, R.: Neural Network approach for multicriteria solid transportation problem. Computers & Industrial Engineering 33(3-4), 465–468 (1997)
[17] Liu, L., Lin, L.: Fuzzy fixed charge solid transportation problem and its algorithm. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, pp. 585–589 (2007)
[18] Schell, E.D.: Distribution of a product of several properties. In: Proc. 2nd Simposium in Linear Programming, pp. 615–642. DCS/ controller, H.Q. U.S. Air Force Washington, D.C (1955)
[19] Tzeng, G.H., Teodorovic, D., Hwang, M.J.: Fuzzy bicriteria multi-index transportation problems for coal allocation planning of Taipower. European Journal of Operational Research 95, 62–72 (1996)
[20] Vignaux, G.A., Michalewicz, Z.: A genetic algorithm for the linear transportation problem. Systems, Man and Cybernetics 21(2), 445–452 (1991)
[21] Zitouni, R., Keraghel, A.: Resolution of a capacitated transportation problem with four subscripts. Kybernetes 32(9/10), 1450–1463 (2003)
[22] Zitouni, R., Keraghel, A., Benterki, D.: Elaboration and Implantation of an Algorithm Solving a Capacitated Four-Index Transportation Problem. Applied Mathematical Sciences 53(1), 2643–2657 (2007)

Modified Differential Evolution for Constrained Optimization Problems

Musrrat Ali[1] and Millie Pant[2]

[1] Amity University Noida
[2] IIT Roorkee
musrrat.iitr@gmail.com, millifpt@iitr.eernet.in

Abstract. differential evolution (DE) is well known optimization tool for solving global optimization problems. In the present study we present a DE variant called MSDE [22] for solving constrained optimization problems. The proposed algorithm is applied on a set of 6 constrained benchmark problems proposed in CEC 2006 [1]. Numerical results indicate the competence of the proposed algorithm.

Keywords: Differential evolution, constraints, mixed strategy.

1 Introduction

The general non-linear constrained optimization problem (COP) is defined as:

$$\textit{Minimize} \quad f(X): X \in S \to R$$

$$\textit{Subject to:} \quad X = (x_1, x_2, ..., x_D) \in S$$

where *S* is defined by:

$$g_j(X) \le 0, \ j = 1, 2, ..., l$$

$$h_k(X) = 0, \ k = 1, 2, ..., m$$

$$l_i \le x_i \le u_i \ (i = 1, ..., D).$$

l and *m* are the number of inequality and equality constraints respectively, l_i and u_i are lower and upper bounds of the decision variable x_i. D represents the dimension of the problem.

The search space in COPs consists of two kinds of solutions: feasible and infeasible. Feasible points satisfy all the constraints, while infeasible points violate at least one of them. Therefore, the final solution of an optimization problem must satisfy all constraints. A solution *X* is called feasible if

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 933–941.
springerlink.com

$$g_j(X) \le 0, \text{ for all } j = 1, ..., l$$

$$|h_k(X)| - \varepsilon \le 0, \text{ for all } k = 1, ..., m$$

Here equality constraints are transformed into inequality constraints and usually ε is set as 0.0001. A measure of the average constraint violation, which is often useful while handling constraints, is defined as:

$$\overline{v} = \frac{\sum_{j=1}^{l} G_j(X) + \sum_{k=1}^{m} H_k(X)}{l+m}$$

$$\text{where } G_j(X) = \begin{cases} g_j(X) & if \quad g_j(X) > 0 \\ 0 & if \quad g_j(X) \le 0 \end{cases}$$

$$H_k(X) = \begin{cases} |h_k(X)| & if \quad |h_k(X)| - \varepsilon > 0 \\ 0 & if \quad |h_k(X)| - \varepsilon \le 0 \end{cases}$$

COPs has always been a focus of attention to the researchers working in the field of global optimization. Considering the limitations of traditional techniques, nature inspired methods have become popular for dealing with COPs. In the present study, a modified DE called MSDE [22] is used for solving the COP. The rest of the paper is organized as follows: Section 2 gives a brief literature review of DE for solving COP. In section 3, we give the constraint handling technique used in the present study. Benchmark problems and performance metrics are given in section 4. Parameter settings are given in section 5. Results are analyzed and discussed in section 6 and finally, paper is concluded in section 7.

2 DE for Solving COP

DE was initially proposed by Storn and Price in 1995 for solving unconstrained optimization problems. However with a few modifications it can be easily modified for solving constrained problems as well. Several variants of it are available in literature. A feasible region shrinking mechanism was proposed by Storn [2]. The aim was to relax all the constraints of the problems at the beginning of the process. As the time progresses, the pseudo-feasible region shrink at each generation until it matches the real feasible region. Storn also proposed the idea of each parent to generate more than one offspring. In Storn's approach the process finishes when one offspring is better than its parent or when (say) certain numbers of offspring have been generated. He applied his approach on *"DE/best/1/bin"* version. He also added an aging mechanism to avoid a solution to remain in the population for too long. His approach was well suited for problems having inequality constraints, but was not very efficient while dealing with equality constraints. A static-penalty approach, coupled with DE to solve engineering

design problems was proposed by Lampinen and Zelinka (1999,[3] – [5]. The main drawback of their approach is that the careful tuning is required for the penalty factors. Yung-Chien and Feng-Sheng (2002) [6] proposed an Augmented Lagrangian approach with an adaptive mechanism to update the penalty parameters. The approach performed well against typical EA-based techniques. An extension of DE to solve constrained optimization problems was proposed by Lampinen (2002) [7]. The original DE replacement mechanism (based only on the objective function value of the parent and its corresponding offspring) was substituted by three selection criteria based on feasibility originally proposed by Deb (2000) [8]. The difference between Deb's and Lampinen's approach is in the third rule. In Deb's approach the solution with the lowest sum of constraint violation is selected. On the other hand, in Lampinen's technique, the solution which Pareto-dominates the other in the constraints space will be selected. Mezura et al. (2004) [9] proposed a DE-based approach where the newly generated offspring is added to the current generation (instead of including the solution to the next generation). The idea was to allow newly generated solutions to be selected to influence the selection of search directions of the offspring in the current generation and to speed up convergence. Some other DE variants for solving constrained optimization problems can also be found in [10] – [20].

3 Mixed Strategy Differential Evolution MSDE

MSDE is inspired by the evolutionary game theory where the individuals of the population are considered as players trying to decide the optimal mutation strategy for optimizing the solution. MSDE was proposed by the authors for solving unconstrained optimization problems in [22]. Encouraged by its performance we suitably modified it for solving the constrained problems as well. Pareto ranking method proposed by Deb in 2000 [8] for constraint-handling is adopted for dealing with COP. This method is based on the following three rules: (1) Between two feasible vectors, the one with the best value of the objective function is preferred (2) If one vector is feasible and the other one is infeasible, the feasible one is preferred (3) Between two infeasible vectors, the one with the lowest sum of constraint violation is preferred. Besides following the above three rules, equality constraints were transformed into inequations by using the following tolerance value: $\varepsilon = 1e^{-4}$.

4 Test Problems

6 benchmark problems provided in the special session on constrained problems in CEC 2006 [1] are considered in the present paper for analyzing the performance of MSDE. These problems are specially designed to analyze the performance of an optimization algorithm. The list of problems along with the function code, actual minima and other characteristics is summarized in Table 1. This list consists of quadratic, polynomial and cubic models.

Table 1. Name of constrained test problems, assigned codes and characteristics

Function	D	f(X*)	Type of function	N_{EQ}	N_{IEQ}	N_a
g01	13	-15.000	quadratic	0	9	6
g02	20	-0.803	nonlinear	0	2	1
g03	10	-1.000	polynomial	1	0	1
g04	5	-30665.538	quadratic	0	6	2
g05	4	5126.496	cubic	3	2	3
g06	2	-6961.813	cubic	0	2	2

D – Dimension of the problem
f(X*)- best known function value
N_{EQ} – Number of equality constraints
N_{IEQ} – Number of inequality constraints
N_a – Number of active constraints

5 Performance Metrics

The performance measures and their definitions are given below:

- Feasible Run: A run during which at least one feasible solution is found in maximum NFE.
- Successful Run: A run during which the algorithm finds a feasible solution X satisfying (f$(X) - f(X^*) <= \varepsilon$·
- Feasible Rate = (Number of feasible runs) / total runs
- Success Rate = (Number of successful runs) / total runs
- Success Performance = mean (FEs for successful runs) x (Number of total runs) / (Number of successful runs)
- Convergence graphs: The performance of MSDE is also illustrated graphically with the help of convergence graphs (or performance curves) where log of error values are plotted against the NFEs.
- MSDE is compared with 3 other variants of DE for solving the COP considered in this paper: ZRDE [21]; jDE-2 [16]; ε-DE [19]. All these algorithms have been successfully used for solving the constrained benchmark problems proposed in the special session of CEC2006

6 Parameter Settings

Following settings have been taken for the experiments:

- Population Size (NP) = 100 for traditional benchmark problems and 500 for nontraditional problems.
- Scaling/ amplitude Factor F = 0.5.
- Crossover Rate Cr = 0.33
- Maximum NFE = 500000, fixed for all problems [1].
- Accuracy (ε) = 10^{-04} for all the test problems [1]

7 Results and Discussions

Performance of MSDE is evaluated through various performance criteria given in Tables 2 – 4. While Tables 2 and 3 are devoted exclusively to the performance of MSDE, in Table 4, comparison is given with other versions of DE. From Table 3, we see that MSDE gave 100% success in all the test problems except for g03 which it was not able to solve. The feasibility rate was 100% for all the test problems. The satisfactory performance of MSDE is also evident from Fig 1(a) and 1(b). In table 4, we give the comparison of MSDE with some other variants of DE for constrained optimization. Here we see that the performance of MSDE is at par with ZRDE and jDE-2. In fact it performed better then these two for some problems. However, if we do an overall comparison then we see that eDE gave the best performance.

Table 2. Error Values Achieved by MSDE When NFEs= 5×10^3, NFEs = 5×10^4, NFEs = 5×10^5 for Problems 1-6

FES \ Prob.		g01	g02	g03	g04	g05	g06
5×10^3	Best	2.2328e+00(0)	3.50263e-01(0)	9.18082e-01(0)	5.48381e+00(0)	7.88670e+00(3)	1.85916e-01(0)
	Median	3.1881e+00(0)	4.08990e-01(0)	9.95901e-01(0)	9.04867e+00(0)	1.99109e+02(3)	4.16645e-01(0)
	Worst	4.2009e+00(0)	4.50712e-01(0)	1.00046e+00(0)	1.72503e+01(0)	7.6431e+02(0)	3.25477e+00(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,2,3	0,0,0
	$\overline{v}$	0	0	0	0	3.44809e-02	0
	Mean	3.3242e+00	4.18559e-01	9.84616e-01	1.20831e+01	2.75189e+02	8.61851e-01
	Std	6.2025e-01	2.74252e-02	2.31103e-02	4.05658e+00	2.2038e+02	1.0949e+00
5×10^4	Best	1.7234e-07(0)	1.62894e-02(0)	6.85799e-01(0)	8.00355e-11(0)	0(0)	5.72982e-11(0)
	Median	4.62717e-07(0)	8.11970e-02(0)	9.18852e-01(0)	8.00355e-11(0)	2.28064e+01(0)	5.72982e-11(0)
	Worst	3.20588e-06(0)	1.53001e-01(0)	9.90945e-01(0)	8.36735e-11(0)	4.05407e+02(0)	5.72982e-11(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	6.93423e-07	7.34131e-02	8.53123e-01	8.33097e-11	1.25584e+02	5.72982e-11
	Std	0	2.68711e-03	1.20395e-01	1.09139e-12	2.72848e-13	0
5×10^5	Best	0(0)	1.34148e-08(0)	5.85904e-01(0)	8.00355e-11(0)	0(0)	5.72982e-11(0)
	Median	0(0)	9.87368e-08(0)	8.69866e-01(0)	8.00355e-11(0)	0(0)	5.72982e-11(0)
	Worst	0(0)	8.95710e-03(0)	9.43275e-01(0)	8.36735e-11(0)	0(0)	5.72982e-11(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	0	8.95757e-04	7.98692e-01	8.03993e-11	0	5.72982e-11
	Std	0	8.49740e-03	3.80723e-01	3.45129e-12	0	0

Table 3. Number of NFEs to achieve the fixed accuracy level ($(f(\vec{x}) - f(\vec{x}^*)) \leq 0.0001$), Success Rate, Feasible Rate and Success Performance by MSDE

Prob.	Best	Median	Worst	Mean	Std	Feasible Rate	Success Rate	Success Performance
g01	31800	35425	36250	34970	1253	100%	100%	34970
g02	105400	119675	137500	120625	10674	100%	100%	120625
g03	--	--	--	--	--	100%	0%	--
g04	16050	17075	18550	17195	761	100%	100%	17195
g05	16700	117650	218200	122090	55788	100%	100%	122090
g06	7650	8150	9000	8235	361	100%	100%	8235

Table 4. Comparison of MSDE with other algorithms

Problem	Algorithm	Best	Worst	Mean	Feasible Rate (%)	Success Rate (%)	Success Performance
g01	MSDE	31800	36250	34970	100	100	34970
	ZRDE	30511	38028	33414	100	100	33414
	jDE-2	46559	56968	50386	100	100	50386
	εDE	57122	61712	59308	100	100	59308
g02	MSDE	105400	137500	120625	100	100	120625
	ZRDE	95501	129363	113298	100	84	134879
	jDE-2	101201	173964	123490	100	92	145899
	εDE	126152	175206	149825	100	100	149825
g03	MSDE	--	--	--	100	0	--
	ZRDE	-	-	-	100	0	-
	jDE-2	-	-	-	100	0	-
	εDE	86748	91328	89407	100	100	89407
g04	MSDE	16050	18550	17195	100	100	17195
	ZRDE	14048	18362	15986	100	100	15986
	jDE-2	38288	42880	40728	100	100	40728
	εDE	24800	28206	26216	100	100	26216
g05	MSDE	16700	218200	122090	100	100	122090
	ZRDE	16994	204151	107076	100	100	107076
	jDE-2	133340	482304	206620	100	68	446839
	εDE	96812	98589	97431	100	100	97431
g06	MSDE	7650	9000	8235	100	100	8235
	ZRDE	6147	7995	7143	100	100	7143
	jDE-2	26830	31299	29488	100	100	29488
	εDE	6499	8382	7381	100	100	7381

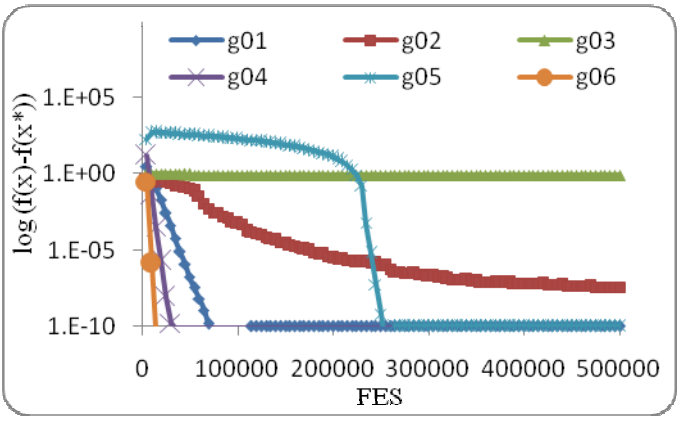


Figure 1 (a). Function error values

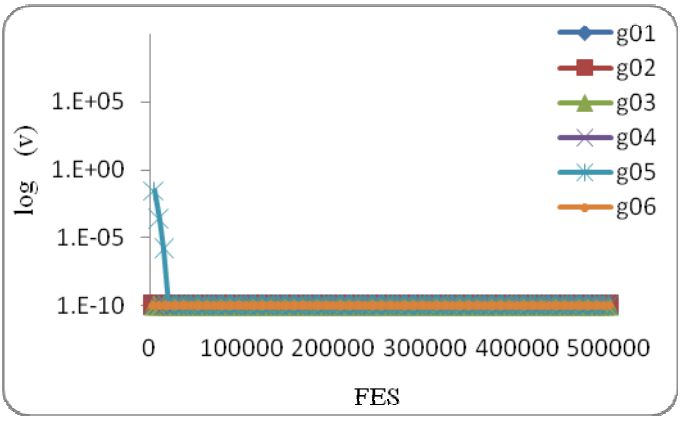


Figure 1 (b). Mean constraint voilation

Fig. 1. Convergence Graph for Problems 01-06 of MSDE

8 Conclusions

DE has emerged as a versatile search technique used for solving constrained and unconstrained optimization problems. In the present study we give some preliminary results of MSDE algorithm for solving constrained optimization problems. We observed that for the considered set of 6 test problems, MSDE gave satisfactory results except for problem g03, which implies that further improvement is needed in MSDE. Experiments are being conducted to test MSDE for all the 24 constrained optimization problems and compare the results with some other constrained DE variants as well.

References

[1] Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, N., Coello, C.A.C., Deb, K.: Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization, Technical Report Report #2006005, Nanyang Technological University, Singapore and et al (December 2006), http://www.ntu.edu.sg/home/EPNSugan

[2] Storn, R.: System Design by Constraint Adaptation and Differential Evolution. IEEE Transactions on Evolutionary Computation 3, 22–34 (1999)

[3] Lampinen, J., Zelinka, I.: Mechanical Engineering Design Optimization by Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 127–146. Mc Graw-Hill, UK (1999a)

[4] Lampinen, J., Zelinka, I.: Mixed Variable Non-Linear Optimization by Differential Evolution. In: Proceedings of 2nd International Prediction Conference on Nostradamus 1999, pp. 45–55. Technical University of Brno, Faculty of Technology Zlin, Department of Automatic Control, Zlin, Czech Republic (October 1999b) ISBN 80-214-1424- 3

[5] Lampinen, J., Zelinka, I.: Mixed Integer-Discrete-Continuous Optimization by Differential Evolution, Part 1: the optimization method. In: Ošmera, P. (ed.) Proceedings of MENDEL 1999, 5th International Mendel Conference on Soft Computing, pp. 71–76. Brno University of Technology, Faculty of Mechanical Engineering, Institute of Automation and Computer Science, Brno, Czech Republic (June 1999c) ISBN 80-214-1131-7

[6] Lin, Y.-C., Hwang, K.-S., Wang, F.-S.: Hybrid Differential Evolution with Multiplier Updating Method for Nonlinear Constrained Optimization Problems. In: Proceedings of the Congress on Evolutionary Computation 2002 (CEC 2002), vol. 1, pp. 872–877. IEEE Service Center, Piscataway (2002)

[7] Lampinen, J.: A Constraint Handling Approach for the Diifferential Evolution Algorithm. In: Proceedings of the Congress on Evolutionary Computation 2002 (CEC 2002), vol. 2, pp. 1468–1473. IEEE Service Center, Piscataway (2002)

[8] Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2/4), 311–338 (2000)

[9] Mezura-Montes, E., Coello Coello, C.A.: An Improved Diversity Mechanism for Solving Constrained Optimization Problems using a Multimembered Evolution Strategy. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 700–712. Springer, Heidelberg (2004)

[10] Lampinen, J.: A Constraint Handling approach for the Differential evolution Algorithm. In: Proc. of the Congress on Evolutionary Computation (CEC 2002), pp. 1468–1473 (2002)

[11] Lampinen, J.: Multi-Constrained Optimization by the Differential Evolution. In: Proc. of the IASTED International Conference Artificial Intelligence Applications (AIA 2001), pp. 177–184 (2001)

[12] Lin, Y.-C., Hwang, K.-S., Wang, F.-S.: Hybrid Differential Evolution with Multiplier updating method for Nonlinear Constrained Optimization. In: Proc. of the Congress on Evolutionary Computation (CEC 2002), pp. 872–877 (2002)

[13] Chiou, J.-P., Wang, F.-S.: Hybrid Method of Evolutionary Algorithms for Static and Dynamic Optimization Problems with Applications to a Fed-Batch fermantation Process. Computers and Chemical Engineering 23, 1277–1291 (1999)

[14] Becerra, R.L., Coello Coello, C.A.: Cultural Differential Evolution for Constrained Optimization. Comput. Methods Appl. Mech. Engrg. (2005)
[15] Zielinski, K., Laur, R.: Constrained Single - Objective Optimization Using Differential Evolution. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 223–230 (2006)
[16] Brest, J., Zumer, V., Maučec, S.: Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 215–222 (2006)
[17] Huang, V.L., Qin, A.K., Suganthan, P.N.: Self-adaptive Differential Evolution Algorithm for constrained Real-Parameter Optimization. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 223–230 (2006)
[18] Fatih Tasgetiren, M., Suganthan, P.N.: A Multi-Populated Differential Evolution Algorithm for Solving Constrained Optimization Problem. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 33–40 (2006)
[19] Takahama, T., Sakai, S.: Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 1–8 (2006)
[20] Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: Modified Differential Evolution for Constrained Optimization. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 25–32 (2006)
[21] Zielinski, K., Laur, R.: Constrained Single - Objective Optimization Using Differential Evolution. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 223–230 (2006)
[22] Ali, M., Pant, M., Abraham, A., Snasel, V.: Differential Evolution Using Mixed Strategies in Competitive Environment. International Journal of Innovative Computing, Information and Control 7(8), 5063–5084 (2011)

Joint Optimization of ICD and Reliability for Component Selection Incorporating "Build-or-Buy" Strategy for Component Based Modular Software System under Fuzzy Environment

Indumati[1], Ompal Singh[1], and U. Dinesh Kumar[2]

[1] Department of Operational Research , University of Delhi , Delhi , India
[2] Indian Institute of Management, Bangalore
{indumati.singh,udkumar}@gmail.com, drompalsingh@live.com

Abstract. Design phase of software development life cycle for designing software with high quality attributes which meets functional requirements is quite a tedious task. In real life the developers can only make ambiguous estimates on the available resources and their aspirations bringing uncertainty (fuzziness) in the problem formulation. In such situations, crisp optimization technique may not serve the purpose to quantify desired parameters. In this paper we have formulated a fuzzy optimization framework which supports build or buy decision strategy i.e., whether to buy software components or to build them in-house or a combination is more beneficial. In this paper, intra-modular coupling density and reliability are jointly optimized under permissible budget and delivery time constraints along with an ICD restriction applied at module level. Redundancy is allowed for the selection procedure for fulfilling each functional requirement. An optimization model for optimal selection of components has been proposed and is illustrated with numerical example.

Keywords: Intra-modular coupling density (ICD), Coupling & Cohesion (C & C), CBSS (Components Based Software System), Fuzzy optimization, Membership function.

1 Introduction

Software is now-a-days daily necessity of everyone's life. It has entered everywhere, in devices, in our pockets, in our vehicles, in our banks, hospitals and homes-and its correct operation is essential for our well-being. In particular, science and technology demand high quality software for making improvements and breakthroughs. Whichever software a company wants to develop & design,

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 943–955.
springerlink.com

they want it to undergo proper process with quantified measures for number of its important characteristics like how much reliable it is, how much easy maintainable it is for developer, client and for the end users etc. When the design of software architecture reaches a good level of maturity, software engineers have to undertake selection decision about software components. COTS (Commercial off-the-shelf) have deeply changed the approach to software design and implementation. COTS components are received from the distributor and are used 'as is'. No changes are normally made to their source codes. Only the code that is necessary to integrate these products is required to develop in house. Large software systems have modular structures and the advancement of technology has made the use of COTS products as modules a possibility.

In the development of modular based conventional software systems, the criteria of minimizing the coupling and maximizing the cohesion of software modules were commonly used (Brito e Abreu & Goulao, 2001). Coupling is about the measure of interactions among software modules while cohesion is about the measure of interactions among the software components which are within a software module. A good software system should possess software modules with high cohesion & low coupling. A highly cohesive module exhibits high reusability and loosely coupled systems enable easy maintenance of a software system (Seker, Vander Merwe, Kotze, Tanik, & Paul, 2004; Kwong et. al., 2010). Large software system has modular structure to perform set of functions with different modules having different components for each module. In the last few years many efforts have been spent to support the design of software architecture. Selecting the appropriate set of components and connectors to make the system meeting functional and non-functional requirements (such as performance and reliability) remains a hard task to accomplish and it very often depends on the architects' experience (Bertolino & Mirandola, 2004). Cortellessa et. al., 2008 introduced a framework that helps developers to decide whether to buy-or-build components of certain software architecture. For building a software system, each component can be either purchased or it can be developed in-house. This is a "build-or-buy" decision that affects the software cost as well as the ability of the system to meet its requirements. In real life situation due to system complexity, varying cost parameters, product availability, intense market competition & many other reasons developers can only make ambiguous estimates on the available resources and their aspirations which bring uncertainty (fuzziness) in the problem formulation. Fuzzy optimization technique is a flexible approach that permits more adequate solution to the real problems even in the presence of vague information (data) & provides well defined mechanism to deal directly with such uncertainties. The idea of fuzzy programming was first given by Bellman and Zadeh (1970) and then developed by Tanaka (1974) and Zimmermann (1976, 1996). Gupta et al. (2009) formulated fuzzy multi objective optimization models for selecting the optimal COTS software products in the development of modular software system. In practice, it is difficult for management to always get precise values of reliability, ICD, interactions data across & within modules and cost for a software system. So it may happen that management decides to put some tolerance level on the objectives / constraints depending upon the availability of the information. So problem

becomes a fuzzy optimization problem & the solution so obtained becomes a fuzzy approximation.

In developing a software system, different functions are required to be performed and different modules are available for those functions. To provide maximum reliable software, one should give emphasis on reliability of modules. In this paper, the reliability of in-house and COTS based modular software is devised with possible redundancy at module level. The objective functions in the model perform weighted maximization of system quality which is assumed to be a weighted function of module reliabilities, weights are decided as per access frequency of each module. We assume that for all the components available for a module, cost increases if higher reliability & ICD is desired. Purchase of high quality COTS products can be justified by the frequent use of a module. Basically, the alternatives/components differ each other for cost and functional/non-functional properties. Also the best amount of testing effort is required to improve reliability of the in-house build component which leads to substantial increase in delivery time (& sometimes on cost also). In this paper, we have incorporated build-or-buy concept for optimal selection of software components for software modules in CBSS development (Cortellessa et al., 2008). The problem is formulated for structure given in Fig. 2. Formulation of a CBSS (Kwong et al., 2010). The optimal selection problem is done by fuzzy objective of maximizing system reliability along with fuzzy objective of maximizing ICD (incorporating impact of maximizing cohesion & minimizing coupling) subject to total budget constraint, total delivery time constraint & ICD w.r.t. each module in which they all have minimum / maximum thresholds with one more system constraint of functional requirement; redundancy is allowed for the selection process. Problem is solved using fuzzy optimization algorithm. The rest of the paper is organized as follows: Section 2 describes the formulation of optimization model for selection of software components for CBSS development. Section 3 describes problem solution procedure. An illustrative example is presented in Section 4. Finally, discussion and conclusions are described in Section 5.

2 Formulation of Optimization Model for CBSS Development

2.1 Crisp Model Description

Let S be a software architecture made of N components for M modules.

2.1.1 Notations

M the number of software modules
N the number of software components
L the number of sets of software components
w_l the frequency of use, of function l
Q_l set of modules required to perform l^{th} function
Sc_i i=1,2,..N, the i^{th} software component
Sc_{ix} i=1,2,..N, the i^{th} software buy component

Sc_{iy} i=1,2,..N, the i^{th} software in-house component
m_j j=1,2,..M, the j^{th} software module
S_k k=1,2,..L, a set of alternative components for the k^{th} functionality
$i \in S_k$ denotes that S_{Ci} belongs to the k^{th} set
Sc_{ij} the i^{th} software component of j^{th} software module, s.t. $Sc_{ij}=Sc_{ij'}= Sc_{ij} \ \forall \ j, j'$
N_{ij}^{suc} number of successful tests attempted on i^{th} in-house component of j^{th} module
N_{ij}^{tot} total number of tests performed on the i^{th} in-house component for j^{th} module
$r_{ii'}$ i,i' =1,2,..N, the number of interactions between S_{Ci} & $S_{Ci'}$.s.t. $r_{ii'}= r_{i'i.}$
$R_{ij,}$ i=1,2,..N, j=1,2,..M,reliability of i^{th} COTS component of j^{th} module
$\rho_{ij,}$ i=1,2,….N, j=1,2,….M, reliability of i^{th} in-house component of j^{th} module
$C_{ij,}$ i=1,2,..N, j=1,2,..M, cost of i^{th} COTS component for j^{th} module
$c_{ij,}$ i=1,2,..N, j=1,2,..M, unitary development cost for i^{th} in-house component of j^{th} module
$S_{ij,}$ i=1,2,..N, j=1,2,..M, joint reliability of i^{th} component available for j^{th} module
π_{ij} probability that a singe execution of software fails on a test case chosen from a certain input distribution while testability of i^{th} component of j^{th} module for in-house product ; π_{ij} is assumed to be $0.002 \ \forall i, j$
t_{ij} estimated development time of i^{th} in-house component of j^{th} module
d_{ij} delivery time of i^{th} COTS component available for j^{th} module
τ_{ij} average time required to perform test case is assumed to be $\tau_{ij} = 0.05 \forall i, j$
R_j reliability of j^{th} software module $; 0 \le R_j \le 1 ; \forall j$
$R_{j'}$ minimum reliability required to attain for j^{th} software module $; 0 \le R_{j'} \le 1; \forall j'$
H a threshold value of ICD_j of each module & set as H=0 by decision makers
C maximum budget limit set by the decision makers
T maximum threshold given on delivery time of the whole system
x_{ij} i=1,2,..,N, j=1,2,..,M, x_{ij} is the binary variable ; x_{ij} =1, if S_{Cix} is selected for m_j for COTS product; otherwise 0
y_{ij} i=1,2,..,N, j=1,2,..,M, y_{ij} is the binary variable ; y_{ij} =1, if S_{Ciy} is selected for m_j for in-house product; otherwise 0
z_{ij} i=1,2,….N, j=1,2,….M, z_{ij} is the binary variable ; z_{ij} =1, if S_{Ci} is selected for m_j for either buy S_{Cix} or in-house product S_{Ciy} ; otherwise 0

2.1.2 Assumptions

1. Atleast one component is supposed to get selected from each module either COTS or in-house, redundancy is allowed for both the components.
2. A threshold value of ICD_j, budget, delivery time & reliability are set by the decision makers.
3. More than one software component in S_k either in-house or COTS or combination of both) may get selected to implement k^{th} functional requirement.
4. The cost of component is the development cost, if developed in house; otherwise it is the buying price of the COTS product. Reliability data set for COTS components is known.

5. The cost and reliability of an in-house component can be specified by using basic parameters of the development process, e.g. a component cost may depend on measure of developer skills, or the component reliability depends on the amount of testing.
6. Interaction data for COTS or in-house components is same for all modules.
7. Different COTS alternatives with respect to cost, reliability and delivery time of a module are available.
8. Different in- house alternatives w.r.t. unitary development cost, estimated development time, average time and testability of a module are available.

2.1.3 Objective Functions and Constraints

2.1.3.1 Testability Conditions

Basing on the testability definition, we can assume that the number N_{ij}^{suc} of successful tests performed on the same component can be obtained as:

$$N_{ij}^{suc} = (1-\pi_{ij})N_{ij}^{tot} \qquad i = 1,2,...,N;\ j,\ j' = 1,2,....,M \tag{1}$$

2.1.3.1.1 Build versus Buy Decision

If i[th] component of j[th] module is bought (i.e. some $x_{ij} = 1$) then there will be no in-house development (i.e. $y_{ij} = 0$) and vice versa.

$$x_{ij} + y_{ij} \le 1 \quad i = 1,2,.....,N; j = 1,2,.....,M \tag{2}$$

2.1.3.1.2 Redundancy

The equation stated below guarantees that redundancy is allowed for both the build and buy components (i.e. in-house and COTS components).

$$z_{ij} = x_{ij} + y_{ij} \ ;\ \sum_{i=1}^{N} z_{ij} \ge 1 \ ;\ i = 1,2,...,N; j = 1,2,...,M \tag{3}$$

2.1.3.2 Cost Constraint

The in-house development cost of the i[th] component available for j[th] module can be expressed as $c_{ij}(t_{ij} + \tau_{ij}N_{ij}^{tot})$. So over all cost constraint can be expressed as:

$$\sum_{j=1}^{M}[\sum_{i=1}^{N} c_{ij}(t_{ij} + \tau_{ij}N_{ij}^{tot})y_{ij} + (\sum_{i=1}^{N} C_{ij}x_{ij})] \le C \tag{4}$$

2.1.3.3 Delivery Time Constraint

The delivery time of i[th] in-house developed component for j[th] module is $(t_{ij} + \tau_{ij}N_{ij}^{tot})$ .So overall delivery time constraint can be expressed as:

$$\sum_{j=1}^{M}[\sum_{i=1}^{N} (t_{ij} + \tau_{ij}N_{ij}^{tot})y_{ij} + (\sum_{i=1}^{N} d_{ij}x_{ij})] \le T \tag{5}$$

2.1.3.4 Reliability of an In-House Developed Component

Cortellessa et al. 2008 defined the probability of failure on demand of an in-house developed i[th] component of j[th] module under the assumption that the on-field users' operational profile is the same as the one adopted for testing (Bertolino and Strigini, 1996). Let A be the event "N_{ij}^{suc} failure – free test cases have been performed" & B be the event "alternative is failure free during a single run". If ρ_{ij} is the probability that the in- house developed alternative is failure free during a single run given that N_{ij}^{suc} test cases have been successfully performed, then from the Bayes Theorem we get - $\rho_{ij} = P(B/A) = \frac{P(A/B)P(B)}{P(A/B)P(B) + P(A/\bar{B})P(\bar{B})}$

The following equalities come straightforwardly:

$P(A/B) = 1$; $P(B) = 1 - \pi_{ij}$; $P(A/\bar{B}) = (1-\pi_{ij})^{N_{ij}^{suc}}$; $P(\bar{B}) = \pi_{ij}$ therefore, we have

$$\rho_{ij} = \frac{1-\pi_{ij}}{\left(1-\pi_{ij}\right) + \pi_{ij}\left(1-\pi_{ij}\right)^{N_{ij}^{suc}}} \quad ; \quad 0 \le \rho_{ij} \le 1 \; ; i = 1,2,...,N; j = 1,2,...,M \tag{6}$$

Reliability equation of both in-house and COTS components

$$S_{ij} = \rho_{ij} y_{ij} + R_{ij} x_{ij} \quad ; \quad 0 \le \rho_{ij}, R_{ij}, S_{ij} \le 1; \; i = 1,2,...,N; \; j = 1,2,...,M \tag{7}$$

2.1.3.5 Reliability Objective Function

Reliability objective function maximizes system quality. So total system reliability $(0 \le R \le 1)$ can be expressed as:

$$R = \sum_{l=1}^{L} w_l \prod_{j \in Q_l} R_j \quad where \quad 0 \le \left\{ R_j = \prod_{i \in M_j} [\rho_{ij}]^{y_{ij}} [R_{ij}]^{x_{ij}} \right\} \le 1 \; ; \; i = 1,2,...,N; j = 1,2,...,M \tag{8}$$

2.1.3.6 Cohesion, Coupling & ICD Objective Function (As per "Build or Buy")

The cohesion (Abreu & Goulão, 2001) incorporating both in-house & COTS components impact (Cortellessa et al., 2008) within the j[th] module, $(CI_{IN})_j$ is:

$$(CI_{IN})_j = \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} r_{ii'} z_{ij} z_{i'j} \tag{9}$$

All interactions including C & C associated with the j[th] module, CA_j becomes:

$$CA_j = (CI_{IN})_j + (CI_{OUT})_j = \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} r_{ii'} z_{ij} \left(\sum_{j'=1}^{M} z_{i'j'} \right) \tag{10}$$

All interactions including C & C of a software system, CA_j can be expressed as:

$$CA = \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} r_{ii'} \left(\sum_{j=1}^{M} z_{ij} \right) \left(\sum_{j=1}^{M} z_{i'j} \right) \tag{11}$$

The sum of cohesions within all modules CI_{IN} can be expressed as:

$$CA=\sum_{j=1}^{M}\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}z_{ij}z_{i'j} \tag{12}$$

Intra-modular coupling density (ICD) was introduced to measure ratio relationship between C & C of modules in the design of an object oriented software system as:

$$ICD=\frac{CI_{IN}}{CI_{IN}+CI_{OUT}} \rightarrow ICD_j=\frac{(CI_{IN})_j+1}{(CI_{IN})_j+(CI_{OUT})_j} \tag{13}$$

where CI_{IN} is the number of class interactions within modules ; CI_{OUT} is the number of interactions between classes of distinct modules ; ICD_j is the intra-modular coupling density for the j[th] module (If one module contains only one component then the value of ICD for that module becomes zero. To make up for the deficiency, another measure of ICD_j was given); $(CI_{IN})_j$ is the number of component interactions within the j[th] module; & $(CI_{OUT})_j$ is the number of component interactions between the j[th] module & other modules. The value of ICD would have greater impact on the maintainability of a CBSS & can be expressed as:

$$ICD=\frac{\sum_{j=1}^{M}\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}z_{ij}z_{i'j}}{\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}\left(\sum_{j=1}^{M} z_{ij}\right)\left(\sum_{j=1}^{M} z_{i'j}\right)} \quad ; \quad 0\le ICD\le 1 \tag{14}$$

Crisp optimization problem can be formulated as below problem (P1):

$$Max\ ICD=\frac{\sum_{j=1}^{M}\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}z_{ij}z_{i'j}}{\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}\left(\sum_{j=1}^{M} z_{ij}\right)\left(\sum_{j=1}^{M} z_{i'j}\right)} \tag{15}$$

$$Max\ R=\sum_{l=1}^{L} w_l \prod_{j\in Q_l} R_j \quad where \quad 0\le\left\{R_j=\prod_{i\in M_j}[\rho_{ij}]^{y_{ij}}[R_{ij}]^{x_{ij}}\right\}\le 1 \ ; \ i=1,2,...,N; j=1,2,...,M \tag{16}$$

$S_{ij}=\rho_{ij}y_{ij}+R_{ij}x_{ij} \quad i=1,2,.....,N; \ j=1,2,.....,M$

subject to $S=\left\{z_{ij}=(x_{ij},y_{ij})\ binary\ variable\in\{0,1\}\right\}$; i = 1,..,N; j = 1,..,M

$$\frac{\sum_{j=1}^{M}\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}z_{ij}z_{i'j}+1}{\sum_{i=1}^{N-1}\sum_{i'=i+1}^{N} r_{ii'}z_{ij}\left(\sum_{j'=1}^{M} z_{i'j'}\right)}\ge H \quad j,j'=1,2,.....,M \tag{17}$$

$$\prod_{i\in M_j}[\rho_{ij}]^{y_{ij}}[R_{ij}]^{x_{ij}}\ge R_{j'} \quad j,j'=1,2,.....,M \tag{18}$$

$$N_{ij}^{suc}=(1-\pi_{ij})N_{ij}^{tot} \quad i=1,2,.....,N; j,j'=1,2,.....,M \tag{19}$$

$$\rho_{ij}=\frac{1-\pi_{ij}}{\left(1-\pi_{ij}\right)+\pi_{ij}\left(1-\pi_{ij}\right)^{N_{ij}^{suc}}} \ i=1,2,....,N; j=1,2,....,M \tag{20}$$

$$\sum_{j=1}^{M}[\sum_{i=1}^{N} c_{ij}(t_{ij}+\tau_{ij}N_{ij}^{tot})y_{ij}+(\sum_{i=1}^{N} C_{ij}x_{ij})]\leq C \tag{21}$$

$$\sum_{j=1}^{M}[\sum_{i=1}^{N} (t_{ij}+\tau_{ij}N_{ij}^{tot})y_{ij}+(\sum_{i=1}^{N} d_{ij}x_{ij})]\leq T \tag{22}$$

$$\sum_{i\in S_k}\sum_{j=1}^{M} z_{ij}\geq 1 \tag{23}$$

$$z_{ij}=x_{ij}+y_{ij} \quad i=1,2,.....,N; j=1,2,.....,M \tag{24}$$

$$x_{ij}+y_{ij}\leq 1 \quad i=1,2,.....,N; j=1,2,.....,M \tag{25}$$

$$\sum_{i=1}^{N} z_{ij}\geq 1 \quad j=1,2,.....,M \Big\} \tag{26}$$

2.2 Fuzzy Multi-objective Optimization Model

Principle to multi-objective optimization is the concept of an "efficient solution", where any improvement of one objective can only be achieved at the expense of another. Fuzzy approach can be used as an effective tool for quickly obtaining a good compromised solution in such scenario. In general reliability optimization problem is solved with an assumption that the coefficients are specified precisely. Sometimes it is not possible to get relevant precise data for reliability or ICD & imprecise data is not always well represented by random variable selected from a probability distribution. Fuzzy number may represent this data in a better way, so fuzzy optimization method with fuzzy parameters is what we needed for the purpose. So in such case fuzzy multi-objective optimization model can be formulated based on vague aspiration levels, decision makers may decide aspiration levels based on past experience and knowledge. To express vague aspiration levels of the decision, various membership functions have been proposed (Zimmermann, 1976, 1996). Fuzzy objective function is shown by $\supset$ & rest of the inequalities are kept as crisp. The fuzzy multi-objective optimization model based on maximizing both the fuzzifier objectives of reliability & ICD subject to rest of the crisp constraints of S is given as follows:

$Max\ I\tilde{C}D$

$Max\ \tilde{R}$

subject to

$$Z\in S=\left\{z_{ij}=(x_{ij},y_{ij})\in\{0,1\}\ /\ \text{Satisfying eq. (17) to (26);}\ i=1,....N;\ j=1,2,.....,M\right\} \tag{P2}$$

3 Problem Solution

Following algorithm (Jha et. al, 2011) specifies the sequential steps to solve the fuzzy mathematical programming problems.

1. Compute the crisp equivalent of the fuzzy parameters using a defuzzification function (ranking of fuzzy numbers) for each of the parameters. Here we use the de fuzzification function of type, $F_2(A)=(a_l+2a+a_u)/4$; where (a_l,a,a_u) are triangular fuzzy numbers.
2. Incorporating objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The above problem (P2) can be rewritten as:

   Find Z

   subject to

   $R(Z)\tilde{\geq}R_0$

   $ICD(Z)\tilde{\geq}ICD_0$

$$Z\in S=\left\{z_{ij}=(x_{ij},y_{ij})\in\{0,1\}\ /\ \text{Satisfying eq. (17) to (26);}\ \ i=1,....N;\ j=1,2,.....,M\right\} \tag{P3}$$

   where R_0 & ICD_0 are defizzified aspiration levels of system reliability & ICD.
3. Define appropriate membership functions for fuzzy inequalities for 'less than or equal to' & 'greater than or equal to' type is:

$$\mu_R(Z)=\begin{cases}1 & ;R(Z)\geq R_0\\ \dfrac{R(Z)-R^*}{R_0-R^*} & ;R^*\leq R(Z)<R_0\\ 0 & ;R(Z)<R^*\end{cases}\ \&\ \mu_{ICD}(Z)=\begin{cases}1 & ;ICD(Z)\geq ICD_0\\ \dfrac{ICD(Z)-ICD^*}{ICD_0-ICD^*} & ;ICD^*\leq ICD(Z)<ICD_0\\ 0 & ;ICD(Z)<ICD^*\end{cases}$$ respectively.

   where R* & ICD* are the corresponding tolerance levels.
4. Employ extension principle to identify the fuzzy decision for converting problem objectives to constraints where each constraint is considered to be an objective for the decision maker & the problem can be looked as a fuzzy multi-objective mathematical programming problem. Further each objective can have different level of importance & can be assigned different weights to measure relative importance for defining the following crisp optimization problem:

   $Maximize\ \ \alpha$

   subject to

   $\mu_R(Z)\geq\lambda_1\alpha$

   $\mu_{ICD}(Z)\geq\lambda_2\alpha$

   $Z\in S=\left\{z_{ij}=(x_{ij},y_{ij})\in\{0,1\}\ /\ \text{Satisfying eq. (17) to (26);}\ \ i=1,....N;\ j=1,2,.....,M\right\}$

$$\sum_{i=1}^{2}\lambda_i=1\ ;\quad \alpha>0,\ \alpha\leq 1,\{\lambda_i\geq 0: i=1,2\},Z\geq 0 \tag{P4}$$

4 Numerical Illustration

A numerical example of a financial system design is illustrated to illustrate the proposed methodology of optimization in Table 1.

Table 1. Example Description (Kwong, et al., 2010)

Functional requirements	S_k	S/W components	
Facsimile/Fax,R'1	S_1	Sc_{1x}, $Sc_{1v} = Sc_1$	OpalVOIP
		Sc_{2x}, $Sc_{2v} = Sc_2$	HylaFAX
		Sc_{3x}, $Sc_{3y} = Sc_3$	Faxman
		Sc_{4x}, $Sc_{4v} = Sc_4$	HylaPEx
Encryption, R'2	S_2	Sc_{5x}, $Sc_{5v} = Sc_5$	CryptoXpress CF
Credit Card Authorization, R'3	S_3	Sc_{6x}, $Sc_{6y} = Sc_6$	IBiz E
eCommerce/Account, R'4	S_4	Sc_{7x}, $Sc_{7v} = Sc_7$	Symphero
		Sc_{8x}, $Sc_{8v} = Sc_8$	Shopformat
		Sc_{9x}, $Sc_{9v} = Sc_9$	Account manager
		Sc_{10x}, $Sc_{10y} = Sc_{10}$	Account services manager

A software system is decomposed into three modules m_1, m_2 & m_3, ten COTS components (Sc_{1x}–Sc_{10x}) available in market & ten components (Sc_{1y}–Sc_{10y}) that can be build in-house to make up four sets of alternative software components (S1–S4) are considered. Total twenty components are available for selection & may be represented as (Sc1–Sc10). Table 2 shows the degrees of interaction among software components (both COTS & in-house). The range of the degrees is 1–10. The degree '1' means a very low degree of interaction while the degree '10' refers to a very high degree of interaction. In Table 3 & 5, reliability ($0 \le R_{ij} \le 1$ for all i, j); cost (in 100$ unit) & delivery time (in hours) associated with COTS is given. In Table 4 development cost (in 100$ unit) & estimated time (in hours) associated with in-house components is given.

Table 2. Interactions among software COTS & in-house components

		Sc_1	Sc_2	Sc_3	Sc_4	Sc_5	Sc_6	Sc_7	Sc_8	Sc_9	Sc_{10}
S_1	Sc_1	0	0	5	10	0	0	0	0	4	10
	Sc_2	0	0	0	8	6	6	2	3	0	10
	Sc_3	5	0	0	4	0	2	6	3	6	3
	Sc_4	10	8	4	0	0	9	0	5	6	4
S_2	Sc_5	0	6	0	0	0	8	0	7	0	4
S_3	Sc_6	0	6	2	9	8	0	0	0	0	0
S_4	Sc_7	0	2	6	0	0	0	0	7	0	0
	Sc_8	0	3	3	5	7	0	7	0	10	8
	Sc_9	4	0	6	6	0	0	0	10	0	0
	Sc_{10}	10	10	3	4	4	0	0	8	0	0

Table 3. Reliability & Cost Data Set (COTS)

Module m_1		Module m_2		Module m_3		Module m_1		Module m_2		Module m_3	
R_{11}	0.95	R_{12}	0.99	R_{13}	0.98	C_{11}	10	C_{12}	7	C_{13}	6
R_{21}	0.98	R_{22}	0.98	R_{23}	0.91	C_{21}	6	C_{22}	8	C_{23}	7
R_{31}	0.92	R_{32}	0.99	R_{33}	0.94	C_{31}	8	C_{32}	9	C_{33}	8
R_{41}	0.95	R_{42}	0.97	R_{43}	0.96	C_{41}	9	C_{42}	10	C_{43}	9
R_{51}	0.96	R_{52}	0.96	R_{53}	0.9	C_{51}	8	C_{52}	6	C_{53}	7
R_{61}	0.95	R_{62}	0.97	R_{63}	0.95	C_{61}	6	C_{62}	7	C_{63}	8
R_{71}	0.96	R_{72}	0.96	R_{73}	0.9	C_{71}	7	C_{72}	8	C_{73}	9
R_{81}	0.98	R_{82}	0.9	R_{83}	0.95	C_{81}	8	C_{82}	9	C_{83}	6
R_{91}	0.95	R_{92}	0.96	R_{93}	0.96	C_{91}	9	C_{92}	6	C_{93}	7
R_{101}	0.96	R_{102}	0.9	R_{103}	0.97	C_{101}	6	C_{102}	10	C_{103}	8

Table 4. Development Cost & Estimated Development Time Data Set (In-House)

Module m_1		Module m_2		Module m_3		Module m_1		Module m_2		Module m_3	
c_{11}	5	c_{12}	2	c_{13}	1	t_{11}	9	t_{12}	6	t_{13}	5
c_{21}	2	c_{22}	3	c_{23}	2	t_{21}	6	t_{22}	7	t_{23}	6
c_{31}	3	c_{32}	4	c_{33}	3	t_{31}	7	t_{32}	4	t_{33}	7
c_{41}	4	c_{42}	5	c_{43}	4	t_{41}	8	t_{42}	9	t_{43}	8
c_{51}	3	c_{52}	1	c_{53}	2	t_{51}	7	t_{52}	5	t_{53}	6
c_{61}	1	c_{62}	2	c_{63}	3	t_{61}	5	t_{62}	6	t_{63}	7
c_{71}	2	c_{72}	3	c_{73}	4	t_{71}	6	t_{72}	7	t_{73}	8
c_{81}	3	c_{82}	4	c_{83}	1	t_{81}	7	t_{82}	5	t_{83}	5
c_{91}	4	c_{92}	1	c_{93}	2	t_{91}	8	t_{92}	5	t_{93}	6
c_{101}	1	c_{102}	5	c_{103}	3	t_{101}	5	t_{102}	9	t_{103}	7

Table 5. Delivery Time Data Set (COTS)

Module m_1		Module m_2		Module m_3	
d_{11}	1	d_{12}	4	d_{13}	5
d_{21}	5	d_{22}	3	d_{23}	4
d_{31}	3	d_{32}	2	d_{33}	3
d_{41}	2	d_{42}	1	d_{43}	2
d_{51}	3	d_{52}	5	d_{53}	4
d_{61}	5	d_{62}	4	d_{63}	3
d_{71}	4	d_{72}	3	d_{73}	2
d_{81}	3	d_{82}	2	d_{83}	5
d_{91}	2	d_{92}	5	d_{93}	4
d_{101}	5	d_{102}	1	d_{103}	3

The value of below fuzzy numbers are assumed to be specified by the management based on the past experiences and\or expert opinion. Aspiration levels set for reliability & ICD are very high $R_0 = 0.99$ & $ICD_0 = 1$, keeping in mind the client requirement of highest maintainability & reliability for the system. The tolerance level are set as R*=0.82 & ICD*=0.970. The constraints corresponding to reliability & ICD are very important constraints which may be assigned different weights as per

their relative importance. It is reasonable to assign equal weights $(\lambda_1, \lambda_2) = (0.5, 0.5)$ to both the objectives using weighted min-max technique as follows:

$$Maximize \ \alpha$$

subject to $$\mu_R(Z) = \frac{\sum_{l=1}^{L} w_l \prod_{j \in Q_l} R_j - 0.82}{0.99 - 0.82} \geq 0.5 * \alpha \quad ; \quad R_j = \prod_{i \in M_j} [\rho_{ij}]^{y_{ij}} [R_{ij}]^{x_{ij}} \quad i = 1,2,...,N; j = 1,2,...,M$$

$$\mu_{ICD}(Z) = \frac{\left(\frac{\sum_{j=1}^{M} \sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} r_{ii'} z_{ij} z_{i'j} + 1}{\sum_{i=1}^{N-1} \sum_{i'=i+1}^{N} r_{ii'} z_{ij} \left(\sum_{j'=1}^{M} z_{i'j'} \right)} \right) - 0.970}{1 - 0.970} \geq 0.5 * \alpha$$

$$Z \in S = \left\{ z_{ij} = (x_{ij}, y_{ij}) \in \{0,1\} \ / \text{Satisfying eq. (17) to (26); } i = 1,....N; j = 1,2,....,M \right\} \qquad \text{(P5)}$$

$$\alpha > 0, \alpha \leq 1, Z \geq 0$$

Solving problem (P5) we get the following solution in Table 6.

Table 6. Input – Output

Results of selection of software components for modules		Module m_1	Module m_2	Module m_3	R	ICD	Objective Value
Cases	Input	Output					
Case 1	R_1'=0.99, R_2'=0.99, R_3'=0.99, C=60, T=60	Sc_{71y}	Sc_{52y}, Sc_{62y}	Sc_{13y}	0.99	1	1
Case 2	R_1'=0.96, R_2'=0.92, R_3'=0.94, C=63, T=57	Sc_{71y}	Sc_{52y}, Sc_{62x}	Sc_{13y}	0.97	1	0.999900
Case 3	R_1'=0.94, R_2'=0.96, R_3'=0.97, C=64, T=56	Sc_{11x}	Sc_{52y}, Sc_{62x}	Sc_{13y}, Sc_{73y}	0.93	0.99	0.958800
Case 4	R_1'=0.91, R_2'=0.94, R_3'=0.89, C=65, T=58	Sc_{71x}	Sc_{72x},	Sc_{13y}, Sc_{53y}, Sc_{63x}	0.90	0.99	0.9674855
Case 5	R_1'=0.88, R_2'=0.76, R_3'=0.78, C=70, T=63	Sc_{51x} Sc_{61x}	Sc_{72x}	Sc_{13x}	0.87	1	0.7022541

The solution is a local optimal solution for the above mentioned Integer Non-Linear Programming Formulation for the given data sets, solved on the software Lingo (Version 11), solver type Branch & Bound. The optimal solution for problem (P5) is an optimal solution for problem (P2). After reading all five solutions we can observe if we have more delivery time & budget availability then we must prefer to create in-house component, as in-house is more reliable as compared to COTS in which we are having more risk of low quality at higher cost.

5 Discussion and Conclusion

In this paper, a methodology of selecting software components with impact of whether to build-or-buy components for CBSS development is described. A fuzzy multi-objective optimization model is formulated to perform selection of software components for the software modules of a CBSS. An example of a financial system design was used to illustrate the methodology along with weighted min-max approach. This methodology involves some subjective judgments from software development teams such as determination of the scores of interaction, reliability, weights technique etc., so it can be noted that the fuzzy optimization method provides sub-optimal solutions due to lot of subjective involved in the nature of the fuzzy optimization techniques.

References

[1] Abreu, F.B., Goulão, M.: Coupling & cohesion as modularization drivers:Are we being over-persuaded. In: Proceedings of 5th European Conference on Software Maintenance & Re-engineering. IEEE Computer Society, Washington, DC, USA (2001)

[2] Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. Management Science 17, 141–164 (1970)

[3] Bertolino, Mirandola, R.: CB-SPE tool, Putting component-based performance engineering into practice. In: Proceedings of 7th International Symposium on Component Based Software Engineering, Conference CASCON 97, Toronto, Ontario, pp. 10–13 (2004)

[4] Bertolino, A., Strigini, L.: On the use of testability measures for dependability assessment. IEEE Transactions on Software Engineering 22(2), 97–108 (1996)

[5] Cortellessa, V., Marinelli, F., Potena, P.: An optimization framework for 'build-or-buy'decisions in software architecture. Computers & Operations Research 35, 3090–3106 (2008)

[6] Gupta, P., Mehlawat, M.K., Mittal, G., Verma, S.: A Hybrid Approach for Selecting Optimal COTS Products. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2009. LNCS, vol. 5592, pp. 949–962. Springer, Heidelberg (2009)

[7] Jha, P.C., Indumati, S.O., Gupta, D.: Bi-criterion release time problem for a dis-crete SRGM under fuzzy environment. International Journal of Mathematics in Operational Research 3(6), 680–696 (2011)

[8] Kwong, C.K., Mu, L.F., Tang, J.F., Luo, X.G.: Optimization of software components selection for component based software system development. Elsevier, Computer & Industrial Engineering (2010)

[9] Seker, R., Van der Merwe, A.J., Kotze, P., Tanik, M.M., Paul, R.: Assessment of coupling and cohesion for component based software by using Shannon languages. Journal of Integrated Design & Process Science 8(4), 33–43 (2004)

[10] Tanaka, H., Okuda, T., Asai, K.: On fuzzy mathematical programming. Journal of Cybernetics 3, 37–46 (1974)

[11] Zimmermann, H.J.: Fuzzy Set Theory and its Applications, 3rd edn. Kluwer Academic Publishers, Nowell (1996)

Promotional Allocation Problem for a Multi Generational Product in Segmented Market

P.C. Jha, Anshu Gupta, and Yogender Singh

Department of Operational Research, University of Delhi, Delhi, India
{jhapc,anshu_or}@yahoo.com, aeiou.yogi@gmail.com

Abstract. Multiple generations of many durable products serves the markets simultaneously. As technology is growing at a very fast rate, new technology creates the need of higher technology products, and makes firms to introduce new products even when a product is doing well in the market in the order of competition, survival etc. Existence of several generations altogether in the market directly or indirectly affects the sales of the each other by the way of the effects of substitution, upgradation, leapfrogging etc. Products either new or old need to be promoted in the market - initially to create the awareness and later on the grounds of market share and capture. Promotional effort is an important component of marketing mix that determines the success of a product, so must be spent judiciously. In the marketing literature many promotional allocation problems have been discussed under various concerns, but most of them either consider a single product or multiple products each of which is different from other. The effect of existence of several generations of the same product has been ignored while making these allocations. In this paper we formulate a promotional allocation problem for durable technology product using a diffusion substitution model to capture the sales of multiple generations of the product for a segmented market. The model captures the sales of the products along with their substitution effect. The optimization model formulated is a non-linear programming problem. It allocates promotional efforts to the different generations of a product in a planning horizon maximizing sales under budgetary constraint. The model application is illustrated through a numerical example.

Keywords: Promotional resources, Technology substitution diffusion models, Promotion effort allocation, Segmented market.

1 Introduction

Promotion is the most dominant marketing mix variable as it creates impression of the product in the minds of the customers. It influences everyone whether a potential customer or not. An effective promotion generates value of product in the minds of potential segment and persuades them to buy the product while other

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 957–970.
springerlink.com

may be so affected that they become potential adopter for it in future. In either case it determines the success of the product. An effective promotional strategy can make even a weak product to survive for long in the market. Promotion demands firm's resources usually financial. No firm can spent infinite resources on promotion; they are limited and fixed for every planning period during the product life cycle. As such it is desired that they should be efficiently used so as to create the maximum impact which can be measured through sales. All the adopters of any product does not possess similar characteristics, they differ with each other. The differences in the adapters can be demographic, psychographic, and behavioral etc. Therefore marketer of the product divides the population of its potential adopters into distinct groups of consumers with common characteristics. These groups of potential adopters are called as segments in marketing. Since the potential adopter population is not homogeneous therefore promotional efforts made by the firm are target oriented so as to target each segment of consumer distinctly.

In a segmented market it is assumed that each segment responds uniquely to the promotional effort done exclusively for it, the firms desire to allocate optimally the promotional resources among the segments so as to maximize total product sales. Such a problem can be formulated as a mathematical programming problem. Formulation of a promotional effort allocation problem requires a relationship between sales, planning period and promotional resources for each segment. In the marketing literature many promotional allocation problems [13,15,17,20] have been discussed under various concerns. Such problems are formulated using the diffusion models to describe the relationship between sales and time or promotional resources. Diffusion models provide foundation to implement and forecast marketing strategies. Most of these problems either consider a single product [17,20] or multiple products each of which are different from other and use diffusion models applicable to single products, to describe this relationship. In case of multiple products [13,15], products are assumed to be independent of each other and again diffusion models for single product are used. These problems falls in the category of nonlinear optimization problems and are solved using nonlinear programming methods such as dynamic programming, multi-objective and goal programming methods.

If we carefully see the segment of durable technology, several generations of the same product exist simultaneously in the market due to rapidly growing technology and demand of durable products based on most recent technology. In such cases the multiple generations of the same product does not behave independent to each other while there exist relationship between their sales, multiple generations shows the various interactions like substitution effect, upgradation effect, increase market potential, leapfrogging etc [8,19,22,27] between their sales. Introduction of a newer generation offers competition to the existing generation. However the sales of the existing generation may not always decrease due to existence of newer generation. It can increase due to expansion of its consumer segment as the price of older product is lowered with the introduction of new product etc. While it may decrease, due to switching of some of its potential segment to the newer product. The net effect can be either increasing or decreasing. New product also gets its potential segments from the switchers and upgraders of previous generation and new potential from the

market. Clearly in such a case diffusion models applicable to single independent products may not result in optimal allocation of promotional resources.

Fourt and Woodlock [9], and Bass [1] are the earliest models of new product diffusion in marketing that attempt to describe the penetration and saturation aspects of the diffusion process. The main momentum underlying diffusion research is the Bass model. A number of modifications have been carried out in the model to capture the variability in adopter behavior and other market conditions (Easingwood et al. [7], Horsky and Simon [11,12], Mahajan and Muller[23],Mahajan and Peterson [24,25], Kalish [16]. Many economists have suggested that Bass Model is incomplete because it does not unite the contagion effects with traditional economic variables such as price and advertising. Bass [2] developed a generalized version of the Bass model called as "The Generalized Bass Model (GBM)" that reflects the current effect of dynamic marketing variables on the conditional probability of adoption at a particular time. Jha et al. [14] proposed an alternative formulation of GBM, defining the promotional effort intensity function to represents current effect of dynamic marketing variables. All these models describe relationship between sales and time or promotional resources spent considering a single product existing independent of other products. In the literature few model have also been developed to describe the diffusion and substitution effect between the various generations of a product such as Norton [26], Norton-Bass model and some variation of the model [27,28], Mahajan and Muller [22], Kapur et al. [19] etc. Here we use the diffusion substitution model due to Kapur et al. [19] to describe the relationship between sales and promotional effect of multiple generation of the product and formulate a promotional allocation problem for a durable technology product in a segmented market. The optimization model formulated is a non-linear programming problem. It allocates promotional efforts to the different generations of a product in a planning horizon maximizing sales under budgetary constraint using the multi-objective nonlinear programming optimization method and solution is obtained with the help of LINGO software.

Rest of the paper is organized as follows: In section 2 we discuss the diffusion substitution sales growth model for describing the relationship between adoption and time for the successive generations of a product. In section 3 using the model promotional allocation model is formulated to optimally allocate the promotional efforts to the multiple generations of a product in a segmented market. Section 4 describes the solution methodology and a numerical is shown for the applicability of the model in section 5. Finally the paper is concluded in section 6.

2 Diffusion Substitution Model

Notations

$N(t)$	:	Expected sale of the base product by time t.
$N_i(t)$	:	Expected sales (or cumulative number of adopters) of the i^{th} generation by time t.
$N_i(t-t_j)$	:	Expected sales (or cumulative number of adopters) of the i^{th} generation in time interval $(t_j, t_{j+1}]$

t_{i-1} : Is the time of introduction of generation i.

a : Expected number of potential adopters of the base technological product.

ap_i : Expected potential market captured by i^{th} generation product

$X_{ij}(t)$: Cumulative promotional effort for the i^{th} generation in time interval $(t_j, t_{j+1}]$ with $x_{ij}(t)$ as its pdf

$b_{ij}(X_{ij}(t))$: Rate of diffusion per remaining potential adopters of generation i in time interval $(t_{j-1}, t_j]$.

a_{ij} : Expected remaining potential market to be captured by i^{th} generation product in time (t_{j-1}, t).

n : total number of generations of the product.

Time period $t_0(=0) \le t \le t_1$

Assuming that we start at the time point $t_0(=0)$ when first generation of a product is introduced in the market. In the time interval $t_0(=0) \le t \le t_1$ there is only one generation of the product, therefore overall sales of the product is same as the growth of adoption of generation one. Adoption rate equation following modified Bass model [14] for generation one in this time interval is given by

$$\frac{d}{dt}N_1(t) = \frac{d}{dt}N_1(t-t_o) = \frac{b_{11}}{1+\beta_{11}e^{-b_{11}X_{11}(t)}}\left[a\ p_1 - N_1(t-t_o)\right]x_{11}(t) \tag{1}$$

Solving equation (1) under the initial condition $t = 0, N_1(t-t_o) = 0$ expected number of adopters for generation one and overall sales for the product is given by

$$\begin{aligned} N(t) = N_1(t-t_0) &= ap_1\left(1-\left(\frac{1+\beta_{11}\exp^{-b_{11}X_{11}(t_0)}}{1+\beta_{11}\exp^{-b_{11}X_{11}(t)}}\right)\exp^{-b_{11}X_{11}(t-t_0)}\right) \\ &= a_{11}\left(\frac{1-\exp^{-b_{11}X_{11}(t)}}{1+\beta_{11}\exp^{-b_{11}X_{11}(t)}}\right) \end{aligned} \tag{2}$$

Time period $t_1 < t \le t_2$

In this time period, second generation of product is introduced in the market at time t_1, when the expected remaining potential of generation one is $a_{12} = a\ p_1 - N_1(t_1 - t_0)$ and expected potential of generation II is $a_{22} = a\ p_2$. As soon as generation II of the product is introduced a change in the adoption rate per remaining adopters of the generation I is observed. This change in the adoption rate per remaining adopter can be accounted due to the fact that generation II product have a substitution effect on the generation I. The substitution effect

brings changes in the adopter's adoption behavior, governing the diffusion of the base technology. Some of the potential adopters of the generation I who are to adopt the generation I product may instead buy the generation II. However the difference between the potential adopters of generation II and those who buy generation II product after adopting generation I for upgrading their technology can be avoided as they also represent the first purchaser of the generation II. Cannibalization is taken care implicitly due to changes in adoption rate. The resulting adoption rate equations in time interval $(t_1, t_2]$ are

$$\frac{d}{dt}N_i(t-t_1) = \frac{b_{i2}}{1+\beta_{i2}e^{-b_{i2}X_{i2}(t)}}\left[a_{i2} - N_i(t-t_1)\right]x_{i2}(t); \qquad i=1,2 \tag{3}$$

the expected number of adopters of both generations in time interval $(t_1, t_2]$ under the initial condition $t=t_1, N_1(t-t_1)=N_2(t-t_1)=0$ is obtained as

$$N_i(t-t_1) = a_{i2}\left(1-\left(\frac{1+\beta_{i2}\exp^{-b_{i2}X_{i2}(t_1)}}{1+\beta_{i2}\exp^{-b_{i2}X_{i2}(t)}}\right)\exp^{-b_{i2}X_{i2}(t-t_1)}\right); \qquad i=1,2 \tag{4}$$

Total expected number of adopters of the two generations and overall product growth by any time *t*, after the time t_1 are given by

$$N_1(t) = N_1(t_1-t_0)+N_1(t-t_1)\,,\; N_2(t)=N_2(t-t_1)\,,\; N(t)=N_1(t)+N_2(t) \tag{5}$$

Time period $t_2 < t \le t_3$

In this time period, third generation of product is introduced in the market at time t_2, when the expected remaining potential of generation one and two $a_{13}=a_{12}-N_1(t_2-t_1)$ and $a_{23}=a_{22}-N_2(t_2-t_1)$ respectively. In this time period, some of the potential adopters of the generation I who are to adopt the generation I product may instead adopt either of the generation II or III product. Potential adopters of the generation I who are adopting the generation III product, are leapfrogging the generation II. The adoption rate equations in time interval $(t_2, t_3]$ are

$$\frac{d}{dt}N_i(t-t_2) = \frac{b_{i3}}{1+\beta_{i3}e^{-b_{i3}X_{i3}(t)}}\left[a_{i3} - N_i(t-t_2)\right]x_{i3}(t); \qquad i=1,2,3 \tag{6}$$

the expected number of adopters of three generations in time interval $(t_2, t_3]$ under the initial condition $t=t_2, N_1(t-t_2)=N_2(t-t_2)=N_3(t-t_2)=0$ is obtained as

$$N_i(t-t_2) = a_{i3}\left(1-\left(\frac{1+\beta_{i3}\exp^{-b_{i3}X_{i3}(t_2)}}{1+\beta_{i3}\exp^{-b_{i3}X_{i3}(t)}}\right)\exp^{-b_{i3}X_{i3}(t-t_2)}\right); \qquad i=1,2,3 \tag{7}$$

and the total expected number of adopters of the three generations and over all product growth by any time *t*, after the time t_2 are given by

$$N_1(t) = N_1(t_1 - t_0) + N_1(t_2 - t_1) + N_1(t - t_2),\ N_2(t) = N_2(t_2 - t_1) + N_2(t - t_2)$$
$$N_3(t) = N_3(t - t_2),\qquad N(t) = N_1(t) + N_2(t) + N_3(t) \tag{8}$$

Time period $t > t_{n-1}$

In this time period, n[th] generation of product is introduced in the market at time t_{n-1}. Equation for expected cumulative number of adopters for i[th] generation in time interval (t_{n-1}, t) is given by

$$N_i(t - t_{n-1}) = a_{in}\left(1 - \left(\frac{1 + \beta_{i,n}\exp^{-b_{i,n}X_{i,n}(t_{n-1})}}{1 + \beta_{i,n}\exp^{-b_{i,n}X_{i,n}(t)}}\right)\exp^{-b_{i,n}X_{i,n}(t - t_{n-1})}\right) \tag{9}$$

Equation for expected cumulative number of adopters for i^{th} generation product by time t is given by

$$N_i(t) = \sum_{j=i}^{n-1} N_i(t_j - t_{j-1}) + N_i(t - t_{n-1}) \tag{10}$$

Equation for expected cumulative number of adopters of the base product by time t is given by

$$N(t) = \sum_{i=1}^{n} N_i(t) \tag{11}$$

3 Promotional Allocation Problem

In different time periods different number of generations exists in the market simultaneously. But usually two to three generations are seen to be operating in the market in any time period. For allocation the promotional budget here we start from the time period t_0 when only I[st] generation is operating. The problem for finding the optimal amount of promotional resources to be allocated to segment j in time period $(t_0, t_1]$ that would maximize the total sales under budget constraint is formulated as

$$\text{Maximize}\ \ G_1(X) = \sum_{j=1}^{M} N_{1j}(t - t_0) = \sum_{j=1}^{M} a_{11j}\left(\frac{1 - \exp^{-b_{11j}X_{11j}(t)}}{1 + \beta_{11j}\exp^{-b_{11j}X_{11j}(t)}}\right)$$
$$\text{Subject to}\ \ \sum_{j=1}^{M} X_{11j} \le Z_0 \tag{p1}$$
$$X_{11j} \ge 0 \qquad j = 1, \ldots, M$$

where M is the number of market segments for the generation one and Z_0 is the allocated budget for the time period $(t_0, t_1]$.

After time t_1 both generation one and two operate simultaneously in the market, so the promotional efforts should be readjusted for the time period till next generation is introduced. It can be noted here that for allocating promotional effort for the first generation we need to consider only the sales that can be captured in time period $(t_1, t_2]$ and the sales that has already been captured in the time $(t_0, t_1]$ can be ignored. So the allocation problem for the time period $(t_1, t_2]$ is formulated as

$$\text{Maximize} \quad G_2(X)$$
$$= \sum_{j=1}^{M} N_{1j}(t-t_1) + \sum_{j=1}^{M} N_{2j}(t-t_1) = \sum_{i=1}^{2}\sum_{j=1}^{M} a_{i2j}\left(1 - \left(\frac{1+\beta_{i2j}\exp^{-b_{i2j}X_{i2j}(t_1)}}{1+\beta_{i2j}\exp^{-b_{i2j}X_{i2j}(t)}}\right)\exp^{-b_{i2j}X_{i2j}(t-t_1)}\right)$$

$$\text{Subject to} \quad \sum_{j=1}^{M} X_{i2j} \le w_i Z_1; \qquad i = 1,2 \qquad \text{(p2)}$$
$$X_{i2j} \ge 0 \qquad i = 1, 2; \qquad j = 1, \ldots, M$$
$$w_1 + w_2 = 1; \quad w_i \ge 0; \; i = 1,2$$

where Z_1 is the allocated budget for the time period $(t_0, t_1]$ and w_i is proportion of budget allocated to ith generation. Proceeding in the similar manner say, when there are n generations of the product in the market by time period $t > t_{n-1}$ the allocation problem of allocation promotional resources to the n generations of the product is formulated as

$$\text{Maximize } G_n(X) = \sum_{i=1}^{n}\sum_{j=1}^{M} N_{ij}(t-t_{n-1}) = \sum_{i=1}^{n}\sum_{j=1}^{M} a_{inj}\left(1 - \left(\frac{1+\beta_{inj}\exp^{-b_{inj}X_{inj}(t_1)}}{1+\beta_{inj}\exp^{-b_{inj}X_{inj}(t)}}\right)\exp^{-b_{inj}X_{inj}(t-t_{n-1})}\right)$$

$$\text{Subject to} \quad \sum_{j=1}^{M} X_{inj} \le w_i Z_1; \qquad i = 1,2,\ldots,n \qquad \text{(p3)}$$
$$X_{inj} \ge 0 \qquad i = 1,2,\ldots,n \qquad j = 1, \ldots, M$$
$$\sum_{i=1}^{n} w_i = 1; \quad w_i \ge 0; \; i = 1,\ldots n$$

4 Solution Methodology

Cumulative adoption is a function of promotional efforts spent by that time. Also resources are continuously spent in the market and sales of the product increases. At the same time, the planning period for the product promotion is almost fixed. Therefore without loss of generality the cumulative adoption can be assumed to be a function of promotional efforts explicitly in the problems defined in section 3. At any time period the problem maximizes sum of one or more nonlinear functions subject to budgetary constraint and non negative restrictions. These functions can be expressed as

$$F_{ikj}(X_{ikj}) = N_{ij}(t - t_{k-1}) = a_{ikj}\left(1 - \left(\frac{1+\beta_{ikj}\exp^{-b_{ikj}X_{ikj})}}{1+\beta_{ikj}\exp^{-b_{ikj}X_{ikj}}}\right)\exp^{-b_{ikj}X_{ikj}}\right) \tag{12}$$

The functions $F_{ikj}(X_{ikj})$ can further be expressed as $F_{ikj}(X_{ikj}) = f_{ikj}(X_{ikj}) / g_{ikj}(X_{ikj})$ $i = 1,..,n,\ \ k = i,\ j = 1,\ldots M$. The derivatives of $f_{ikj}(X_{ikj})$ and $g_{ikj}(X_{ikj})$ and are ever non-increasing and non-decreasing functions of X_{ikj} respectively hence the functions $f_{ikj}(X_{ikj})$ and $g_{ikj}(X_{ikj})$ are concave and convex respectively. The ratio of concave and convex functions is pseudo-concave function and the sum of pseudo-concave functions is not necessarily a pseudo-concave function. Such a problem cannot be solved directly to obtain an optimal solution for such class of problems. Dur et. al. [6] has proposed a method to solve such class of problems converting sum of ratio functions of the objective to a multiple objective fractional programming problem. Dur's equivalent of the problem under consideration when there are *k* generations in the market can be written as

Maximize $\quad G_k(X) = \left(F_{ikj}(X_{ikj});\ i = 1,\ldots,k;\ j = 1,\ldots,M \right)^T$ (p4)

Subject to $X \in S = \left\{ X \in R^{kM} \Big/ \begin{array}{l} \sum_{j=1}^{M} X_{ikj} \le w_i Z_k,\ \ i=1,2,\ldots,k;\ \ X_{ikj} \ge 0,\ \ i=1,\ldots,k,\ j=1,\ldots,M; \\ \sum_{i=1}^{k} w_i = 1\ \ w_i \ge 0;\ \ i=1,\ldots k \end{array} \right\}$

Let $F_{ikj}(X_{ikj}) = f_{ikj}(X_{ikj}) / g_{ikj}(X_{ikj}) \ge y_{ikj}$ then an equivalent parametric problem for (p4) can further be written as following multi objective programming problem given by Bhatia *et al.* . [5]

Maximize $\quad y = \left(y_{ikj};\ i = 1,\ldots,k;\ j = 1,\ldots,M \right)^T$

Subject to $\quad f_{ikj}(X_{ij}) - y_{ikj} g_{ikj}(X_{ijk}) \ge 0$

$$\sum_{j=1}^{M} X_{ikj} \le w_i Z_k, \qquad i = 1,2,\ldots,k; \tag{p5}$$

$$X_{ikj} \ge 0, \quad i = 1,\ldots,k,\ \ j = 1,\ \ldots, M;$$

$$\sum_{i=1}^{k} w_i = 1; \qquad w_i \ge 0; \quad i = 1,\ldots k$$

Then equivalent (Geoffrion's [10]) scalarized formulation with suitable adjustment (i.e. taking both functions together having same variable) of the problem (p5) for fixed weights of the objective function is as follows

Maximize $\quad \sum_{i=1}^{k} \sum_{j=1}^{M} \lambda_{ikj} y_{ikj}$ (p6)

Subject to $f_{ikj}(X_{ij}) - y_{ikj} g_{ikj}(X_{ijk}) \geq 0$

$$\sum_{j=1}^{M} X_{ikj} \leq w_i Z_k, \qquad i = 1,2,...,k;$$

$$X_{ikj} \geq 0, \quad i = 1,...,k, \quad j = 1, \ldots, M;$$

$$\sum_{i=1}^{k} w_i = 1, \qquad w_i \geq 0; \quad i = 1,....k$$

$$\lambda \in \Omega = (\lambda \in R^{kM} / \sum \lambda_{ikj} = 1, \lambda_{ikj} \geq 0; \ i = 1,...,k, j = 1,...,M)$$

Now we state some lemma [4,5,6,10] and a theorem which would help us to obtain solution of the problem.

Lemma 1: The optimal solution of the problem (p3) for any $k = 1,\ldots n$ is an efficient solution of the problem (p4).

Lemma 2: A properly efficient solution $X^* = \{X_{ikj};\ i = 1,...k; j = 1,...M\}$ of the problem (p5) is also a properly efficient solution $X^* = \{X_{ikj};\ i = 1,...k; j = 1,...M\}$ for the problem (p4).

Lemma3: The optimal solution $\{X^*, Y^*\} = \{X_{ikj}, Y_{ikj},;\ i = 1,...k; j = 1,...M\}$ of the problem (p6) is a properly efficient solution $X^* = \{X_{ikj};\ i = 1,...k; j = 1,...M\}$ for the problem (p5).

Theorem 1: If relative importance is attached to each of the objective of the problem (P6) and $\{X^*, Y^*\} = \{X^*_{ikj}, Y^*_{ikj},;\ i = 1,...k; j = 1,...M\}$ is an optimal solution of the problem (P6) then $X^* = \{X_{ikj};\ i = 1,...k; j = 1,...M\}$ is an optimal solution for the problem (P3) for any $k = 1,\ldots n$.

Optimal solution of problem (P6) can be found out by assigning weights to different segments ($\lambda_{ikj} \geq 0;\ i = 1,...,k, j = 1,...,M$) and different generations ($w_i \geq 0;\ i = 1,....k$). These weights are decided either by top management or expert's judgment etc. Mathematical approach to facilitate decision making process could be by Analytical Hierarchy Process (AHP) which prioritizes goals or alternatives based on pair wise comparison or judgment. The purpose of assigning different weights is to prioritize the segments and generations according to which resources can be allocated. It remains to obtain an optimal solution of the problem (P6) by assigning these weights. Dur et al. has considered ratio functions only in the objective function whereas in problem (p3), ratio functions exist in the constraint functions as well and the equivalent formulation of the same has been considered. Our promotional resource allocation fits in the same class of problem. Problem (P6) can be solved by standard mathematical programming approach using LINGO [29] software if there exists a feasible solution to the problem (p3).

5 Numerical Illustration

The optimization model developed here can be applied to a product for which n generation exist in the market simultaneously. It is applicable to the situation when the product is launched anew in the market and subsequently many generations are introduced or the product is operating since a long time and we want to do the allocations starting from the ith generation. In this case we can assume that the planning period starts at the time of introduction of the ith generation i.e. treating is as t_0. Here we assume that generation one of the product is launched in the market at time t_0 and parameters for the first generation have already been estimated from some similar product history or otherwise these parameters an be estimated by observing the product for some reference period. Here we assume that the product is targeted to the six market segments and the parameters of the diffusion substitution model for the first generation in time period $t_0(=0) \le t \le t_1$ are as given in columns 2, 3 and 4 of table 1. The promotional allocations are made for the six segments for the first generation of the product with a budget of `70,000 following the problem (p1). Here we assume that equal weights (λ) are assigned to each segment for the sake of illustration although the solution procedure remains same if actual weights are assigned. The allocations made to each segment and expected the potential market capture due to this allocation and remaining potential is shown in column 5, 6 and 7 of table 1.

Table 1. Data and results of the optimization problem in time period $t_0(=0) \le t \le t_1$

Generation -1 in time $t_0(=0) \le t \le t_1$						
Segment	a_{11j}	b_{11j}	β_{11j}	X^*_{11j}	$a_{11j}(X^*_{11j})$	Remaining potential
S1	35000	0.00031910	26.55565	11959	21603	13397
S2	46000	0.00036375	27.21550	13800	38732	7268
S3	45000	0.00032134	26.54803	13876	34024	10976
S4	64000	0.00034644	28.11991	12786	29602	34398
S5	28000	0.00040125	29.12487	7500	9755	18245
S6	31000	0.00041145	25.10141	10079	21839	9161
Total	249000			70000	155555	93445

At the time period t_1 2nd generation of the product is introduced in the market. At this time generation one continue to serve the market but due to the introduction of the 2nd generation its remaining potential may decrease due to the switching of some of its potential customer to the second generation product, but it may also see market expansion due to enlargement of the potential customer size on the account of effectiveness of promotion. Now we can re-estimate the remaining potential of the generation one i.e. a_{12} and other parameters of both generations for the period $t_1 < t \le t_2$, forgoing the market potential for the generation one which

have already been captured in time $t_0(=0) \le t \le t_1$ using the total cumulative adoption equation for the time period $t_1 < t \le t_2$ for both generations i.e.

$$N(t-t_1) = N_1(t-t_1) + N_2(t-t_1) \tag{13}$$

Now for estimating these parameters we can use the data obtained from period $t_0(=0) \le t \le t_1$ for the first generation and using similar product history. The values of the parameters of the sales growth model in time period $t_1 < t \le t_2$ are assumed to be known and are as given in columns 1,2 and 3 of table 2 and 3 for generation one and two respectively. The promotional allocations are made in the six segments for the both generation one and two with a budget of `1,60,000 following the problem (p2). The promotional budget is divided among the two generations in the ration 4.5:5.5, i.e. in w_1=0.45 and w_2=0.55. The allocations made to each segment and expected the potential market capture due to this allocation and remaining potential is shown in column 5, 6 and 7 of table 2 and 3 for generation one and two respectively.

Table 2. Data and results of the problem for generation one in time period $t_1 < t \le t_2$

Generation one in time $t_1 < t \le t_2$						
Segment	a_{12j}	b_{12j}	β_{12j}	X^*_{12j}	$a_{12j}(X^*_{12j})$	Remaining potential
S1	13960	0.000376538	32.397893	13498	9734	4226
S2	7573	0.000429225	33.20291	7445	5207	2366
S3	11437	0.000379185	32.3885966	12249	6667	4770
S4	35843	0.0004088	34.3062902	17864	31710	4133
S5	19012	0.000473471	35.5323414	13625	15777	3235
S6	9537	0.000485511	30.6237202	7319	7614	1923
Total	97362			72000	76709	20653

Table 3. Data and results of the problem for generation two in time period $t_1 < t \le t_2$

Generation two in time $t_1 < t \le t_2$						
Segment	a_{22i}	b_{22i}	β_{22j}	X^*_{22j}	$a_{22j}(X^*_{22j})$	Remaining potential
S1	25000	0.00024860	30.6327	16444	16094	8906
S2	32000	0.00030980	34.2978	16593	26349	5687
S3	40000	0.00049810	30.8723	11991	36847	3153
S4	37000	0.00035930	27.1878	14850	32415	4585
S5	41000	0.00032140	28.1432	16656	35867	5133
S6	35000	0.00048230	25.1795	11466	31692	3308
Total	210000			88000	179264	30736

The promotional allocations can be made similarly for the future time periods on the introduction of new generations of the product**.**

6 Conclusions

In this paper we have formulated an optimization model for promotional allocation problem for a durable technology product in a segmented market. Multiple generations of such a product can exist simultaneously in the market so a diffusion substitution model is used to describe the relationship between sales and promotion over the planning horizon. We have also discussed the solution procedure and presented a numerical application of the model to show the applicability of the model in real life. Similar models discussed in the literature doesn't consider the effect on sales of an existing generation when a new generation of the product is introduced in the market and effect on sales of each other due to the existence of multiple generations simultaneously. Our model overcomes this disadvantage by using a technology substitution diffusion model to describe the sales of multiple generations of a product and allocating promotional efforts jointly to multiple generations with a single optimization model. The model offers a big scope of further research. The model formulated here considers only the budget constraint. It can further be extended for other system constraints and multiple products etc.

Acknowledgements. One of the authors Yogender Singh gratefully acknowledge the financial support of the Council of Scientific and Industrial Research, New Delhi, India through his Junior Research Fellowship (JRF) scheme CSIR Award no.: 09/045(0663)/2006-EMR-I) for his research work.

References

[1] Bass, F.M.: A new product growth model for consumer durables. Management Sciences 15, 215–227 (1969)
[2] Bass, F.M., Krishnan, T.V., Jain, D.C.: Why the Bass model fits without decision variables. Marketing Science 13(3), 203–223 (1994)
[3] Bass, P.I.: Generational diffusion model data: The IBM mainframe case (2003), http://www.basseconomics.com
[4] Bector, C.R., Chandra, S., Bector, M.K.: Generalized fractional programming, duality: A parametric approach. Journal of Optimization Theory and Applications 60, 243–260 (1989)
[5] Bhatia, D., Kumar, N., Bhudhiraja, R.K.: Duality theorem for non differential multiobjective programs. Indian Journal of Pure and Applied Mathematics 28(8), 1030–1042 (1997)
[6] Dur, M., Horst, R., Thoai, N.V.: Solving sum of ratios fractional programs, using efficient points. Optimization 1, 447–466 (2001)
[7] Easingwood, C., Mahajan, V., Muller, E.: A non-uniform influence innovation diffusion model for new product acceptance. Marketing Science 2, 273–295 (1983)
[8] Fisher, J.C., Pry, R.H.: A simple substitution model for technological change. Technological Forecasting and Social Change 3, 75–88 (1971)

[9] Fourt, L.A., Woodlock, J.W.: Early prediction of Market Success for grocery products. Journal of Marketing 25, 31–38 (1960)
[10] Geoffrion, A.M.: Proper efficiency and theory of vector maximization. Journal Of Mathematical Analysis and Applications 22, 613–630 (1968)
[11] Horsky, D., Simon, L.S.: A diffusion model incorporating product bene-fits, price, income and information. Marketing Science 9, 342–385 (1990)
[12] Horsky, D., Simon, L.S.: Advertising and diffusion of new products. Marketing Science 2(1), 1–17 (1983)
[13] Jha, P.C., Mittal, R., Singh, O., Kapur, P.K.: Allocation of promotional resource for multi-products in a segmented market. Presented in ICORAID-2005-ORSI and Published in Proceedings (2005)
[14] Jha, P.C., Anshu, G., Kapur, P.K.: Bass model revisited. Journal of Statistics and Management Systems 11(3), 413–437 (2006)
[15] Jha, P.C., Anshu, G., Kapur, P.K.: On maximizing the adoption of multi-products in multiple market segments. Journal of Information and Optimization Sciences 28(2), 219–245 (2007)
[16] Kalish, S.: A new product adoption model with pricing, advertising and uncertainty. Management Science 31, 1569–1585 (1985)
[17] Kapur, P.K., Aggarwal, A.G., Kaur, G., Basirzadeh, M.: Optimising adoption of a single product in multi-segmented market using innovation diffusion model with consumer balking. Int. J. Technology Marketing 5(3), 234–249 (2010)
[18] Kapur, P.K., Bardhan, K.A., Jha, P.C.: An alternative formulation of innovation diffusion model and its extension. In: Kapoor, V.K. (ed.) Mathematics and Information Theory: Recent Topics and Applications, Anamaya Publishers, New Delhi (2004)
[19] Kapur, P.K., Anshu, G., Jha, P.C., Kumar, U.D.: Adoption of newer successive technologies. In: Kapur, P.K., Verma, A.K. (eds.) Quality Reliability & Infocom Tecnology. Macmillan Advance Research Series, pp. 10–25 (2007)
[20] Kapur, P.K., Jha, P.C., Bardhan, A.K., Singh, O.: Allocation of promotional resource for a new product in a segmented market. In: Kapoor, V.K. (ed.) Mathematics and Information Theory: Recent Topics and Applications, Anamaya Publishers, New Delhi (2004)
[21] Kotler, P., Armstrong, G.: Principles of Marketing, 6th edn. Prentice Hall of India Pvt. Ltd., New Delhi (1995)
[22] Mahajan, V., Muller, E.: Timing, Diffusion and Substitution of Successive Generations of Technological Innovations: The IBM Mainframe Case. Technological Forecasting and Social Change 51, 109–132 (1996)
[23] Mahajan, V., Muller, E.: Innovation diffusion and new product growth models in marketing. Journal of Marketing 43, 55–68 (1979)
[24] Mahajan, V., Peterson, R.A.: Innovation diffusion in a dynamic potential adopter population. Management Science 24, 1589–1597 (1978)
[25] Mahajan, V., Peterson, R.A.: Models for innovation diffusion. Sage Publications, California (1985)
[26] Norton, J.A.: Growth, Diffusion and Technological Substitution in Industrial Markets: An Examination of the Semiconductor Industry. Doctoral Dissertation. University of Texas at Dallas (1986)

[27] Norton, J.A., Bass, F.M.: A diffusion theory model of adoption and substitution for successive generations of high-technology products. Management Science 33(9), 1069–1086 (1987)
[28] Norton, J.A., Bass, F.M.: Evolution of Technological Generations: The Law of Capture. Sloan Management Review 33(2), 66–77 (1992)
[29] Thirez, H.: OR software LINGO. European Journal of Operational Research 124, 655–656 (2000)

Single-Source, Single-Destination Coordination of EOQ Model for Perishable Products with Quantity Discounts Incorporating Partial/Full Truckload Policy under Fuzzy Environment

Sandhya Makkar[1], P.C. Jha[1], and Nisha Arora[2]

[1] Department of Operational Research, University of Delhi, Delhi, India
[2] AMITY Business School, AMITY University, Gurgaon, India
{bajajsandhya,nishaarora4}@gmail.com, jhapc@yahoo.com

Abstract. A supply chain may be considered as a group of organizations, connected by a series of trading relationships. This group covers the logistics and manufacturing activities from raw materials to the final consumer. Each organization in the chain procures and then transforms materials into intermediate/final products, and distributes these to customers. Thus, the supply chain is a network linking and interweaving different supply chains of all the companies involved in a production process. The content of supply chain management with in a firm varies considerably with the type of business. Procurement and distribution of products of any business are the two vital functions. In this paper, an attempt to integrate these two functions is done. It incorporates quantity discounts and freight policies for multi items that are perishable in nature, which are supplied from single source to single destinations under fuzzy environment. It is assumed that, single source or a supplier is offering various discounts on the purchase of quantity and are supplying to a destination. A transporter hired by the buyer also offers different policies to choose on distribution of products, thereby buyer is enticed by dual benefit on procurement as well as distribution. A fuzzy optimization problem is converted to crisp mathematical programming problem using triangular membership function. The problem is solved and a case is presented to validate the model.

Keywords: Discount schemes, Transportation cost, Perishable items, Lingo, Fuzzy Programming.

1 Introduction

Over the past few years prevailing economic condition have made the companies around the world more customer centric. Customer satisfaction and service have become the highest priority [5]. Supply chain management has come up as a source of gaining competitive advantage in the business world. Organizations are

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 971–982.
springerlink.com

grappling with identifying and improving the strategic issues that their supply chain should cater to [12]. Due to pressure from increased competition, globalization of supply and networks, corporate restructuring, introduction of new manufacturing methods, high level of services, as well as low price expectations, the importance of supply chain will continue to intensify [13]. Suppliers frequently offers discount schedules based on all units model, which divides the range of possible order quantities into intervals with progressively lower unit costs [11]. Transporters these days are looking for more innovative ways to attract customers by thriving to satisfy their ever increasing demand of cost effective transportation by offering freight policies such as truck load and less than truck load. In the supply chain management literature, there are several articles that address about quantity discounts and freight cost. Least has been said about perishability/deterioration of stock (which is the main factor in determining the productivity of stock) and its implication on the procurement and distribution of products. Following factor may lead to deterioration of stock viz. damage, spoilage, obsolescence, decay, decreasing usefulness etc. The first model where the factor of perishability was considered as noteworthy is projected by [6] who pointed out that inventory decay could exert a significant impact on the overall inventory cost, while considerable cost saving could be achieved if the inventory analysis took the inventory decay into consideration. [2] finds the replenishment schedule for an inventory system with shortages, in which items deteriorate at a constant rate and demand rates are decreasing over a known and finite planning horizon. [10] built up an inventory model for perishable items with limited storage space. They took demand rate for the items are finite; items deteriorate at constant rates and are replenished instantaneously. All the above authors studied about constant rate of deteriorating items but none of them included quantity discounts and freight policies.

The few studies performed on the subject of quantity discounts and freight also reveal interesting results. [7] quoted that quantity discounts provide a practical foundation for inventory coordination in supply chains. [9] analyzed that Multi-site organizations must balance conflicting forces to determine the appropriate degree of purchasing centralization for their respective supplies. The ability to garner quantity discounts represents one of the primary reasons that organizations centralize procurement. [8] examined that if both the buyer and supplier can find a coordination mechanism to make joint decisions, the joint profit in this situation is more than the sum of their profits in the decentralized decision situation. They show that quantity discount policy is a way that may be implemented to achieve coordination. [3] emphasized that, transportation costs has enhanced the need to develop models with transportation consideration explicitly.

With increasing competition, the need for coordination of different entities crop up to minimize the cost of supply chain. [3] developed an unconstrained integrated inventory-transportation model to decide optimal order quantity for inventory system over a finite horizon. [4] took a single stage multi incapacitated dynamic lot sizing problem (MILSP) with transportation cost and assumed finite planning horizon with dynamic demand.

In this study, we develop an integrated procurement-distribution model where multiple perishable products are ordered by a single buyer from a single supplier who offers discounts on the purchase of bulk buying and hires a distributer that also offers well known schemes *viz.* Truckload(TL), Less-than-Truckload(LTL) or combination of TL & LTL. In TL transportation; there is a fixed cost per load up to a given capacity. With TL transportation, a company may use less than the capacity available and transport this freight at the cost of a full load. However, in some cases the quantity of the freight may not be large enough to justify the cost associated with a TL shipment. For these small quantities, a LTL carrier may be used. LTL may be defined as a shipment which does not completely fill a truck or which weighs less than the weight required for the application of the TL freight rate or it takes the weight between partial and full truck load at constant unit transportation cost [1]. Also, since the demand cannot be identified accurately ever, so we have supposed fuzziness in demand and total cost of the model.

The rest of the paper is as follows. Second section describes the notations and symbols used in model following with assumptions. The next section further describes the initial analysis and constructs fuzzy objective and price breaks. Section 4 covers the fuzzy optimization formulation of model together with solution algorithm. In section 5, a case of real life problem has been discussed to testify and validate the procedure of the models. The data has been presented in modified form for limited period due to cutting edge competition. Finally, Section 6 concludes the research work with some discussions.

2 Model Assumptions, Symbols and Notations

2.1 Assumptions

1) Demand is dynamic 2) Supply is instantaneous 3) Lead time is zero
4) Initial inventory of each product for the beginning of planning horizon is zero.
5) Constant rate of deterioration for perishable items, as a percentage of defective items of stored units. 6) Constant rate of inspecting the multiple perishable items.

2.2 Sets

- Product set with cardinality P and indexed by i.
- Period set with cardinality T and indexed by t.
- Item discount break point set with cardinality L and indexed by small l.
- Destination set with cardinality M and indexed by m.

2.3 Decision Variables

X_{it} Amount of product *i* ordered in period *t*

R_{ilt} If the ordered quantity falls in l^{th} price break then the variable takes value 1 otherwise zero

$$R_{ilt} = \begin{cases} 1 & if\ X_{it}\ falls\ \text{in}\ l^{th}\ pricebreak \\ 0 & \text{otherwise} \end{cases}$$

I_{it} Inventory level at the end of period t for product i

IN_i Inventory level at the beginning of planning horizon for product i

δ_t Total products ordered (transported in weights) in period t

α_t Total number of truckloads in period t

y_t Amount in excess of truckload capacity (in weights)

u_t The variable u_t (or, 1-u_t) reflects usage of policies, either TL & LTL policies or only TL policy or only LTL i.e.

$$u_t = \begin{cases} 1, \text{ if considering TL \& LTL both policies or LTL only} \\ 0, \text{ if considering only TL policy} \end{cases}$$

2.4 Parameters

$\tilde{C}$ Fuzzy total cost

C_0 Aspiration level of fuzzy total cost

C_0^* Tolerance level of fuzzy total cost

$\tilde{D}_{it}$ Fuzzy demand for product i in period t

$\bar{D}_{it}$ Defuzzfied demand for product i in period t

h_i Inventory holding cost per unit of product i

w_i Per unit weight of product i

ϕ_{it} Unit purchase cost for i^{th} product in t^{th} period

β_t Fixed freight cost for each truckload in period t

d_f Slab for discounts (price breaks)

ω Weight transported in each full truck (in kgs.)

s Cost/kg of shipping in LTL policy

a_{ilt} Limit beyond which l^{th} price break becomes valid in period t for product i

d_{ilt} It reflects the fraction of regular price that the buyer pays for purchased products.

m_i Rate of inspection of i^{th} item

η Percentage of defective items of the stored units

3 Initial Analysis

The inventory level at period t is dependent upon X_{it} and at fuzzy demand $\tilde{D}_{it}$ in the following way:

$$I_{it} = I_{it-1} + X_{it} - \tilde{D}_{it} - \eta I_{it} \quad where\ i = 1...P \ , \ t = 2...T \tag{1}$$

The inventory level at the end of the period 1 for item type i is composed of the inventory level at the beginning of the planning horizon, and the net change at the end of period one,

$$I_{i1} = IN_i + X_{i1} - \tilde{D}_{i1} - \eta I_{i1} \quad where\ i = 1...P \tag{2}$$

Total fuzzy demand in all the periods is less than or equal to total ending inventory level and ordered quantity of all the periods.

$$(1 - \eta) \sum_{t=1}^{T} I_{it} + \sum_{t=1}^{T} X_{it} \geq \sum_{t=1}^{T} \tilde{D}_{it}, \quad where\ i=1...P \tag{3}$$

The buyer will order minimum quantity a_{ilt} to get discount i.e.

$$X_{it} \geq \sum_{l=1}^{L} a_{ilt} R_{ilt} \ where\ i = 1...P; t = 1...T \tag{4}$$

It shows that the order quantity of all items in period t exceeds the price break threshold.

In any period, exactly one level will be activated either discount or no discount situation therefore,

$$\sum_{l=1}^{L} R_{ilt} = 1 \ \ where\ i = 1...P; t = 1...T \tag{5}$$

Transported quantity according to item weightage is:

$$\delta_t = \sum_{i=1}^{P} \left[w_i X_{it} \sum_{l=1}^{L} R_{ilt} \right] where\ t = 1, ..., T \tag{6}$$

Above constraint is an integrated constraint for procurement and distribution.

The minimum weighted quantity transported is equal to:

$$\delta_t \leq (y_t + \alpha_t \omega) u_t + (\alpha_t + 1)\omega(1 - u_t) \quad where\ t = 1...T \tag{7}$$

Overhead units from truckload capacity in weights are:

$$\delta_t = (y_t + \alpha_t \omega) \quad where\ t = 1...T \tag{8}$$

3.1 Construction of Objective Function and Price Breaks

In a real-life situation for strategic decisions of supply chain problem many input information, related to the various products are not known with certainty. Vagueness in the critical information cannot be captured in a deterministic problem and therefore the optimal results of these deterministic formulations may not serve the real purpose of modeling the problem. Due to this, we have considered the model

with fuzzy objective function. The objective is to minimize the cost incurred in holding inventory, inspection cost, purchasing the items and transportation cost under fuzzy environment.

$$Min\ \tilde{C}= \sum_{t=1}^{T}\sum_{i=1}^{P}\left\{h_i I_{it} + \mathrm{m_i X_{it}} + \sum_{l=1}^{L} R_{ilt} d_{ilt} \phi_{it} X_{it}\right\} + \sum_{t=1}^{T} (sy_t + \alpha_t \beta_t) u_t + (\alpha_t + 1)\beta_t (1-u_t)$$

Price breaks are defined as:

$$d_f = \begin{cases} d_{ilt} & a_{ilt} \le X_{it} \le a_{il+1t} \\ d_{iLt} & X_{it} \ge a_{iLt} \end{cases}$$

$$i = 1, ..., P; \quad l = 1, ..., L; \quad t = 1, ..., T$$

Lower limit of first price break in the models is zero and after the upper limit of first break buyer will get discount.

4 Fuzzy Optimization Model Formulation

Most of our traditional tools of modeling are crisp, deterministic, and precise in character. But for many practical problems there are incompleteness and unreliability of input information. This is caused to use fuzzy optimization method with fuzzy parameters. Crisp mathematical programming approaches provide no such mechanism to quantify these uncertainties. Fuzzy optimization is a flexible approach that permits more adequate solutions of real problems in the presence of vague information, providing the well defined mechanisms to quantify the uncertainties directly. Therefore, we formulate fuzzy optimization model for on vague aspiration levels on total cost and demand, the decision maker may decide his aspiration levels on the basis of past experience and knowledge possessed by him.

$$Min\ \tilde{C}= \sum_{t=1}^{T}\sum_{i=1}^{P}\left\{h_i I_{it} + \mathrm{m_i X_{it}} + \sum_{l=1}^{L} R_{ilt} d_{ilt} \phi_{it} X_{it}\right\} + \sum_{t=1}^{T} (sy_t + \alpha_t \beta_t) u_t + (\alpha_t + 1)\beta_t (1-u_t)$$

$$I_{it} = I_{it-1} + X_{it} - \tilde{D}_{it} - \eta I_{it} \quad \text{where } i = 1...P \ , \ t = 2...T \tag{1}$$

$$I_{i1} = IN_i + X_{i1} - \tilde{D}_{i1} - \eta I_{i1} \quad where\ i = 1, ..., P \tag{2}$$

$$(1-\eta)\sum_{t=1}^{T} I_{it} + \sum_{t=1}^{T} X_{it} \ge \sum_{t=1}^{T} \tilde{D}_{it}, \ \text{where } i = 1, ..., P \tag{3}$$

$$X_{it}, I_{it}, \delta_t, \alpha_t, y_t \ge 0 \ and\ integer;\ R_{ilt}, u_t \in \{0,1\}\ i = 1...P; t = 1...T; l = 1...L$$

$$X_{it} \ge \sum_{l=1}^{L} a_{ilt} R_{ilt} \quad where\ i = 1...P \ \ t = 1...T \tag{4}$$

$$\sum_{l=1}^{L} R_{ilt} = 1 \ \ where\ i = 1...P, t = 1...T \tag{5}$$

$$\delta_t = \sum_{i=1}^{P}\left[w_i X_{it} \sum_{l=1}^{L} R_{ilt} \right], \text{ where } t = 1...T \tag{6}$$

$$\delta_t \le (y_t + \alpha_t \omega)u_t + (\alpha_t + 1)\omega(1 - u_t) \quad \text{where } t = 1,...,T \tag{7}$$

$$\delta_t = (y_t + \alpha_t \omega) \text{ where } t = 1...T \tag{8}$$

4.1 Solution Algorithm

Following algorithm [14] specifies the sequential steps to solve the fuzzy mathematical programming problems.

1. Compute the crisp equivalent of the fuzzy parameters using a defuzzification function. Same defuzzification function is to be used for each of the parameters. Here we use the defuzzification function of the type where a^1, a^2, a^3 are the triangular fuzzy numbers.

$$F_2(A) = \left(a^1 + 2a^2 + a^3 \right) \Big/ 4$$

Let $\bar{D}_{it}$ be the defuzzified value of $\tilde{D}_{it}$ and $(D_{it}^1, D_{it}^2 \text{ and } D_{it}^3)$ be triangular fuzzy numbers then,

$$\bar{D}_{it} = \frac{D_{it}^1 + 2D_{it}^2 + D_{it}^3}{4} \text{ where } i = 1...P; t = 1...T$$

$\tilde{D}_{it}$ and C_0 are defuzzified aspiration levels of model's demand and cost.

2. Define appropriate membership functions for each fuzzy inequalities as well as constraint corresponding to the objective function. The membership function for the fuzzy is given as

$$\mu_C(X) = \begin{cases} 1 & ; C(X) \le C_0 \\ \dfrac{C_0^* - C(X)}{C_0^* - C_0} & ; C_0 \le C(X) < C_0^* \\ 0 & ; C(X) > C_0^* \end{cases}$$

3. Where C_0 is the restriction and C_0^* is the tolerance levels to the fuzzy total cost. Employ extension principle to identify the fuzzy decision, which results in a crisp mathematical programming problem and on substituting the values for $\tilde{D}_{it}$ as $\bar{D}_{it}$ and $\mu_C(x)$ the problem becomes given by

Maximize θ

subject to: $\mu_C(X) \ge \theta$,

$$I_{it} = I_{it-1} + X_{it} - \bar{D}_{it} - \eta I_{it} \quad \text{where } i = 1,...,P \ , \ t = 2,...,T \tag{1}$$

$$I_{i1} = IN_i + X_{i1} - \bar{D}_{i1} - \eta I_{i1} \quad \text{where } i = 1,...,P \tag{2}$$

$$(1-\eta)\sum_{t=1}^{T} I_{it} + \sum_{t=1}^{T} X_{it} \geq \sum_{t=1}^{T} \bar{D}_{it}, \text{ where } i = 1...P \tag{3}$$

$$X \in S = \{X_{it}, I_{it}, \delta_t, \alpha_t, y_t \geq 0 \text{ and integer}; R_{ilt}, u_t \in \{0,1\}$$

$$\text{/Satisfying eq.(4) to (8); } i = 1...P; t = 1...T; l = 1...L\}; \theta \in [0,1]$$

can be solved by the standard crisp mathematical programming algorithms.

5 Case Study

As the society today getting health conscious day by day, Mr Beans owner of leading restaurant in the city "The Bean's" decided to en-cash this prevailing health phenomenon by introducing "Whole Grain" meals in his restaurant. He decided to keep the items made from "Whole Grain" in the menu. With the concept in mind he introduced dishes made from three whole grains: Wheat, Maize and Ragi. To keep the quality of the food to max he decided to open a small storage and processing unit in his restaurant only.

After a thorough search he zeroed down on grain mill "Modi mills" which can provide all the three whole grains to "The Beans" in a raw form, after lengthy negotiations it was decided that Modi mills will supply the grains per sack, Wheat 20 kg, Maize 15 kg and Ragi 10 kg each, supplier also decided to offer various quantity discounts to the beans in order to bag max order from their customers as ordered qty will be very vague due to uncertain demand of their product.

Mr Beans also hired the transport services of "Delta Transporters" to transport the grains from the warehouse of modi mills to the processing the centre of the restaurant. Transporters also offered various schemes (like Truckload, less-than Truckload or a combination) to suit the needs of the customer.

Beans was also faced with the herculean task of minimizing the holding cost, as special environment is needed to store the grains in storage house and also minimizing inspection cost due to perishable nature of the grains which tend to decay of almost 10% of grains per sack, therefore periodic inspections are needed to keep decay of the grains to minimum.

To sum up, Mr. Beans was entrusted with the task to minimize restaurant's total cost that includes inspection, holding, purchase and transportation cost. Also the relevant data is provided as under(Table 1 to Table 6):

Table 1. Purchase Cost for i[th] Product in t[th] Period (in $)

Product/Period	1	2	3	4
Wheat	400	420	430	435
Maize	390	380	370	375
Ragi	445	435	425	450

Table 2. Discounts Factors (DF) for Products in all Periods

Wheat(in Kg)	DF	Maize(in Kg)	DF	Ragi(in Kg)	DF
$0 \le X_{it} < 40$	1	$0 \le X_{it} < 30$	1	$0 \le X_{it} < 35$	1
$40 \le X_{it} < 80$	0.98	$30 \le X_{it} < 60$	0.95	$35 \le X_{it} < 70$	0.96
$80 \le X_{it} < 160$	0.96	$60 \le X_{it} < 90$	0.90	$70 \le X_{it} < 100$	0.92
$160 \le X_{it}$	0.94	$90 \le X_{it}$	0.85	$100 \le X_{it}$	0.88

Table 3. Holding Cost for i[th] Product (in $)

Product	Wheat	Maize	Ragi
Cost	65	90	100

Table 4. Per Sack Weight for i[th] Product (in Kg)

Product	Wheat	Maize	Ragi
Weight	20	25	10

Table 5. Freight Cost in each Period (in $)

Period	1	2	3	4
Cost	2300	2500	2900	2550

Table 6. Inspection Cost of each Product (in $)

Product	Wheat	Maize	Ragi
Cost	40	40	45

Weight transported in each full vehicle ω= 2000 kgs;

Cost of shipping one unit in LTL mode of transportation, s=$2

The various cost and demand parameters C_0 and $\tilde{D}_{it}$ resp. are the triangular fuzzy numbers represented as $A = (a_l, a, a_u)$. The value of these fuzzy numbers are specified by the management based on the past experiences and\or expert opinion. The values of these fuzzy parameters are assumed and are tabulated in table 7.

Using the defuzzification function $F_2(A) = (a_1 + 2a + a_u)/4$ we defuzzify these fuzzy numbers. Defuzzified values of these parameters are also given in table 7.

Using the values of the fuzzy parameters as given in Table 7 and substituting in the defuzzification values in above problem, we obtained the following solution of the Case Study.

Table 7. Fuzzy Demand Parameters

Fuzzy Parameter (A)	a_l	a	a_u	Defuzzified value ($\bar{D}_{it}$)
$\tilde{D}_{11}$	30	35	36	34
$\tilde{D}_{12}$	55	56	49	54
$\tilde{D}_{13}$	90	98	90	94
$\tilde{D}_{14}$	87	91	91	90
$\tilde{D}_{21}$	98	97	92	96
$\tilde{D}_{22}$	72	76	80	76
$\tilde{D}_{23}$	45	40	35	40
$\tilde{D}_{24}$	78	81	80	80
$\tilde{D}_{31}$	52	58	60	57
$\tilde{D}_{32}$	70	65	68	67
$\tilde{D}_{33}$	69	68	63	67
$\tilde{D}_{34}$	38	36	30	35

The aspiration level of total cost is C_0=\$320,000 and tolerance level of cost is C* = \$480,000

5.1 Case Solution

Mr. Beans is able to estimate the ordered quantity for all the products of every period for his restaurant. For Wheat, he will order 88 sacks in first period, nothing in second period, 94 sacks in third period and 90 sacks in last period. Thereby he will have to store extra 54 sacks in first period which he will use in second period. He will receive a discount of 4% in first, third and fourth period.

Regarding Maize, he orders 96, 109, 10 and 80 sacks in four periods thereby after fulfilling demand left with only 30 sacks in period two. He will receive a discount of 15% in first and second periods, no discount in third period and 10% in last period. And for Ragi, his order sizes of sacks are 57, 67, 100 and 5 in four periods resp, therby getting discounts of 4% in first and second periods, 12% in third period and no discount in last period.

As ordersizes of sacks decides the weighted quantity for distribution, so weighted quantity of all the grains in every period is 3,770 kg; 2,305kg; 3,030kg and 3,050 kgs resp. In first period TL policy will be used and 2 trucks will ship the whole quantity. In second, third and fourth periods both TL & LTL policies will be used. In each of these periods 1 full truck will go and rest amount of 305 kg, 1,030 kg; and 1,050 kgs will use LTL policy for distributing the grains.

As far as the vagueness is concerned, Mr. Beans able to minimize the uncertainity by more than 75% and incurring a total cost of $359,405.

6 Conclusion

During the last three decades Supply Chain Management has emerged as important and productive aim of organizations. It coordinates production, shipment and delivery of goods required to meet business needs. SCM becomes utmost important for organization where the nature of goods they are dealing with is perishable. The study was on multi-items (ordered from a single source to single destination) that can perish with passing time with supplier and transporter offering quantity discounts and freight policies. There are many potential advantages of quantity discounts with transported policies. Supplier may ask for higher prices when buyer is not ordering full truckload. In that case discounted schemes on purchasing and different transporting policies may be helpful to the buyer. Based on the optimal response of the buyer, we analyze and applied schemes of supplier and transporters. We also provide a case which validated the formulated problem under fuzzy environment.

References

[1] Mendoza, A., Ventura, J.A.: Incorporating quantity discounts to the EOQ model with transportation costs. Int. J. of Prod. Econ. Amst. 113(2), 754 (2008)

[2] Lakdere, B.: On an inventory model with deteriorating items and decreasing time-varying demand and shortages. European Journal of Operational Research 86, 293–299 (1995)

[3] Darwish, M.A.: Joint determination of order quantity and reorder point of continuous review model under quantity and freight rate discounts. Computers & Operations Research 35(12), 3902–3917 (2008)

[4] Ertogral, K., Darwish: Multi-item single source ordering problem with transportation cost: A Lagrangian decomposition approach. European Journal of Operations Research 191, 154–163 (2008)

[5] Loukmidis, G., Hartweg, E.: Supply chain management: Conception and system support. Emerging Issues on Supply Chain Management, 24–32 (2004)

[6] Ghare, P.M., Shrader, G.F.: A model for exponentially decaying inventories. Journal of Industrial Engineering 14, 238–243 (1963)

[7] Shina, H., Benton, W.C.: A quantity discount approach to supply chain coordination. European Journal of Operational Research 180(2), 601–616 (2007)

[8] Li, J., Liu, L.: Supply chain coordination with quantity discount policy. Int. Journal of Production Economics 101(1), 89–98 (2006)

[9] Munsona, C.L., Hub, J.: Incorporating quantity discounts and their inventory impacts into the centralized purchasing decision. European Journal of Operational Research 201(2), 581–592 (2010)
[10] Mandala, N.K., Roy, T.K., Maitic, M.: Inventory model of deteriorated items with a constraint: A geometric programming approach. European Journal of Operational Research 173, 199–210 (2006)
[11] Rubin, P.A., Benton, W.C.: Evaluating jointly constrained order quantity complexities for incremental discounts. European Journal of Operational Research 149, 557–570 (2003)
[12] Sahay, B.S.: Section – I, Supply Chain: Strategic Issues. Emerging Issues on Supply Chain Management (2004)
[13] Sahay, B.S., Mohan, R.: Emerging Issues on Supply Chain Management, pp. 3–23. Macmillan India Ltd. (2004)
[14] Zimmermann, H.J.: Description and optimization of fuzzy systems. International Journal of General Systems 2, 209–215 (1976)

Optimal Media Selection for a Product in Segmented Market under Fuzzy Environment

Remica Aggarwal[1], Sugandha Aggarwal[1], and Sadia Samar Ali[2]

[1] Department of Operational Research, University of Delhi, Delhi, India
[2] Fortune Institute of International Business, New Delhi, Delhi, India
remica_or@rediffmail.com, sugandha_or@yahoo.com, sadiasamarali@gmail.com

Abstract. Media planning & allocation has been a major concern while advertising a product. Appropriate media selection involves choosing the media that are effective to the target segments of the potential market. Effectiveness of the media can be judged through the target audience responses to a specific media. Predicting future audience responses as well as human capability to understand & analyze past audiences is a complex problem. Hence the statistical data of audience response is bound to be uncertain or imprecise. An appropriate method to deal with imprecise judgment or uncertainties in data is fuzzy logic. So far the cited literature confines itself to an exact or certain statistical data of audience's responses. In this paper, a model has been developed which deals with optimal allocation of advertising budget for a product which is advertised through different media in a segmented market under fuzzy predictions of the audience impact. The problem is formulated as a fuzzy multi objective problem and is solved using fuzzy goal programming technique to arrive to a compromised solution. A case study is presented to show the real life application of the model.

Keywords: Advertising reach, media allocation, segmented market, fuzzy goal programming.

1 Introduction

It is not enough for a business to have good products sold at attractive prices. To generate sales and profits, the benefits of products have to be communicated to customers. In marketing, this is commonly known as "promotion". A business' total marketing communications programme is called the "promotion mix" and consists of blend of advertising, personal selling, sales promotion and public relations tools. Advertising holds an important position in the promotion mix of marketing due to its varied advantages. Media being the most common form of advertising is used to convey an advertising message to the public through newspapers, magazines, direct mail, radio, television, internet etc.

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 983–995.
springerlink.com

Choosing the right media mix involves the choice of appropriate media that can target the right audience, message to be given to the masses etc. An appropriate media needs to be chosen depending upon the kind of potential consumer market, kind of product to be marketed etc. Potential market is heterogeneous, consisting of buyers of different tastes, likes and wants. To capture maximum from the potential market, marketers or firm's marketing executives stresses on segmenting the market among homogenous groups with like needs, traits, and wants. This is called market segmentation. Depending upon the kind of product a firm needs to advertise, the market can be segmented among kids, adults, aged customers. On the same lines media selection also varies with the kind of product. For example, a product meant to be sold to adults in the potential population is of no use if it is advertise on the channel meant for kids. An energy drink like boost or junior horlicks which centers on kids or youngsters are largely advertised on channels like cartoon network or Pogo. This is because in this way they will be able to capture larger target audiences and the more audiences captured, the more effective is the media. So whenever a firm decides to advertise their product, it very carefully chooses the type of media. They have been decided based on the popularity and the number of audiences.

Another important aspect of media allocation for a firm, after having chosen the kind of media to advertise, becomes the amount of financial resources available with him to be spent on media advertising. It is generally perceived that more you spend on advertising the more profit margins you will generate. So heavy advertising generally benefits but this is not possible with firms who have limited amount of financial resources with them. Hence optimal allocation of the budget among the various media so that the advertising of the product brings the maximum return measured in terms of reach is of prime concern to them. Allocation of budget among various media also depends on the weights given to them. Every media cannot be given the same importance as all of them cannot be equally effective. These weights can be decided by the management based on their experience or through past collection.

Researches on media advertising & selection have been done over last decades. A Goal Programming model for media selection was introduced by Charnes et al. [5] which addresses the problems associated with the critical advertisement measurement of frequency and reach. Lee [17] considers a similar problem and also uses the goal programming approach. De Kluyver [6] proposed the more realistic use of hard and soft constraints for linear programming models used in media selection. Keown and Duncan [11] developed an integer goal programming model to solve media selection problems and improved upon sub-optimal results produced by linear programming and non-integer GP models. Moynihan et al. [16] & Fructhter and Kalish [8] contended that the mathematical requirements of the MCDM model for media selection force the media planner to create an artificial structuring of the media selection criteria. Lee and Kwak [13] have developed an information resource planning using an AHP based goal programming model. An approach to planning an advertising campaign of goods and services was discussed by Belenky [1,2] in conformity to single as well as multiple products which are advertised by a firm within a certain period of time. Mihiotis & Tsakiris [14]

reviewed the recent study to advertisement planning. The study discussed the best possible combination of placements of commercial (channels, time and frequency) with the goal of the highest rating subject to constrained advertisement budgets. Kwak *et al*. [12] has presented a case study that considers two options: industrial and consumer products. In order to resolve the strategic decision making about dual market high technology products, a mixed integer goal programming model is developed to facilitate the advertising media selection process. A chance constraints goal programming model for the advertising planning problem by U.K. Bhattacharya [4] presents a model which has been designed to decide the number of advertisement in different advertising media and the optimal allocation of the budget assigned to the different media. But these models ignore the practical aspect of segmentation.

Predicting future audience responses as well as human capability to understand & analyze past audiences is a complex problem. Hence the statistical data of audience's responses is bound to be uncertain. The optimal solution of the problem so obtained, is not actually representative of the complete & exact information. Implementation of such solution may result in huge losses due to vague definition of the advertising model. Fuzzy set theory [3,8,19,20,12] builds a model to represents a subjective computation of possible effect of the given values on the problem and permits the incorporation of vagueness in the conventional set theory that can be used to deal with uncertainty quantitatively.

In this paper, a model has been developed which deals with optimal allocation of advertising budget for a product (a professional education programme) which is advertised through different media (print media as well as electronic media) in a segmented market (student and professionals) under fuzzy predictions of the audience impact (like circulations figures of the audiences/ readers, advertisement cost per advertisement etc.). The data available for estimation and analysis is triangular fuzzy data. Fuzzy predictions of the statistical data have been defuzzified using Heilpern's defuzzifier [10] and a crisp multi-objective media selection model has been developed using the defuzzified values. It is desired to find the number of advertisements to be given in different media, within the allowable budget assigned for different media, in order to maximize the desirable reach to the target audiences in different segments. Weights have been attached to different media so as to maximize the total advertising reach from all the media in the allotted budget. The media selection problem is formulated as a multi-objective fuzzy mathematical programming problem involving fuzzy objective and fuzzy cost budget and advertising reach constraints. Fuzzy goal programming technique is used to provide a feasible/ compromise solution.

The rest of this paper is organized as follows. Model formulation for the multi-objective programming problem has been given in Section 2. A case for an upcoming academic institution for its one year Post Graduate Diploma in Marketing Management has been discussed in section 3. Numerical illustrations and observations on the case problem have been made in section 4. Concluding remarks & scope for future research is discussed in section 5.

2 Model Formulation

2.1 Optimal Media Selection Model in a Segmented Market

Notations

i = 1,2,…..N segments

j = 1,2,…M medium of advertising

k_j: Media option of j^{th} medium

l_j : Insertion/slot in j^{th} medium

s_c : Criteria under consideration for a given segment

$\tilde{a}_{ijk_jl_j}$: Triangular fuzzy advertising reach to the target audience for an advertisement in i^{th} segment , j^{th} media , k_j^{th} media option , l_j^{th} slot/insertion.

$a_{ijk_jl_j}$: Defuzzified advertising reach to the target audience for one advertisement in i^{th} segment , j^{th} medium , k_j^{th} media option , l_j^{th} slot/insertion

$\tilde{C}_{ijk_jl_j}$: Triangular fuzzy average number of readers/viewers in i^{th} segment, j^{th} medium, k_j^{th} media option , l_j^{th} slot/insertion

$c_{ijk_jl_j}$: Defuzzified average number of readers/viewers in i^{th} segment, j^{th} medium, k_j^{th} media option , l_j^{th} slot/insertion

$\tilde{c}_{ijk_jl_j}$: Triangular fuzzy advertisement cost of inserting one advertisement in i^{th} segment, j^{th} medium , k_j^{th} media option , l_j^{th} slot/insertion

$c_{ijk_jl_j}$: Defuzzified advertisement cost of inserting one advertisement in i^{th}-segment , j^{th} medium , k_j^{th} media option , l_j^{th} slot/insertion

$t_{ijk_jl_j}$: Minimum number of advertisements in different positions in i^{th} segment, j^{th} medium, k_j^{th} media option , l_j^{th} slot/insertion

$u_{ijk_jl_j}$: Maximum number of advertisements in different positions in i^{th} segment, j^{th} medium, k_j^{th} media option , l_j^{th} slot/insertion

$x_{ijk_jl_j}$: Decision variable corresponding to i^{th} segment, j^{th} medium, k_j^{th} media option , l_j^{th} slot/insertion

W_{s_cj} : Weight corresponding s_c^{th} criteria & j^{th} medium

$p_{is_cjk_j}$: Percentage of people in i^{th} segment ,s_c criteria, j^{th} media & k_j^{th} media option

$\tilde{C}$: Triangular fuzzy advertising budget expected from various media

C_0 : Defuzzified advertising budget available for all the media

$A_j(X)$:Defuzzified total desired advertising reach for the j^{th} media

$C(X)$: Defuzzified total desired advertising budget expected from various media

2.2.1 Problem Formulation

The company is manufacturing a product which is to be advertised in different media such as newspaper, radio, T.V, magazines etc. The problem for finding the optimal number of advertisements to be allocated to different media sources *j*, *j=1,2,….M* that would maximize the advertising reach of the product which needs

to be advertised under different media options and fuzzy cost–budget constraint can be written as the following multi-objective programming problem:

Maximize

$$Z = \begin{bmatrix} \tilde{A}_1(X) = \sum_{i=1}^{N}\sum_{k_j=1}^{K}\sum_{l_j=1}^{L} \tilde{a}_{i1k_1l_1} x_{i1k_1l_1} \\ \tilde{A}_2(X) = \sum_{i=1}^{N}\sum_{k_j=1}^{K}\sum_{l_j=1}^{L} \tilde{a}_{i2k_2l_2} x_{i2k_2l_2} \\ . \\ . \\ . \\ \tilde{A}_M(X) = \sum_{i=1}^{N}\sum_{k_j=1}^{K}\sum_{l_j=1}^{L} \tilde{a}_{iMk_Ml_M} x_{iMk_Ml_M} \end{bmatrix}$$

Subject to

$$\begin{aligned} &\tilde{C}(X) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k_j=1}^{K}\sum_{l_j=1}^{L} \tilde{c}_{ijk_jl_j} x_{ijk_jl_j} \lesssim \tilde{C} \\ &\tilde{A}_j(X) \gtrsim \tilde{A}_0^j \quad ; j = 1,2,....M \\ &X \in S = \{x_{ijk_jl_j}; X = [x_{ijk_jl_j}; x_{ijk_jl_j} \geq t_{ijk_jl_j} \quad ; x_{ijk_jl_j} \leq u_{ijk_jl_j}\} \end{aligned} \tag{P1}$$

Where $\tilde{c}_{ijk_jl_j}$ is the fuzzy cost and $\tilde{a}_{ijk_jl_j}$ is the fuzzy advertising reach given by $\tilde{a}_{ijk_jl_j} = \left(\sum_{c=1}^{q} w_{s_cj} * p_{is_cjk_jl_j}\right) * \tilde{C}_{ijk_jl_j} \;\forall\; i = 1,2,....N;\ j = 1,2,....M;\ l_j = 1,2,.....L;\ k_j = 1,2,.....K$ and X is a vector based element $X = x_{ijk_jl_j} \geq 0$ & integers $\forall i = 1,2,....N;\ j = 1,2,....M;\ l_j = 1,2,.....L;\ k_j = 1,2,.....K$.

2.2.2 The Solution

1. Compute the crisp equivalent of the fuzzy parameters using a defuzzification function. Here we use the Heilpern's defuzzification function $F_2(A) = \left(a^1 + 2a^2 + a^3\right)/4$ where a^1, a^2, a^3 are the triangular fuzzy numbers.
2. Incorporate the objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The above problem (P1) can be rewritten as

   Find X

   Subject to

$$\begin{aligned} &C(X) \lesssim C_0 \\ &A_j(X) \gtrsim A_0^j \qquad \forall\ j = 1,2,....M \\ &X \in S \end{aligned} \tag{P2}$$

$a_{ijkjlj} = EV(\tilde{a}_{ijkjlj})$ & $c_{ijkjlj} = EV(\tilde{c}_{ijkjlj})$ is the defuzzified advertisement reach and defuzzified advertisement cost. A_0^j is the defuzzified firm's aspiration level for the j[th] media objective.

3. Define appropriate membership functions for each fuzzy inequalities as well as constraint corresponding to the objective function. The membership function for the $fuzzy \leq type$ & $\geq type$ are given as

$$\mu_C(X) = \begin{cases} 1 & ; C(X) \leq C_0 \\ \dfrac{C^* - C(X)}{C^* - C_0} & ; C_0 < C(X) \leq C^* \\ 0 & ; C(X) > C^* \end{cases} \quad \mu_{A_j}(X) = \begin{cases} 1 & ; A_j(X) \geq A_0{}^j \\ \dfrac{A_j(X) - B_0{}^{j*}}{A_0{}^j - A_0{}^{j*}} & ; A_0{}^{j*} \leq A_j(X) < A_0{}^j \\ 0 & ; A_j(X) < A_0{}^{j*} \end{cases} \quad ;j=1,2\ldots M$$

respectively, where C_0 and A_0^j are the restriction and aspiration levels respectively and where C*and A_0^j* are the tolerance levels .

4. Employ extension principle to identify the fuzzy decision, which results in a crisp mathematical programming problem given by

   Maximize α

   Subject to

$$\begin{aligned} &\mu_C(x) \geq \alpha \\ &\mu_{A_j}(X) \geq \alpha \qquad \forall\ j = 1, 2, \ldots M \\ &X \in S \ ; \quad \alpha \in [0,1] \end{aligned} \tag{P3}$$

and can be solved by the standard crisp mathematical programming algorithms.

5. While solving the problem following steps 1-4, objective of the problem is also treated as a constraint. Further each objective can have different level of importance and can be assigned weight to measure the relative importance. The resulting problem can be solved by the weighted *min max* approach. The crisp formulation of the weighted problem is given as

   Maximize α

   Subject to

$$\begin{aligned} &\mu_C(x) \geq \alpha \\ &\mu_{A_j}(X) \geq w_j \alpha \qquad \forall\ j = 1, 2, \ldots M \\ &X \in S \ ; \quad \alpha \in [0,1] \quad w_j \geq 0\ ,\ j = 1, 2 \ldots M \ ; \sum_{j=1}^{M} w_j = 1 \end{aligned} \tag{P4}$$

   where *M* is the number of constraints in (P4) and α represents the degree up to which the aspiration of the decision maker is met. The problem (P4) can be solved using standard mathematical programming approach.

6. On substituting for $\mu_C(X)$ & $\mu_{A_j}(X)$ *(j=1,2…M)* the problem becomes

   Maximise α

   Subject to

$$\begin{aligned} &C(X) \leq C_0 + (1-\alpha)(C^* - C_0) \\ &A_j(X) \geq A_0^j - (1 - w_j\alpha)(A_0^j - A_0^{j*}) \\ &X \in S; \alpha \in [0,1] \quad w_j \geq 0\ ,\ j = 1, 2 \ldots M \quad , \sum_{j=1}^{M} w_j = 1 \end{aligned} \tag{P5}$$

7. If a feasible solution is not obtainable for the problem (P4) or (P5) then fuzzy goal programming approach can be used to obtain a compromised solution [16]. The method is discussed in detail in the numerical illustration.

3 Case Problem

An institution is considering a PGDM (Marketing) for Professionals(PF) and Students (ST). It wants to advertise the program so that it can have the maximum reach among the target segments. Three different types of newspapers, two types of internet websites & three different types of T.V channels have been taken as the sources of media to advertise the professional program. In a single newspaper, there could be many possibilities for example advertisement on front page, last page or in between etc. Similarly, the product can be advertised in different time slots in different channels or in different websites so as to cater to large number of audiences. Two different time slots (prime time (PT) & other time (OT)) for each of the TV channel has been decided based on the importance and number of viewers. The data available for estimation and analysis corresponding to circulation figures, advertising cost etc. is triangular fuzzy data. Every media has a budgetary goal associated with them. The main aim is to advertise in different media so as to maximize the reach to the target segments with in its allowable budgets assigned for the different media without violating the maximum and minimum number of advertisements for various media. Related criteria taken into consideration are given in Table 1 below.

Table 1. Criteria

	Professionals(PF)	Students(ST)
Criterion 1: Age	Between 30 and 50 years	Between 20 and 30 years
Criterion2: Income	2 lacs rupees & above	Fifty thousand rupees & above.

Circulation figures

It defines the average number of readers (in case of print media) or the average number of viewers (in case of T.V/internet media). Let LC_j, MC_j & UC_j represents the fuzzy lower, middle & upper circulation figures respectively for the j^{th} media. (Refer Table 3.1a-Table 3.1c of Annexure A)

Advertisement cost

Advertisement cost is cost incurred in displaying an advertisement of the given product in different media. Let LA_j, MA_j & UA_j represents the fuzzy lower, middle & upper advertising costs for the j^{th} media.(Table 3.2a -3.2c of Annexure A).

Profile Matrix

The readership profile/viewership profile presents the percentage of people in different criteria who are reading or viewing the advertisement. Viewership profile matrix based on a random sample of size 200 can be obtained for each website separately. (Refer Table 3.3a &3.3b of Annexure A)

Assignment of Weights

The assignment of weights is based on the expert's judgment for the age and income criteria. Weights assigned for age and income are 0.4 and 0.6 respectively for professionals & 0.6 and 0.4 respectively for students.

Min. & Max. number of ads in various media (*Table 3.5Annexure A)*

The defuzzified values are attached as **Annexure B.** Using the defuzzified values of the advertising reach coefficients for all media, a multi-objective programming problem combining all the objectives can be written as:

Maximize

$$A_1(X) = 240,000x_{1111} + 120,000x_{1112} + 184,150x_{1121} + 106,000x_{1122} + 204,000x_{1131} + 125,000x_{1132} + 232,500x_{2111} + 150,000x_{2112} + 192,000x_{2121} + 135,000x_{2122} + 183,550x_{2131} + 135,000x_{2132}$$

$$A_2(X) = 240,000x_{1211} + 120,000x_{1212} + 270,000x_{1221} + 100,000x_{1222} + 256,000x_{2211} + 120,000x_{2212} + 268,950x_{2221} + 109,500x_{2222}$$

$$A_3(X) = 313,500x_{1311} + 120,000x_{1312} + 297,675x_{1321} + 104,000x_{1322} + 341,775x_{1331} + 138,675x_{1332} + 188,000x_{2311} + 141,000x_{2312} + 296,050x_{2321} + 135,600x_{2322} + 246,225x_{2331} + 135,000x_{2332}$$

Subject to

$$\begin{aligned} & C(X) \lesssim \widetilde{C} \\ & A_j(X) \gtrsim \tilde{A}_0^j \quad j = 1, 2, 3 \\ & X \in S \end{aligned} \tag{P6}$$

where $C(X) = 35,000x_{1111} + 17,250x_{1112} + 35,000x_{1121} + 17,000x_{1122} + 30,000x_{1131} + 17,250x_{1132} + 30,000x_{2111} + 12,250x_{2112} + 30,000x_{2121} + 15,000x_{2122} + 34,000x_{2131} + 15,000x_{2132} + 40,000x_{2111} + 30,000x_{2112} + 45,000x_{2221} + 25,000x_{2222} + 40,000x_{2211} + 30,000x_{2212} + 45,000x_{2221} + 25,000x_{2222} + 36,250x_{1311} + 27,250x_{1312} + 37,000x_{1321} + 27,000x_{1322} + 37,750x_{1331} + 27,250x_{1332} + 37,750x_{2311} + 22,250x_{2312} + 36,500x_{2321} + 25,000x_{2322} + 34,500x_{2331} + 25,000x_{2332}$

$$S = \left\{ \begin{array}{l} x_{1111} \geq 2, x_{1112} \geq 0, x_{1121} \geq 3, x_{1122} \geq 0, x_{1131} \geq 2, x_{1132} \geq 0; x_{1112} \leq 6, , x_{1122} \leq 7, x_{1132} \leq 6; \\ x_{2111} \geq 2, x_{2112} \geq 0, x_{2121} \geq 3, x_{2122} \geq 0, x_{2131} \geq 3, x_{2132} \geq 0; x_{2112} \leq 6, x_{2122} \leq 7, x_{2132} \leq 6; \\ x_{1211} \geq 1, x_{1212} \geq 0, x_{1221} \geq 1, x_{1222} \geq 0, x_{2211} \geq 2, x_{2212} \geq 0; x_{2221} \geq 3, x_{2222} \geq 0, x_{2212} \leq 6, x_{2222} \leq 7; x_{1212} \leq 6, x_{1222} \leq 7; \\ x_{1311} \geq 3, x_{1312} \geq 0, x_{1321} \geq 3, x_{1322} \geq 0, x_{1331} \geq 3, x_{1332} \geq 0; x_{1312} \leq 6, , x_{1322} \leq 7, x_{1332} \leq 6; \\ x_{2311} \geq 3, x_{2312} \geq 0, x_{2321} \geq 3, x_{2322} \geq 0, x_{2331} \geq 3, x_{2332} \geq 0; x_{2312} \leq 6, x_{2322} \leq 7, x_{2332} \leq 6; \end{array} \right\}$$

The cost budget constraints as well as the aspiration level constraints are imprecise. If the available budget is specified as TFN given by $\widetilde{C} = (18,00,000; 20,00,000; 22,00,000)$ & the aspiration level for the newspaper objective is a TFN given by $\tilde{A}_0^1 = (55,00,000; 60,00,000; 65,00,000)$; $\tilde{A}_0^2 = (75,00,000; 80,00,000; 85,00,000)$; $\tilde{A}_0^3 = (75,00,000; 80,00,000; 85,00,000)$;

Then again using the defuzzification function we get the defuzzified values as $C_0 = 20,00,000; A_0^1 = 55,00,000; A_0^2 = 80,00,000; A_0^3 = 80,00,000.$

Find X

Subject to

$$\begin{aligned} & C(X) \lesssim 20,00,000; A_1(X) \gtrsim 55,00,000; A_2(X) \gtrsim 80,00,000; A_3(X) \gtrsim 80,00,000 \\ & X \in S \end{aligned} \tag{P7}$$

Where

$$\mu_C(x) = \left\{ \begin{array}{ll} 1 & ; C(x) < 2000000 \\ \dfrac{2500000 - C(X)}{2500000 - 2000000} & ; 2000000 \leq C(x) \leq 2500000 \\ 0 & ; C(x) > 2500000 \end{array} \right\} \mu_{A_1}(x) = \left\{ \begin{array}{ll} 1 & ; A_1(x) > 6000000 \\ \dfrac{A_1(X) - 5500000}{6000000 - 5500000} & ; 5500000 \leq A_1(x) < 6000000 \\ 0 & ; A_1(x) < 5500000 \end{array} \right\}$$

$$\mu_{A_2}(x)=\begin{cases}1 & ;A_2(x)>8000000\\ \dfrac{A_2(X)-7500000}{8000000-7500000} & ;7500000\le A_2(x)<8000000\\ 0 & ;A_2(x)<7500000\end{cases}\quad \mu_{A_3}(x)=\begin{cases}1 & ;A_3(x)>8000000\\ \dfrac{A_3(X)-7500000}{8000000-7500000} & ;7500000\le A_3(x)<8000000\\ 0 & ;A_3(x)<7500000\end{cases}$$

Now the equivalent crisp optimization problem based on extension principle corresponding to problem (P7) considering different weights for the cost and advertising reach objectives, is given as

Maximize α

Subject to

$$\begin{aligned}&\mu_C(X)\ge\alpha;\ \ \mu_{A_1}(X)\ge w_1\alpha;\mu_{A_2}(X)\ge w_2\alpha;\mu_{A_3}(X)\ge w_3\alpha;\\ &X\in S;\quad \alpha\in[0,1];\quad w_1,w_2,w_3\ge 0;\quad w_1+w_2+w_3=1\end{aligned}\tag{P8}$$

The problem is a crisp linear programming problem and can be solved using standard mathematical programming methods.

Fuzzy Goal Programming Approach

On solving the problem, it is found that the problem (P8) is not feasible, hence the management goal can't be achieved for a feasible value of $\alpha\in[0,1]$. Now we use fuzzy goal programming technique to obtain a compromised solution. The approach is based on the goal programming technique for solving crisp goal programming problem [16]. The fuzzy goal programming formulation for the given problem (P8) introducing the negative and positive deviational variables η_j and ρ_j is given as

Minimize u

Subject to

$$\begin{aligned}&\mu_C(X)+\eta-\rho=1\\ &\left.\begin{aligned}&\mu_{A_j}(X)+\eta_j-\rho_j=1\ ;\qquad u\ge w_j*\eta_j\\ &\eta_j*\rho_j=0;\quad \eta_j,\rho_j\ge 0\end{aligned}\right\}\ j=1,2,3\\ &X\in S\ ;\ w_1,w_2,w_3\ge 0;\quad w_1+w_2+w_3=1\ ;\ u=1-\alpha\end{aligned}\tag{P9}$$

4 Numerical Illustration

Programming problem (P1)-(P2), (P6)-(P9) can easily be solved using standard mathematical programming methods. Due weights which are given to both budget as well as to different media so as to capture maximum reach for both the media in the allocated budget. For example, considering the weights as 0.8-0.1-0.1for newspaper, website & TV media respectively, *U* is obtained as 0.5480411 giving budget as Rs 22,61,750 (media budget range was Rs 20,00,000-25,00,000). Considering another set of weights as 0.6-0.15-0.25 corresponding to newspaper, website & TV media, *U* is obtained as 0.7619781 giving budget as Rs24,12,750. Soon assigning different weights, alternative solutions can be found. W_{NP} W_{WS} & $W_{T.V}$ denotes the weights given to the newspaper, website & T.V media respectively.

Table 2. Weights & reach achieved for different media

Media	W_{NP}=0.8,W_{WS} =0.1, $W_{T.V}$ =0.1	W_{NP}=0.2,W_{WS} =0.2, $W_{T.V}$ =0.6
	Reach achieved	Reach achieved
NP's	56,94,100	52,27,000
WS's	52,66,350	60,14,850
T.V	50,49,675	56,45,025
Total	1,60,10,125	1,68,86,875

Table 3. Number of ads allocated to different media in different slots

X_{1111}	2	X_{1132}	0	X_{2131}	3	X_{1222}	0	X_{1311}	3	X_{1332}	0	X_{2331}	3
X_{1112}	0	X_{2111}	2	X_{2132}	0	X_{2211}	16	X_{1312}	0	X_{2311}	3	X_{2332}	0
X_{1121}	3	X_{2112}	3	X_{1211}	4	X_{2212}	0	X_{1321}	3	X_{2312}	0		
X_{1122}	6	X_{2121}	3	X_{1212}	0	X_{2221}	3	X_{1322}	0	X_{2312}	0		
X_{1131}	2	X_{2122}	0	X_{1221}	1	X_{2222}	0	X_{1331}	8	X_{2322}	0		

5 Conclusions

The media selection problem is formulated as a multi-objective programming problem and fuzzy goal programming technique is used to provide a feasible solution. The model in this paper is formulated for single product advertising which may be generalized for multiple products.

References

[1] Belenky, A.: An approach to planning an advertising campaign of goods and services. Computers Mathematics Application 42(6/7), 993–1008 (2001)

[2] Belenky, A., Belenkii, I.: Optimization of planning an advertising campaign of goods and services. Mathematics & Computer Modelling 35(316), 1391–1403 (2002)

[3] Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. Management Science 17, 141–164 (1970)

[4] Bhattacharya, U.K.: A chance constraints goal programming model for the advertising planning problem. European Journal of Operational Research 192, 382–395 (2009)

[5] Charnes, A., Cooper, W.W., DeVoe, J.K., Learner, D.B., Reinecke, W.: A goal programming model for media planning. Management Science 14, 422–430 (1968)

[6] De Kluyver, C.A.: Hard and soft constraints in media scheduling. J. of Advertisement Res. 18, 27–31 (1978)

[7] Fructhter, G.E., Kalish, S.: Dynamic promotional budgeting and media allocation. European Journal of Operational Research 111(1), 15–27 (1998)

[8] Gladish, B., Gonzalez, I., Terol, A., Parra, M.: Planning a TV advertising campaign: A crisp multi objective programming model from fuzzy basic data. Omega 38, 84–94 (2010)

[9] Haimes, Y.Y., Steuer, R.E.: Research and practice in multiple criteria decision making. Lecture Notes in Economics and Mathematical Systems, vol. 487, p. 552. Springer, Berlin (2000)
[10] Heilpern, S.: The expected value of a fuzzy number. Fuzzy Sets and Systems 47, 81–86 (1992)
[11] Kwak, N.K., Lee, C.W., Kim, J.H.: An MCDM model for media selection in the dual consumer/industrial market. European Journal of Operational Research 166, 255–265 (2005)
[12] Lee, C.W., Kwak, N.K.: Information resource planning using AHP-based goal programming model. J. Opl. Res. Soc. 51, 1–8 (1999)
[13] Mihiotis, A., Tsakiris, I.: A mathematical programming study of advertising allocation problem. Applied Mathematics & Computation 148(2), 373–379 (2004)
[14] Mohamed, R.H.: The relationship between goal programming and fuzzy programming. Fuzzy Sets & Systems 89, 215–222 (1997)
[15] Moynihan, G.P., Kumar, A., D'Souza, G., Nockols, W.G.: A decision support system for media planning. Computer and Industrial Engg. 29(1), 383–386 (1995)
[16] Lee, S.M.: Goal Programming for Decision Analysis. Auerbach Publishers, Philadelphia (1972)
[17] Zimmermann, H.J.: Fuzzy set theory and its applications. Kluwer Academic Publishers, Boston (1985)
[18] Zimmermann, H.J.: Fuzzy set theory &its applications. Academic Publisher (1991)

Appendix –A

Table 3.1a. Fuzzy Circulation figures of newspaper for professionals& students

		NP 1 ('000)			NP2 ('000)			NP3 ('000)		
		LC_n	MC_n	UC_n	LC_n	MC_n	UC_n	LC_n	MC_n	UCn
FP	PF	350	400	450	250	300	350	300	340	380
OP		250	300	350	240	270	300	200	250	300
FP	ST	350	400	450	250	300	350	300	340	380
OP		250	300	350	240	270	300	200	250	300

Table 3.1b. Fuzzy Circulation figures of websites for professionals &students

		WS 1('000)			WS 2('000)		
		LC_w	MC_w	UC_w	LC_w	MC_w	UC_w
PT	PF	350	400	450	400	450	500
OT		250	300	350	200	250	300
PT	ST	350	400	450	400	450	500
OT		250	300	350	200	250	300

Table 3.1c. Fuzzy Circulation figures of channels for professionals& students

		CH 1 ('000)			CH 2 ('000)			CH 3 ('000)		
		LC_T	MC_T	UC_T	LC_T	MC_T	UC_T	LC_T	MC_T	UC_T
PT	PF	500	520	550	450	470	500	500	550	570
OT		250	300	350	240	250	300	300	320	350
PT	ST	450	470	490	450	480	500	350	370	380
OT		250	300	350	240	270	350	200	250	300

Table 3.2a. Fuzzy advertising cost of newspaper for professionals& students

		NP 1			NP 2			NP 3		
		LA_n	MA_n	UA_n	LA_n	MA_n	UA_n	LA_n	MA_n	UAn
FP	PF	300	350	400	300	350	400	250	300	350
OP		150	170	200	140	170	200	150	170	200
FP	ST	250	300	350	250	300	350	300	340	380
OP		100	120	150	140	150	160	100	150	200

Table 3.2b. Fuzzy advertising cost of website for professionals & students

		WS 1('00)			WS 2('00)		
		LA_w	MA_w	UA_w	LA_w	MA_w	UA_w
PT	PF	350	400	450	400	450	500
OT		250	300	350	200	250	300
PT	ST	350	400	450	400	450	500
OT		250	300	350	200	250	300

Table 3.2c. Fuzzy advertising cost of channels for professionals& students

		CHL1 ('00)			CHL 2 ('00)			CHL 3 ('00)		
		LA_T	MA_T	UA_T	LA_T	MA_T	UA_T	LA_T	MA_T	UA_T
PT	PF	310	370	400	320	380	400	350	380	400
OT		250	270	300	240	270	300	250	270	300
PT	ST	350	380	400	300	380	400	300	350	380
OT		200	220	250	240	250	260	200	250	300

Table 3.3a. Readership profile matrix for professionals & students

		NP 1		NP 2		NP 3	
		FP	OP	FP	OP	FP	OP
Age	PF	.6	.4	.6	.4	.6	.4
Income		.6	.4	.65	.4	.6	.5
Age	ST	.6	.45	.6	.5	.65	.5
Income		.6	.5	.7	.5	.7	.6

Table 3.3b. Viewership profile matrix for professionals & students

		WS 1		WS 2		CH1		CH2		CH3	
		PT	OT	PT	OT	PT	OT	PT	OT	PT	OT
Age	PF	.6	.4	.66	.46	.6	.4	.6	.4	.6	.4
Income		.6	.4	.6	.43	.6	.45	.65	.4	.65	.45
Age	ST	.65	.45	.68	.46	.6	.45	.6	.5	.65	.5
Income		.62	.4	.6	.40	.6	.5	.65	.45	.67	.6

Table 3.4. Min. & max no. of ads in different media for professionals & students

		NP1	NP2	NP3	CH 1	CH 2	CH 3	WS1	WS2
PT	PF	[2,N]	[3,N]	[2,N]	[3,N]	[3,N]	[3,N]	[1,N]	[1,N]
OT		[0,6]	[0,7]	[0,6]	[0,6]	[0,7]	[0,6]	[0,6]	[0,7]
PT	ST	[2,N]	[3,N]	[3,N]	[3,N]	[3,N]	[3,N]	[2,N]	[3,N]
OT		[0,6]	[0,7]	[0,6]	[0,6]	[0,7]	[0,6]	[0,6]	[0,7]

Appendix B: Defuzzified values of advertising reach for different media

Newspaper media			T.V media			Website media	
	PF	ST		PF	ST		PF
NP1(FP)	240,000	232,500	CH1(PT)	313,500	188,000	WS1(PT)	240,000
NP1OP)	120,000	150,000	CH1(OT)	120,000	141,000	WS1 (OT)	120,000
NP2(FP)	184,150	192,000	CH2(PT)	297,675	296,050	WS2 (PT)	270,000
NP2(OP	106,000	135,000	CH2(OT)	104,000	135,600	WS2 (OT)	100,000
NP3(FP)	204,000	183,550	CH3 (PT)	341,775	246,225		ST
NP3(OP	125,000	135,000	CH3(OT)	138,675	135,000	WS1(PT)	256,000
						WS1 (OT)	120,000
						WS2 (PT)	268,950
						WS2 (OT)	109,500

Multicriteria Credibilistic Portfolio Rebalancing Problem with Fuzzy Chance-Constraint

Pankaj Gupta, Garima Mittal, and Mukesh K. Mehlawat

Department of Operational Research, University of Delhi, Delhi, India
pgupta@or.du.ac.in, {garima28, mukesh0980}@yahoo.com

Abstract. In this paper, we propose multicriteria credibilistic framework for portfolio rebalancing (adjusting) problem with fuzzy parameters considering return, risk and liquidity as key financial criteria. Transaction cost is an important factor to be taken into consideration in portfolio selection. It is not trivial enough to be neglected and the optimal portfolio depends upon the total costs of transaction. We assume that the investor pays changeable transaction costs based on incremental discount schemes, which are adjusted in the net return of the portfolio. A hybrid intelligent algorithm, integrating fuzzy simulation and real-coded genetic algorithm is developed to solve the portfolio rebalancing (adjusting) model.

Keywords: Fuzzy portfolio selection, Credibility measure, Multi-objective programming, Real-coded genetic algorithm, Transaction costs.

1 Introduction

Portfolio selection as a field of study began with the mean-variance model [17] in which return is quantified as the mean and risk as the variance. In most of the existing portfolio selection models, return and risk are considered as the two fundamental factors that govern investors' choice. However, it is often found that not all the relevant information for portfolio selection can be captured in terms of return and risk only. The other considerations/criteria might be of equal, if not greater, importance to investors. By considering these in the portfolio selection model, it may be possible to obtain portfolios in which a deficit on account of the return and risk criteria is more than compensated by portfolio performance on other criteria, resulting in greater overall satisfaction for investors. Thus, the multicriteria portfolio selection models have received great interest from researchers in the recent past.

Traditionally, the portfolio selection models are based on the assumption that investors have complete information for decision making. However, in reality, the information is often incomplete and hence decisions are made under uncertainty. Moreover, the financial markets are affected by vagueness and ambiguity associated with the use of linguistic expressions such as "high risk", "low profit" and "low liquidity" by the investors and the investment experts. The portfolio selection

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 997–1010.
springerlink.com

formulations have benefited greatly from the fuzzy set theory [23, 24] in terms of integrating quantitative and qualitative information, subjective preferences of the investors and knowledge of the experts. Assuming that the returns are fuzzy, a large literature is available on the fuzzy mean-variance models, see [8, 13, 21, 25, 26]. Most of the work on fuzzy portfolio selection has been developed using possibility theory which has some limitations. One main limitation is that possibility measure is not self-dual. However, the self-dual property is very important both in theory and in practice. A fuzzy event with maximum possibility value 1 can still fail to happen. Additionally, it is possible that two fuzzy events with different occurring chances may have the same possibility value, which provides no information to the decision maker. As an alternative, Liu and Liu [15] defined credibility measure which is self-dual and overcomes the difficulties inherent with the possibility measure. The study of portfolio selection models with credibility measure [10], however, is limited to mainly risk-return tradeoff.

Given an initial holding of portfolio, at any given period of time, the investor may consider rebalancing (adjusting) the existing portfolio by buying or selling assets in response to changing conditions in financial market. In practical situations, there is transaction cost associated with buying and selling of an asset. Transaction cost is one of the main concerns for portfolio managers and helps in constructing more realistic models, which incorporate market frictions, see [3, 11, 22]. Some studies on portfolio optimization consider transaction cost as fixed, on the other hand, there are studies considering variable transaction cost that changes as a proportion of the amount of assets traded, see [4, 18, 26]. Usually the incurred transaction costs depends upon the volume of the transactions in a nonlinear way. This is because the transaction cost rate is relatively large when the volume of the transaction is small and it gradually decreases as the volume of the transaction increases.

In this paper, we propose a new multicriteria credibilistic portfolio rebalancing model with nonlinear transaction costs based on credibility measure of fuzzy event. We use return, liquidity and risk as the criteria for asset selection. For portfolio return, we consider average performance of the asset during the 36-month period. Liquidity is considered in terms of the probability of conversion of an investment into cash (turnover) without any significant loss in value. The portfolio risk is defined using fuzzy parameters based downside risk from expected portfolio return and is expressed as a fuzzy chance constraint. The returns and the liquidity are considered as fuzzy parameters. Further, we assume that the investor pays transaction costs based on incremental discount schemes which are then adjusted in order to get the net return of the portfolio. The proposed model is solved using a hybrid intelligent algorithm developed by integrating fuzzy simulation and real-coded genetic algorithm techniques.

The paper is organized as follows: In Section 2, we present some basic definitions and notations. In Section 3, we present the mathematical model of multicriteria credibilistic portfolio rebalancing problem and details of the hybrid intelligent algorithm to solve it. The proposed model is test-run in Section 4 using 36-month data series in respect of randomly selected 20 assets listed on the National Stock Exchange,

Mumbai, India. This section also includes a discussion of the results obtained. We conclude the paper in Section 5.

2 Preliminaries

Credibility theory [12, 16, 15] is a branch of mathematics for studying the behavior of fuzzy events using credibility measures and is based on five axioms from which a credibility measure is defined. Let Θ be a nonempty set and let $P(\Theta)$ be the power set of Θ (i.e. all subsets of Θ). Each element in $P(\Theta)$ is called an event. Let $Cr\{A\}$ indicates the credibility that event A will occur. In order to ensure that the number $Cr\{A\}$ has certain mathematical properties, the following five axioms should hold:

1. $Cr\{\Theta\} = 1$.
2. Cr is increasing, i.e. $Cr\{A\} \leq Cr\{B\}$ whenever $A \subset B$.
3. Cr is self-dual, i.e. $Cr\{A\} + Cr\{A^c\} = 1$ for any $A \in P(\Theta)$.
4. $Cr\{\cup_i A_i\} \wedge 0.5 = \sup_i Cr\{A_i\}$ for any $\{A_i\}$ with $Cr\{A_i\} \leq 0.5$.
5. Let Θ_k be nonempty sets on which Cr_k satisfy the first four axioms, $k = 1,2,\ldots,n$, respectively, and let $\Theta = \Theta_1 \times \Theta_2 \times \ldots \times \Theta_n$. Then $Cr\{(\theta_1,\theta_2,\ldots,\theta_n)\} = Cr_1\{\theta_1\} \wedge Cr_2\{\theta_2\} \wedge \ldots \wedge Cr_n\{\theta_n\}$ for each $(\theta_1,\theta_2,\ldots,\theta_n) \in \Theta$.

It may be noted that $Cr\{\emptyset\} = 0$ and that the credibility measure takes values between 0 and 1. Traditionally, a fuzzy variable is defined by a membership function [25]. Liu [16] defined it as a function on a credibility space like a random variable is defined as a measurable function on a probability space.

Definition 1 [16]. A fuzzy variable is a function from a credibility space $(\Theta, P(\Theta), Cr)$ to a set of real numbers.

An n-dimensional fuzzy vector is defined as a function from a credibility space to the set of n-dimensional real vectors. Moreover, $(\xi_1,\xi_2,\ldots,\xi_n)$ is a fuzzy vector if and only if $\xi_1,\xi_2,\ldots,\xi_n$ are fuzzy variables.

Definition 2 [16]. Let $f : \Re^n \to \Re$ be a function and let $\xi_1,\xi_2,\ldots,\xi_n$ be fuzzy variables on the credibility space $(\Theta, P(\Theta), Cr)$. Then $\xi = f(\xi_1,\xi_2,\ldots,\xi_n)$ is a fuzzy variable defined as $\xi(\theta) = f(\xi_1(\theta),\xi_2(\theta),\ldots,\xi_n(\theta))$ for any $\theta \in \Theta$.

If the fuzzy variables are defined on different credibility spaces, then $\xi = f(\xi_1,\xi_2,\ldots,\xi_n)$ is a fuzzy variable defined on the product credibility space $(\Theta, P(\Theta), Cr)$ as $\xi(\theta_1,\theta_2,\ldots,\theta_n) = f(\xi_1(\theta_1),\xi_2(\theta_2),\ldots,\xi_n(\theta_n))$ for any $(\theta_1,\theta_2,\ldots,\theta_n) \in \Theta$.

In practice, a fuzzy variable may be specified by a membership function. In such cases we need a formula to calculate the credibility value of fuzzy event.

Definition 3 [16]. Let ξ be a fuzzy variable with membership function μ, and u, r be real numbers. The credibility of a fuzzy event characterized by $\xi \geq r$, is defined as

$$Cr\{\xi \geq r\} = \frac{1}{2}(\sup_{u \geq r} \mu(u) + 1 - \sup_{u < r} \mu(u)).$$

As is known in the literature, the credibility can be seen as an average of possibility and necessity. Namely, we have

$$Cr\{\xi \geq r\} = \frac{1}{2}(Pos\{\xi \geq r\} + Nes\{\xi \geq r\}),$$

where $Pos\{\xi \geq r\} = (\sup_{u \geq r} \mu(u))$ and $Nes\{\xi \geq r\} = 1 - \sup_{u<r} \mu(u)$.

Definition 4 [16]**.** Let ξ_i be fuzzy variables with membership functions μ_i, and let u_i be real numbers, $i = 1,2,\ldots,n$. Define $f : \Re^n \to \Re$. The credibility of the fuzzy event characterized by $f(\xi_1,\xi_2,\ldots,\xi_n) \geq 0$ is given as

$$\begin{aligned} Cr\{f(\xi_1,\xi_2,\ldots,\xi_n) \geq 0\} = \frac{1}{2}(& \sup_{u_1,u_2,\ldots,u_n \in R} \{\min_{1\leq i\leq n} \mu_{\xi_i}(u_i) | f(u_1,u_2,\ldots,u_n) \geq 0\} \\ & + 1 - \sup_{u_1,u_2,\ldots,u_n \in R} \{\min_{1\leq i\leq n} \mu_{\xi_i}(u_i) | f(u_1,u_2,\ldots,u_n) < 0\}). \end{aligned}$$

3 Research Methodology

3.1 Portfolio Rebalancing Problem

In this section, we formulate multiobjective credibilistic portfolio rebalancing problem using piecewise linear transaction costs. We assume that investors allocate their wealth among n assets offering fuzzy returns. The parameters and variables used to formulate the mathematical model are described as follows:

λ_i: the fuzzy return of the i-th asset ,
β_i: the fuzzy turnover rate of the i-th asset ,
r_1: the lower limit of the expected return of the portfolio ,
r_2: the lower limit of the expected liquidity of the portfolio ,
r_3: the severity indicator of the possible loss ,
γ: the upper tolerance for credibility of the portfolio risk ,
b: the target return of the portfolio ,
u_i: the maximal fraction of the capital budget being allocated to the i-th asset ,
l_i: the minimal fraction of the capital budget being allocated to the i-th asset ,
x_i^0: the initial proportion of total fund invested in the i-th asset before rebalancing ,
x_i^+: the proportion purchased while rebalancing the i-th asset ,
x_i^-: the proportion sold while rebalancing the i-th asset ,
x_i: the final proportion of total fund invested in the i-th asset after rebalancing ,
y_i: the binary variable indicating whether the i-th asset is contained in the portfolio or not,

$$y_i = \begin{cases} 1, & if\ i-th\ asset\ is\ contained\ in\ the\ portfolio, \\ 0, & otherwise, \end{cases}$$

h: the number of assets held in the portfolio ,
p_j: the proportional buying transaction cost rate in the interval $[P_{j-1}, P_j]$,
q_k: the proportional selling transaction cost rate in the interval $[Q_{k-1}, Q_k]$,
$B_i(x_i^+)$: the discount function representing buying transaction cost for i-th asset ,

$S_i(x_i^-)$: the discount function representing selling transaction cost for i-th asset ,
$C_i(x_i)$: the transaction cost of i-th asset ,
$C(x)$: the total transaction cost of rebalancing of portfolio .

3.1.1 Objectives

• Net return
We assume that the investor is offered incremental discounts [20], refer to Table 1, on the transaction costs incurred as a result of buying and selling of assets.

Table 1. Incremental discount schemes for buying and selling of assets

Amount bought: x_i^+	Transaction cost: $B_i(x_i^+)$
$0 = P_0 \le x_i^+ \le P_1$	$p_1 x_i^+$
$P_1 \le x_i^+ \le P_2$	$p_1 P_1 + p_2(x_i^+ - P_1)$
$P_2 \le x_i^+ \le P_3$	$p_1 P_1 + p_2(P_2 - P_1) + p_3(x_i^+ - P_2)$
$\vdots$	$\vdots$
$P_{j-1} \le x_i^+ \le P_j = u_i$	$p_1 P_1 + p_2(P_2 - P_1) + \ldots + p_{j-1}(P_{j-1} - P_{j-2}) + p_j(x_i^+ - P_{j-1})$
Amount sold: x_i^-	**Transaction cost: $S_i(x_i^-)$**
$0 = Q_0 \le x_i^- \le Q_1$	$q_1 x_i^-$
$Q_1 \le x_i^- \le Q_2$	$q_1 Q_1 + q_2(x_i^- - Q_1)$
$Q_2 \le x_i^- \le Q_3$	$q_1 Q_1 + q_2(Q_2 - Q_1) + q_3(x_i^- - Q_2)$
$\vdots$	$\vdots$
$Q_{k-1} \le x_i^- \le Q_k = u_i$	$q_1 Q_1 + q_2(Q_2 - Q_1) + \ldots + q_{k-1}(Q_{k-1} - Q_{k-2}) + q_k(x_i^- - Q_{k-1})$

The transaction cost incurred on rebalancing using i-th asset then becomes

$$C_i(x_i) = B_i(x_i^+) + S_i(x_i^-), \quad i = 1, 2, \ldots, n.$$

Hence, the total transaction cost of rebalancing of the portfolio is obtained as

$$C(x) = \sum_{i=1}^{n} C_i(x_i).$$

The credibility maximization form of the net return of the portfolio not less than r_1 after adjusting transaction cost is expressed as:

$$Max\, Cr\{\lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_n x_n - C(x) \ge r_1\}.$$

• Liquidity
For any asset, liquidity may be measured with the help of the turnover rate defined as the ratio between the average stock traded at the market and the tradable stock (shares held by public) of that asset. Because of incomplete information, the turnover rates are, at best, vague estimates. In order to handle uncertainty, liquidity has been considered in fuzzy form, see [4, 5, 6, 7]. The credibility maximization form of the portfolio liquidity not less than r_2 is expressed as:

$$Max\, Cr\{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \ge r_2\}.$$

3.1.2 Constraints

• Risk

Risk is an important issue while modeling portfolio selection problem. In this paper, portfolio risk is expressed as a fuzzy chance-constraint using downside risk measure. In chance-constrained programming [1], it is required that the objectives must be achieved with the stochastic constraints having a finite probability of violation. Following the idea of stochastic chance-constrained programming, Liu [14] developed fuzzy chance-constrained programming. We define portfolio risk as the deviation of net return of the generated portfolio from the target return b for a given severity loss indicator r_3. Using a predefined tolerance level γ for occuring chance of such a risk, the fuzzy chance constraint representing portfolio risk is expressed as:

$$Cr\{b-(\lambda_1 x_1+\lambda_2 x_2+\ldots+\lambda_n x_n-C(x))\geq r_3\}\leq \gamma.$$

• Rebalancing constraints:

$$x_i=x_i^0+x_i^+-x_i^-,\quad i=1,2,\ldots,n.$$

• Complementarity constraints on buying and selling together:

$$x_i^+.x_i^-=0,\quad i=1,2,\ldots,n.$$

• Capital budget constraint: Assuming that the investor does not invest any additional capital during the portfolio rebalancing process, the following equation is used:

$$\textstyle\sum_{i=1}^n x_i+C(x)=1.$$

• Maximal fraction of the capital that can be invested in a single asset:

$$x_i\leq u_i y_i,\quad i=1,2,\ldots,n.$$

• Minimal fraction of the capital that can be invested in a single asset:

$$x_i\geq l_i y_i,\quad i=1,2,\ldots,n.$$

• Number of assets held in the portfolio:

$$\textstyle\sum_{i=1}^n y_i=h.$$

Here, h is the number of assets that the investor chooses to include in the portfolio.

• No short selling of assets:

$$x_i^+\geq 0,\quad x_i^-\geq 0,\quad x_i\geq 0,\quad i=1,2,\ldots,n.$$

3.1.3 The Decision Problem

The bi-objective credibilistic portfolio rebalancing model with fuzzy chance constraint is formulated as:

$$\begin{aligned}
(P1)\quad & Max\, f_1(x)=Cr\{\lambda_1 x_1+\lambda_2 x_2+\ldots+\lambda_n x_n-C(x)\geq r_1\}\\
& Max\, f_2(x)=Cr\{\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n\geq r_2\}\\
& subject\ to\\
& Cr\{(b-(\lambda_1 x_1+\lambda_2 x_2+\ldots+\lambda_n x_n-C(x))\geq r_3\}\leq \gamma, && (1)\\
& x_i=x_i^0+x_i^+-x_i^-\quad i=1,2,\ldots,n, && (2)\\
& x_i^+.x_i^-=0\quad i=1,2,\ldots,n, && (3)
\end{aligned}$$

$$\sum_{i=1}^{n} x_i + C(x) = 1, \tag{4}$$

$$\sum_{i=1}^{n} y_i = h, \tag{5}$$

$$x_i \le u_i y_i, i = 1,2,\ldots,n, \tag{6}$$

$$x_i \ge l_i y_i, i = 1,2,\ldots,n, \tag{7}$$

$$x_i^+ \ge 0, \quad x_i^- \ge 0, \quad x_i \ge 0, \quad i = 1,2,\ldots,n, \tag{8}$$

$$y_i \in \{0,1\}, i = 1,2,\ldots,n. \tag{9}$$

3.2 *Hybrid Intelligent Algorithm (HIA) to Solve Portfolio Rebalancing Model*

Here, we design a HIA by integrating fuzzy simulation and real-coded genetic algorithm to solve the proposed credibility based multicriteria portfolio rebalancing model in general cases, i.e. when membership functions of fuzzy parameters may take any functional form.

3.2.1 Fuzzy Simulation

The technique of fuzzy simulation is discussed in detail by Liu [14]. In the proposed HIA, the fuzzy simulation is applied to compute the credibility values. Let $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ and $x = (x_1, x_2, \ldots, x_n)$. Moreover, let $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ denote the membership function vector of ξ. In order to solve the decision model $P1$, we must handle the following type of uncertain function:

$$U : x \to Cr\{G(x,\xi) \ge r\},$$

where $r > 0$ is given by the investor. It may be noted that for the expression of the type $Cr\{(b - (\xi_1 x_1 + \xi_2 x_2 + \ldots + \xi_n x_n - C(x))) \ge r\}$,

$$G(x,\xi) = b - (\xi_1 x_1 + \xi_2 x_2 + \ldots + \xi_n x_n - C(x)).$$

The fuzzy simulation process for computing $Cr\{G(x,\xi) \ge r\}$ is summarized as:

Step 1. Let $j = 1$.

Step 2. Randomly generate u_{ij} from the $\varepsilon-$level sets of fuzzy variables $\xi_i, i = 1,2,\ldots,n$, respectively, where ε is a sufficiently small positive number.

Step 3. Set $u_j = (u_{1j}, u_{2j}, \ldots, u_{nj})$, and $\mu(u_j) = \mu_{1j}(u_{1j}) \wedge \mu_{2j}(u_{2j}) \wedge \ldots \wedge \mu_{nj}(u_{nj})$.

Step 4. $j \leftarrow j + 1$. Go to Step 2 if $j \le N$, where N is a sufficiently large number, otherwise, go to Step 5.

Step 5. Return L, where L gives the credibility $Cr\{G(x,\xi) \ge r\}$ as follows:

$$L = \tfrac{1}{2}(\max_{1\le j\le N}\{\mu(u_j) | G(x,\xi) \ge r\} + 1 - \max_{1\le j\le N}\{\mu(u_j) | G(x,\xi) < r\}).$$

3.2.2 Real-Coded Genetic Algorithm (RCGA)

Genetic algorithm was proposed by Holland [9], and has been well developed and documented in the literature since then. Here, we use RCGA for solving the decision model $P1$. The following are the steps of the algorithm used.

• Chromosome encoding

In the proposed encoding method, the length of the chromosome is taken to be n, same as the number of available assets. Let the solution $x = (x_1, x_2, \ldots, x_n)$ be represented by the chromosome $Ch[k]$, which is encoded as an array, as follows:

$$Ch[k] = X[k][n] = [x_1\ x_2 \ldots\ x_n], k = 1, 2, \ldots, popsize.$$

Here, *popsize* defines the number of chromosomes initialized to constitute population of one generation. In this encoding method, the position of the gene $x_i, i = 1, 2, \ldots, n$, is used to represent the ID number of the asset and its value is used to represent the corresponding proportion for constructing the portfolio. The initialization algorithm to create first generation of size *popsize* is as follows:

Step 1. For $k = 1$ to *popsize*, repeat Step 2 and Step 3.
Step 2. Randomly select h assets out of n available assets for initialization.
Step 3. For $i = 1$ to n, repeat Step 4.
Step 4. If i-th asset has been selected in Step 2, then assign $y_i = 1$ and randomly generate $x_i \in [l_i y_i, u_i y_i]$, else assign $y_i = 0$ and $x_i = 0$. Thus, initializing $X[k][i] = x_i$.

• Fitness evaluation

The fitness function must take care of all the desired objectives, thus, making a rational trade-off between them. Moreover, any violation of the constraints of model $P1$ would lead to infeasible chromosomes. The resultant fitness function $fit[k]$ corresponding to chromosome $Ch[k]$, $k = 1, 2, \ldots, popsize$ is designed as weighted sum of the objectives with penalty treatment for infeasible chromosomes as follows:

$$fit[k] = w_1 f_1(x) + w_2 f_2(x) - P$$

where w_j, $j = 1, 2$ is the weight of the j-th objective function, highlighting the relative importance of the objective and P is the penalty. Random weighted genetic algorithm suggested by Murata and Ischibuchi [19] have been known to perform well while solving multiobjective optimization problems. In this paper, a random normalized weight vector $w^k = (w_1, w_2)_k^T$ is assigned to the objectives in k-th solution of the population such that $\sum_{j=1}^{2} w_j = 1$. The objective is to find the solution chromosome $Ch[k]$ corresponding to the best found (maximum value) fitness, for the function $fit[k]$.

• Selection

Selection (reproduction) operator is intended to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. We employ 4-player tournament selection as a selection mechanism in this study. Four individuals are randomly selected and the one with the highest fitness is selected for the mating pool. At each generation, elitism of

order four is performed by retaining the four most fit individuals for the population comprising the next generation.

• **Crossover operator**
Crossover is the main genetic operator. It operates on two parents (chromosomes) at a time with some probability of crossover $p_c \in (0,1)$ and generates offsprings by exchanging genes between the two selected parents. Standard crossover operators have a high probability of violating constraint (5) of the model $P1$. Thus, we use shrinking crossover (SX) operator [2]. SX revises the two-point crossover by moving the second crossover point leftward until there are equal number of selected assets between the two crossover points in both the parents and then exchanging the gene values of parent chromosomes to produce offsprings. The algorithm of the shrinking crossover operator is given below:

Step 1. For $k = 1$ to *popsize*, repeat Step 2.
Step 2. Randomly generate a real number r from interval $(0,1)$. The chromosome $Ch[k]$ is selected as a parent if $r < p_c$.
Step 3. Denote the selected parents as $S_1, S_2, \ldots$ and divide them into following pairs $(S_1, S_2), (S_3, S_4), \ldots$.
Step 4. For each pair of parents, for example (S_1, S_2), randomly select two positions $pos_1, pos_2 \in [1, n]$.
Step 5. Until $S_1 \& S_2$ have equal number of selected assets between $pos_1 \& pos_2$ repeat Step6.
Step 6. $pos_2 = pos_2 - 1$.
Step 7. For $i = pos_1$ to pos_2 repeat Step 8 to Step 10.
Step 8. $temp = S_1[i]$.
Step 9. $S_1[i] = S_2[i]$.
Step 10. $S_2[i] = temp$.

• **Mutation operator**
Mutation is a background operator which produces spontaneous random changes in various chromosomes, by altering one or more genes. In this paper, we use swap mutation. With some probability of mutation $p_m \in (0,1)$, a chromosome is selected for the process of mutation. The values of the genes at two randomly selected positions in the selected chromosome are then swapped to recreate a new chromosome.

4 Numerical Illustration

In what follows, we present computational results. The HIA is coded in C++ on a personal computer with Intel Core2Duo CPU, having a speed of 2.8 GHz and a 4 GB RAM. The main attributes of the problem instance solved are summarized in Table 2 and the incremental discount schemes for buying and selling of assets are given in Table 3.

Table 2. Main attributes of the problem instance solved

r_1	r_2	r_3	b	γ	h	$u_i\ \forall i$	$l_i\ \forall i$	p_c
0.18	0.004	0.12	0.25	0.30	7	0.3	0.08	0.9
p_m	*popsize*	Generation runs	p_1	p_2	p_3	s_1	s_2	s_3
0.8	30	3000	0.004	0.003	0.002	0.006	0.005	0.004
$[0,P_1]$	$[P_1,P_2]$	$[P_2,0.3]$	$[0,S_1]$	$[S_1,S_2]$	$[S_2,0.3]$			
$[0,0.15]$	$[0.15,0.2]$	$[0.2,0.3]$	$[0,0.15]$	$[0.15,0.2]$	$[0.2,0.3]$			

Table 3. Transaction cost function for buying and selling

Amount bought: x_i^+	Transaction cost: $B_i(x_i^+)$	Amount sold: x_i^-	Transaction cost: $S_i(x_i^-)$
$0 \le x_i^+ \le 0.15$	$0.004x_i^+$	$0 \le x_i^- \le 0.15$	$0.006x_i^-$
$0.15 \le x_i^+ \le 0.2$	$0.0006 + .003(x_i^+ - 0.15)$	$0.15 \le x_i^- \le 0.2$	$0.0009 + 0.005(x_i^- - 0.15)$
$0.2 \le x_i^+ \le 0.3$	$0.00075 + .002(x_i^+ - 0.2)$	$0.2 \le x_i^- \le 0.3$	$0.00115 + 0.004(x_i^- - 0.2)$

Table 4 gives the fuzzy data in respect of return and liquidity for 20 randomly selected assets, expressed in trapezoidal and general functional forms. We used historical data from National Stock Exchange, Mumbai, India, to create the fuzzy estimates.

We performed 20 HIA runs to check the stability of the solution. Table 5 shows the solution statistics.

Corresponding to the best found fitness, the credibilistic chance of the net return not less than 0.18 is obtained as 69.4091%. The chance of risk of the generated portfolio having severity loss more than 0.12 is 18.4502% which is well contained under the specified tolerance of 30%. Further, the chance of liquidity of the obtained portfolio not less than 0.004 is 75.3396%. Table 6 presents initial holding, buying proportions, selling proportions and final holding of the assets in the obtained adjusted portfolio.

As seen in Table 6, the initial portfolio has been adjusted by selling the entire holdings in assets 5, 10 and 15, investing in new assets 1, 13 and 20, selling a part of holdings in assets 3, 7 and 19 and buying some more proportion of asset 18. Transaction costs for buying and selling of individual assets as well as the total rebalancing cost are also reported in Table 6.

Table 4. Input data for 20 assets

	Asset ID Number			
	A1	A2	A3	A4
Return	(-0.38, -0.175, 0.425, 0.78)	(-0.5, -0.203, 0.547, 0.75)	(-0.383, -0.003, 0.51, 0.65)	(-0.455, -0.095, 0.625, 0.78)
Liquidity	(.0065, .0115, .0205, .032)	(.0003, .0009, .0027, .0033)	(.0003, .00125, .00515, .01145)	(.0014, .0017, .0047, .008)
	A5	A6	A7	A8
Return	(-0.265, -0.145, 0.305, 0.482)	(-0.4, -0.03, 0.51, 0.92)	(-0.38, -0.15, 0.45, 0.69)	(-0.66, -0.275, 0.475, 0.66)
Liquidity	(.0002, .000375, .000825, .001)	(.000276, .000465, .000855, .0022)	(.0008, .00115, .00205, .003)	(.000045, .0001625, .0004175, .000546)
	A9	A10	A11	A12
Return	(-0.75, -0.45, 0.52, 0.88)	(-0.68, -0.22, 0.62, 1.04)	(-0.72, -0.3, 0.3, 0.7)	(-0.74, -0.25, 0.41, 0.79)
Liquidity	(.0011875, .0013875, .0079125, .013)	(.0013, .00285, .00735, .013)	(.001, .001575, .002625, .003185)	(.000845, .0027, .0081, .015)
	A13	A14	A15	A16
Return	(-0.52, -0.275, 0.475, 0.77)	(-0.76, -0.31, 0.29, 0.86)	(-0.79, -0.18, 0.51, 0.92)	(-0.72, -0.27, 0.63, 0.85)
Liquidity	(.0008, .0022, .0082, .010773)	(.000256, .0014, .008, .015493)	(.000252, .0008, .0038, .00576)	(.000691, .00145, .00415, .005058)
	A17	A18	A19	A20
Return	(-0.58, -0.18, 0.66, 0.88)	$\frac{1}{\exp(0.32r)}, r = [2.2, 3]$	$\frac{1}{\exp(0.4r)}, r = [2.5, 3.5]$	$\frac{1}{(r+0.39)}, r = [3, 4]$
Liquidity	(.00039, .001065, .003855, .00472)	$\frac{1}{\exp(-2r)}, r = [2.4, 3]$	$\frac{1}{(350-r)}, r = [4, 6]$	$\frac{1}{\exp(8-r)}, r = [0.7, 0.9]$

Table 5. Solution statistics for 20 HIA runs

Best Fitness	Average Fitness	Standard Deviation	Coefficient of Variation(%)
0.739987	0.69375	0.02034	2.93

Table 6. Adjusted portfolio details

	Asset ID Number						
	A1	A2	A3	A4	A5	A6	A7
Initial holding (x_i^0)	0	0	0.14	0	0.23	0	0.2
Buying proportion (x_i^+)	0.089796	0	0	0	0	0	0
Selling proportion (x_i^-)	0	0	0.036992	0	0.23	0	0.05736
Readjusted new holding (x_i)	0.089796	0	0.103008	0	0	0	0.14264
Transaction cost buying $B_i(x_i^+)$	0.000359184	0	0	0	0	0	0
Transaction cost selling $S_i(x_i^-)$	0	0	0.000221952	0	0.00127	0	0.00034416
	A8	A9	A10	A11	A12	A13	A14
Initial holding (x_i^0)	0	0	0.07	0	0	0	0
Buying proportion (x_i^+)	0	0	0	0	0	0.088493	0
Selling proportion (x_i^-)	0	0	0.07	0	0	0	0
Readjusted new holding (x_i)	0	0	0	0	0	0.088493	0
Transaction cost buying $B_i(x_i^+)$	0	0	0	0	0	0.000353972	0
Transaction cost selling $S_i(x_i^-)$	0	0	0.00042	0	0	0	0
	A15	A16	A17	A18	A19	A20	
Initial holding (x_i^0)	0.06	0	0	0.12	0.18	0	
Buying proportion (x_i^+)	0	0	0	0.132714	0	0.187637	
Selling proportion (x_i^-)	0.06	0	0	0	0.049653	0	
Readjusted new holding (x_i)	0	0	0	0.252714	0.130347	0.187637	
Transaction cost buying $B_i(x_i^+)$	0	0	0	0.000530856	0	0.000712911	
Transaction cost selling $S_i(x_i^-)$	0.00036	0	0	0	0.000297918	0	
Total Transaction Cost(Buying)	0.001956923						
Total Transaction Cost(Selling)	0.00291403						
Total Rebalancing Cost	0.004870953						

5 Concluding Remarks

Based on credibility measure, in this paper, we developed a framework for portfolio adjusting/ rebalancing in fuzzy multiple criteria environment considering return, risk and liquidity. Motivated by the fact that the investor pays transaction costs while trading assets, we design a transaction cost function using incremental discount schemes, which is then used to get the net return of the portfolio. We developed the resultant model $P1$ for portfolio rebalancing and designed a HIA to solve it in most general cases of fuzzy functional forms. We also performed sensitivity analysis by testing the solution. The results of the numerical analysis shows the efficiency of the algorithm. The main advantage of the proposed model is that it is a multicriteria model and takes care of the uncertainty of the investment market when the fuzzy parameters are expressed in any general fuzzy form. The model also takes care of a more realistic situation of investment market involving the transaction costs. One can, however, identify and add more criteria for enhancing the satisfaction of the

investor. Further, the behavior of the model can be tested in other different environments with parameters such as random, fuzzy random or random fuzzy. Use of neural networks for fuzzy simulation can result in lesser computational time.

References

1. Charnes, A., Cooper, W.W.: Chance-constrained programming. Manag. Sci. 6, 73–79 (1959)
2. Chen, J.S., Hou, J.L., Wu, S.M., Chang-Chien, Y.W.: Constructing investment strategy portfolios by combination genetic algorithms. Expert. Syst. Appl. 36, 3824–3828 (2009)
3. Choi, U.J., Jang, B.G., Koo, H.K.: An algorithm for optimal portfolio selection problems with transaction costs and random lifetimes. Appl. Math. Comput. 191, 239–252 (2007)
4. Fang, Y., Lai, K.K., Wang, S.Y.: Portfolio rebalancing model with transaction costs based on fuzzy decision theory. Eur. J. Oper. Res. 175, 879–893 (2006)
5. Gupta, P., Mehlawat, M.K., Saxena, A.: Asset portfolio optimization using fuzzy mathematical programming. Inform. Sci. 178, 1734–1755 (2008)
6. Gupta, P., Mehlawat, M.K., Saxena, A.: A hybrid approach to asset allocation with simultaneous consideration of suitability and optimality. Inform. Sci. 180, 2264–2285 (2010)
7. Gupta, P., Inuiguchi, M., Mehlawat, M.K.: A hybrid approach for constructing suitable and optimal portfolios. Expert. Syst. Appl. 38, 5620–5632 (2011)
8. Hasuike, T., Katagiri, H., Ishii, H.: Portfolio selection problems with random fuzzy variable returns. Fuzzy Set. Syst. 160, 2579–2596 (2009)
9. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
10. Huang, X.: A review of credibilistic portfolio selection. Fuzzy Optim. Decis. Making 8, 263–281 (2009)
11. Kozhan, R., Schmid, W.: Asset allocation with distorted beliefs and transaction costs. Eur. J. Oper. Res. 194, 236–249 (2009)
12. Li, X., Liu, B.: A sufficient and necessary condition for credibility measures. Int. J. Uncertainity, Fuzziness Knowledge-Based Syst. 14, 527–535 (2006)
13. Li, X., Qin, Z., Kar, S.: Mean-Variance-Skewness model for portfolio selection with fuzzy returns. Eur. J. Oper. Res. 202, 239–247 (2010)
14. Liu, B.: Theory and Practice of Uncertain Programming. Physica-Verlag, Heidelberg (2002)
15. Liu, B., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. IEEE Trans. Fuzzy Syst. 10, 445–450 (2002)
16. Liu, B.: Uncertainty Theory: An Introduction to Its Axiomatic Foundations. Springer, Berlin (2004)
17. Markowitz, H.: Portfolio selection. J. Finance 7, 77–91 (1952)
18. Morton, A.J., Pliska, S.R.: Optimal portfolio management with transaction costs. Math. Finance 5, 337–356 (1995)
19. Murata, T., Ischibuchi, H.: MOGA: multi-objective genetic algorithms. In: Proceedings of the Second IEEE International Conference on Evolutionary Computation, pp. 289–294 (1995)
20. Nahmias, S.: Production and Operations Analysis. McGraw/Irwin, New York (2001)

21. Qin, Z., Li, X., Ji, X.: Portfolio selection based on fuzzy cross-entropy. J. Comput. Appl. Math. 228, 139–149 (2009)
22. Yu, J.R., Lee, W.Y.: Portfolio rebalancing model using multiple criteria. Eur. J. Oper. Res. 209, 166–175 (2011)
23. Zadeh, L.A.: Fuzzy sets. Inform. Contr. 8, 338–353 (1965)
24. Zadeh, L.A.: Towards a generalized theory of uncertainity (GTU)-an outline. Inform. Sci. 172, 1–40 (2005)
25. Zhang, W.G., Nie, Z.K.: On admissible efficient portfolio selection problem. Appl. Math. Comput. 159, 357–371 (2004)
26. Zhang, X., Zhang, W.G., Cai, R.: Portfolio adjusting optimization under credibility measures. J. Comput. Appl. Math. 234, 1458–1465 (2010)

Dynamic Testing Resource Allocation of Modular Software System for SRGM Incorporating Testing Efficiency Using Differential Evolution

Kuldeep Chaudhary[1], Prerna Manik[1], and Shivani Bali[2]

[1] Department of Operational Research, University of Delhi, Delhi-110007, India
[2] Lal Bahadur Shastri Institute of Management, Dwarka, Delhi, India
{chaudharyiitr33,prernamanik,lbsshivani}@gmail.com

Abstract. Software reliability engineering has recently been playing a rapidly increasing role in the industry. This has occurred because it carefully plans and guides development and test so that software developing team develop a more reliable product faster and cheaper. Practically, a software testing process consists of several testing stages. A project manager should know how to allocate the specified testing resources among all the modules and develop quality software with high reliability. In this paper, we investigate an optimal resource allocation problem in modular software systems for flexible Software Reliability Growth Model (SRGM). The optimization model proposed for optimal allocation of testing resources among the modules of software subject to budgetary constraint is formulated and solution is obtained using Differential Evolution which stands as a powerful tool for solving search and optimization problems. The allocated testing resources for each of the modules are utilized to detect and remove the faults. The parameters are re-estimated using all the available fault detection data for each of the modules. The new allocation problem is re-formulated and allocation of testing resources for each of the modules is obtained. The process is terminated either the desired level of faults are removed from each of the modules or specified number of iterations is exhausted. The above problem has been discussed using numerical example to illustrate the applicability of the approach.

Keywords: Software Reliability, Non-Homogenous Poisson Process, Testing-Time Allocation, Modular Software System, Differential Evolution.

1 Introduction

The application of computer systems has now crossed many different fields. Systems are becoming more software intensive. Many flight systems are becoming more software intensive. Financial systems including teller, automated teller and loan processing are software intensive. Everything from insurance rates to credit

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1011–1023.
springerlink.com

histories to hotel reservations to long distance telephone calls is performed by software. Software affects our daily lives. A software life cycle consists mainly of the following phases, requirement and specification, design, coding, testing and operations/maintenance. An important phase during software development is its testing phase .During the software testing phase, programs are executed and erroneous outputs are identified. For each incorrect output, we may count it as a failure. Faults that cause the failure are identified and removed. The failure process during the software testing phase may be identified as a fault removal process. The reliability of the software will be increased during the testing phase as more and more faults are removed. The reliability improvement phenomenon is called reliability growth. Software Reliability Engineering (SRE) is a discipline that aims at ensuring high reliability of the software experienced by the user, which employ scientific tools and techniques during software testing to remove as many latent faults in the software as possible. Software Reliability Growth Models (SRGMs) are the tools of SRE used to estimate and predict the reliability of software during testing and operational phase. Many SRGMs are used to estimate the software reliability measures such as remaining number of faults, failure rate and reliability growth during the testing phase.

Many SRGMs have been proposed and studied in the literature under different set of assumptions. Some SRGMs describe the failure phenomenon by an exponential [3] or an S-shaped [21] pattern. While some of the SRGMs are more flexible in the sense that they can describe, depending upon parameter values, both exponential and S-shaped failure patterns (Obha [15], Bittani et al [2], and Kapur and Garg [11], etc.). Similar SRGMs that describe the failure phenomenon with respect to testing efforts are developed in literature [7, 9,10,11, 12, ,16].

During testing, resources such as manpower and time (computer time) are consumed. The failure (fault identification) and removal are dependent upon the nature and amount of resources spent. The time dependent behavior of the testing effort has been studied earlier by Basili et al [1], Huang [6], Musa et al. [14], Putman [18] and Yamada et al. [23]. The exponential and Rayleigh curves are used to describe the relationship between the testing effort consumption and testing time (the calendar time). Exponential curve is used if the testing resources are uniformly consumed with respect to the testing time and Rayleigh curve otherwise. Logistic and Weibull type functions have also been used to describe testing effort. In this paper we propose a Non Homogeneous Poisson Process (NHPP) based SRGM with testing effort (described by any of the curves mentioned above), to represent the growth curve of the fault removal process of the Software which contains two types of faults: mutually independent and mutually dependent incorporating testing efficiency (imperfect debugging). This model captures features of both exponential and S-shaped growth curves for failure (fault identification) phenomenon.

The software modules are coded and then they are tested. The testing process usually consists of three stages - module testing, integration testing and acceptance testing. In integration testing, all the software modules are interconnected according to predetermined logical structure. In acceptance testing the software system is tested by customers or is tested using the test sets supplied by the customer. It has

been estimated that, for large scale software development, resources are expended in software testing. Therefore the software project manager should monitor the testing process closely and effectively allocate the resources in order to reduce the testing cost and to meet the given reliability requirements. All the testing activities of different modules should be completed within a specified time. Typically, module testing is the most time-critical part of testing to be performed. Scope of this paper is restricted to resource allocation problem for module testing (unit testing) level. It may be noted most of the allocation problems studied in literature are related to the allocation of testing resources to each of the module at the module testing level considering each module to be independent of each other. However dependency of modules can also be considered with ease if we optimize the resource allocation at the system testing level when on an input a sequence of modules are called to get the desired output. At the module testing level modules can considered to be independent of each other since they are designed independently. Since the allocation problem discussed in this paper considers the testing at the modular level therefore we consider the modules to be independent of each other. Each module may contain different number of faults and that of different severity. Hence fault removal phenomenon in modules can be represented through distinct SRGMs. In this paper, we propose a dynamic resource allocation model for software module testing where faults are detected using Differential Evolution (DE). This strategy considers the number of faults detected in each module as module testing is proceeding, re-estimates the model parameters using all the available fault detection data, and adjusts the resource allocation dynamically. Differential Evolution (DE) stand up as a powerful tool for solving search and optimization problem. The reason of using differential evolution in the field of Software Reliability is capability to give near optimal results using historical data. As Differential Evolution (DE) is an evolutionary algorithm where automation is the key objective which says the way our human body system is automated, the same way we want our computer software systems to operate automatically once the failure occurs [17,19]. Therefore, in this paper Differential Evolution (DE) is used to search for the near optimal solution for modular software system with the objective of maximizing the number of faults detected under the constraints of availability of limited testing resource expenditure for each module.

2 SRGM with Testing Effort

In this paper, we discuss and solve such a management problem of allocation of testing resources among modules, through a Software Reliability Growth Model (SRGM). SRGM is a relationship between the number of faults removed from software and the execution time / CPU time / calendar time. Several attempts have been made to represent the actual testing environment through SRGM's. [4,8,22]. These models are used to predict the fault content, reliability and release time of software. The influence of testing effort has also been included in some SRGMs [5, 8, 11, 20,23, 24].

2.1 Model Formulation

The parameters of an SRGM should be interpretable in terms of software testing phenomenon and one such popular model is due to Ohba [15]. Other similar models are due to Bittanti et al. [2] and Kapur and Garg [13]. Ohba proposed that the fault removal rate increases with time and assumed the presence of two types of faults in the software. The distinctive Notations and assumptions of the proposed model can be summarized as follows:

Notations:

M : Number of modules in a software (> 1)

v_i : Relative importance of software module i

a_{ij} : Mean number of remaining faults in module i^{th} at the beginning of j^{th} testing period.

w_{ij} : Resource expenditure rate for i^{th} module at the beginning of j^{th} testing period.

$m_{ij}(t)$: Mean number of detected faults in i^{th} module by time t at the beginning of j^{th} testing period.

b_{ij} : Proportionality constant for the i^{th} module at the beginning of j^{th} testing period.

α_{ij} : Constant rate of error generation for i^{th} module at the beginning of j^{th} testing period.

p_{ij} : Probability of perfect debugging for i^{th} module at the beginning of j^{th} testing Period.

z_{ij} : Number of remaining faults in i^{th} module, when j^{th} testing period ends.

q_{ij} : Total amount of resources allocated to i^{th} module in the j^{th} testing period.

Q_j : Total amount of resource allocated to the j^{th} testing Period.

Assumptions

1. Let the software system be composed of M software modules.
2. During software module testing, these modules are tested independently and simultaneously [16].
3. The software failure phenomenon is described by Non-Homogeneous Poisson Process (NHPP).
4. Software is subject to failures during execution caused by faults remaining in the software.
5. Each time a failure is observed, an immediate effort takes place to decide the cause of the failure in order to remove it.
6. Reliability growth during the testing phase is dependent upon the testing efforts spend on testing.
7. During the fault removal process, following may occur:(a) Fault content is reduced by one with probability p (b) Fault content remains unchanged with probability $1-p$.

8. During the fault removal process, new faults can be generated and the fault generation rate is proportional to the rate of fault removal.
9. Fault removal rate per remaining fault is assumed to be non-decreasing inflection S-shaped logistic function.

Under the above assumptions, the removal phenomenon can be described with respect to testing effort as follows:

$$\frac{\frac{d}{dt}m(t)}{w(t)} = p\ b\big(W(t)\big)(a(t)-m(t)) \tag{2.1}$$

Where

$$b(W(t)) = \frac{b}{1+\beta e^{-bW(t)}} \text{ and } a(t) = \big(a+\alpha m(t)\big) \tag{2.2}$$

Solving equation (2.1) with the initial condition that, at $t = 0$, $m(t) = 0$,W(0)=0, we get

$$m(t) = \frac{a}{1-\alpha}\left[1-\left(\frac{(1+\beta)e^{-bW(t)}}{1+\beta e^{-bW(t)}}\right)^{p(1-\alpha)}\right] \tag{2.3}$$

Where, $W(t) = \int_0^t w(u)du$

If $\beta = 0$, then $m(t) = \frac{a}{1-\alpha}\left[1-e^{-bp(1-\alpha)W(t)}\right]$. Since the growth model can exponential as well as flexible under imperfect testing efficiency, both the possibilities must be considered.

2.2 The Model

Let the j^{th} testing period start at T_j and end at T_{j+1} (refer Figure 1). In the j^{th} testing period, the mean number of remaining faults in module i by time t $(T_j < t \le T_{j+1})$ is equal to $\left[a_{ij}(t)-m_{ij}(t)\right]$. Under assumptions, given in Section 2.1, $m_{ij}(t)$ satisfies the following differential equation

$$\frac{\frac{d}{dt}m_{ij}(t)}{w_{ij}(t)} = p_{ij}\ b_{ij}\big(W(t)\big)(a_{ij}(t)-m_{ij}(t)) \tag{2.4}$$

Where $b_{ij}(W(t)) = \frac{b_{ij}}{1+\beta_{ij}\ e^{-b_{ij}W_{ij}(t)}}$ and $a_{ij}(t) = \big(a_{ij}+\alpha_{ij}\ m_{ij}(t)\big)$

The initial condition is $m_{ij}(t) = 0 \; at \; \mathrm{t} = \mathrm{T_j}$. By solving the above differential equation, $m_{ij}(t)$ can be found to be

$$m_{ij}(t) = \frac{a_{ij}}{1-\alpha_{ij}} \left[1 - \left(\frac{\left(1+\beta_{ij}\right) e^{-b_{ij} W_{ij}(t-T_j)}}{1+\beta_{ij}\, e^{-b_{ij} W_{ij}(t-T_j)}} \right)^{p_{ij}\left(1-\alpha_{ij}\right)} \right] \tag{2.5}$$

Putting $\beta_{ij} = 0$,we get exponential SRGM as follows

$$m_{ij}(t) = \frac{a_{ij}}{1-\alpha_{ij}} \left[1 - e^{-p_{ij}\left(1-\alpha_{ij}\right) b_{ij} W_{ij}(t-T_j)} \right]$$

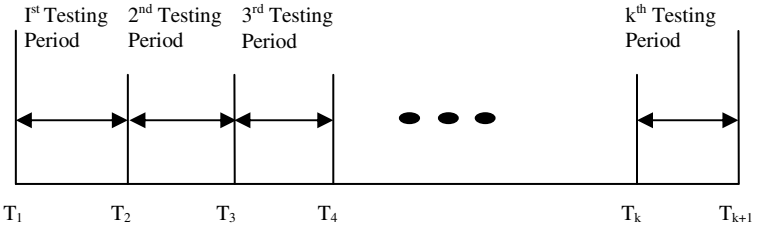


Fig. 1. Testing Periods

3 Dynamic Resource Allocation

When only limited resources are available in the testing of a redundant software system, it is important to allocate the testing resources efficiently so that the maximum reliability of the complete system is achieved. We assume the total testing time for software module testing is divided into K testing periods. In each testing period, the software project manager records total number of detected faults in each software module. At the end of each testing period, the project manager estimates the mean number of remaining faults in each software module and a model parameter based on all the recorded fault detection data. These estimates help manager in determining the amount of resources that should be allocated to each software module in the next testing period.

3.1 Optimal Allocation of Resources

From the estimates of parameters of SRGMs for modules, the total fault content in the software $\sum_{i=1}^{M} a_{ij}$ is known. Module testing aims at detecting maximum number of faults within available time. We consider the j^{th} period

$$Max \sum_{i=1}^{M} v_i m_{ij}(t) = \sum_{i=1}^{l} \frac{v_i a_{ij}}{1-\alpha_{ij}} \left[1 - e^{-b_{ij} q_{ij} p_{ij}(1-\alpha_{ij})}\right] + \sum_{i=l+1}^{M} \frac{v_i a_{ij}}{1-\alpha_{ij}} \left[1 - \left(\frac{(1+\beta_{ij}) e^{-b_{ij} q_{ij}}}{1+\beta_{ij} e^{-b_{ij} q_{ij}}}\right)^{p_{ij}(1-\alpha_{ij})}\right] \quad \text{(P1)}$$

Subject to $\sum_{i=1}^{M} q_{ij} = Q_j$, $q_{ij} \geq 0$, $i = 1, 2, ..., M$

Where q_{ij} is the total amount of resources allocated to module ***i*** in the j^{th} testing period $q_{ij} = W_{ij}(T_{j+1} - T_j)$ and $i = 1, \ldots, M$ and $j = 1, \ldots, K$. The objective function is the predicted mean fault content at the end of the j^{th} testing period. The first constraint ensures that the total amount of resources allocated to the software modules in the j^{th} testing period is Q_j and second constraint ensures that q_{ij} is non negative. This optimization problem in which only flexible SRGM for each module of the software is considered for numerical illustration is solved using Differential Evolution. The Differential Evolution Algorithm is presented in the following section.

4 Differential Evolution Algorithm for Resource Allocation

Differential Evolution (DE) is an exceptionally simple evolution algorithm, which is rapidly growing field of artificial intelligence. This class also includes GAs, evolutionary strategies and evolutionary programming. DE was proposed by Price & Storn in1995. It was developed to optimize real parameter and real valued function. The basic steps of DE are described in detail as follows:

Step 1: Start
Step 2: Generate an initial population of random individuals
Step 3: Create a new population by repeating following steps until the stopping criterion is achieved

- [Selection] Select the random individuals for reproduction
- [Reproduction] Create new individuals from selected ones by mutation and crossover
- [Evolution] Compute the fitness values of the individuals
- [Advanced Population] Select the new generation from target individual &trial individuals.

Suppose we want to optimize a function of *D* real parameters. We must select the size of the population NP. NP parameter vectors have the form

$$X_{i,G} = \left(x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, ..., x_{D,i,G}\right)$$

Where: NP is number of population vectors, D is dimension & i is individual index and G is the number of generation.

4.1 Intialization

First, all solution vectors in a population are randomly initialized. The initial solution vectors are generated between lower and upper bounds $l = \{l_1, l_2, ..., l_D\}$ *and* $u = \{u_1, u_2, ..., u_D\}$ using the equation

$$x_{j,i,0} = l_j + rand_{i,j}[0,1] \times (u_j - l_j)$$

Where, i is individual index, j is component index and $rand_{i,j}$ [0, 1] is a uniformly distributed random number lying between 0 and 1.

4.2 Mutation

Each of the NP parameter vectors undergoes mutation, recombination and selection. Mutation expands the search space. For a given parameter vector $X_{i,G}$ randomly selected three vectors $X_{r_1,G}, X_{r_2,G}, X_{r_3,G}$ such that the indices i, r_1, r_2, r_3 are distinct. The ith perturbed individual, $V_{i,G}$, is therefore generated based on the three chosen individuals as follows:

$$V_{i,G} = X_{r_1,G} + F * (X_{r_2,G} - X_{r_3,G})$$

Where $r_1, r_2, r_3 \in \{1, \ldots\ldots, NP\}$ are randomly selected, such that $r_1 \neq r_2 \neq r_3 \neq i$, $F \in (0, 1.2]$. $V_{i,G}$ is called mutation vector.

4.3 Crossover

The perturbed individual, $V_{i,G} = (v_{1,i,G}, \ldots v_{D,i,G})$, and the current population member, $X_{i,G} = (x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \ldots, x_{D,i,G})$ are then subject to the crossover operation, that finally generates the population of candidates, or "trial" vectors, $U_{i,G} = (u_{1,i,G}, \ldots, u_{D,i,G})$, as follows:

$$u_{j,i.G} = \begin{cases} v_{j,i.G} \; if \; rand_{i,j}[0,1] \leq C_r \vee j = j_{rand} \\ x_{j,i.G} \qquad\qquad otherwise \end{cases}$$

Where $C_r \in [0, 1]$, is crossover probability, $j_{rand} \in \{1, \ldots, D\}$ is a random parameter's index, chosen once for each i.

4.4 Selection

The population for the next generation is selected from the individuals in current population and its corresponding trial vector according to the following rule:

$$X_{i.G+1} = \begin{cases} U_{i.G} \; if \; f(U_{i.G}) \geq f(X_{i.G}) \\ X_{i.G} \qquad\qquad otherwise \end{cases}$$

Where $f(,)$ is the objective function value. Each individual of the temporary population is compared with its counterpart in the current population. Mutation, recombination and selection continue until stopping criterion is reached.

4.5 Constraint Handling in Differential Evolution

Pareto ranking method is used to handle constraints in DE. The value of constraints is calculated at target and trial vectors. The method is based on the three rules: 1). Between two feasible vectors (target and trial), the one with the best value of the objective function is preferred. 2).If out of target and trial vector, one vector is feasible and the other one is infeasible, the feasible one is preferred. 3).Between two infeasible vectors, the one with the lowest sum of constraint violation is preferred.

4.6 Stopping Criterion

Algorithm stopped when either 1).Maximum number of generations is reached or 2). Desired accuracy is achieved i.e. $|f_{\max} - f_{\min}| \leq \varepsilon$

5 Numerical Illustration

We assume that the duration of all the testing periods are equal, the number of software modules M is 6 and the parameters a_{ij}, b_{ij} , α_{ij} , β_{ij} , v_{ij} and p_{ij} ; for the i^{th} module $(i = 1,.....,M)$ have already been estimated from the failure data shown in Table 1. Three testing stages have been considered in this paper. Suppose the total resource available for all the three stages is 40,000. In the first testing period 15,000 of the total resources have been allocated. Problem is solved using DE. The parameters used in DE are given in Table 2. The optimal testing allocation to each module and their corresponding fault detection is shown in Table 3. The total amount of testing expenditure and maximum number of faults detection respectively are1499.82, 1026.

Table 1. Parameter Estimates used for resource allocation problem of 1st Testing Period

Module	a_{i1}	b_{i1}	α_{i1}	β_{i1}	v_{i1}	p_{i1}
M1	218	0.000751	0.0344	1.9736	0.154391	0.95
M2	105	0.001155	0.06513	1.5321	0.074363	0.91
M3	231	0.000617	0.0959	1.4129	0.163598	0.90
M4	286	0.000652	0.1148	1.7276	0.20255	0.92
M5	260	0.000980	0.10433	2.3697	0.184136	0.91
M6	312	0.000653	0.0123	1.6321	0.220963	0.99

Table 2. Parameter of the DE

Parameter	Value
Population Size	100
Selection Method	Roulette Wheel
Scaling Factor(F)	.7
Crossover Probability(Cr)	.9

Table 3. The optimal testing expenditure of each module

Module	w_{i1}	No. of fault detection
M1	2322.85	136
M2	1003.43	48
M3	2356.17	141
M4	3299.34	234
M5	2898.17	238
M6	3119.86	226
Total	1499.82	1026

Table 4, is the second phase of testing, the model parameters are re-estimated, based on the fault detection data up to the 1^{st} testing period. Based on the estimated data the resources will be dynamically allocated to those modules where the remaining number of faults in system is large in number. The remaining number of faults is 496 and the resource allocated to this testing period is 15000. The total number of faults that can be removed through this allocation is 161.

Table 4. Parameter Estimates used for resource allocation problem of 2^{nd} Testing Period

Module	a_i	b_i	α_i	β_i	v_i	p_i
M1	89	0.000741	0.03540	1.9836	0.178893694	0.95
M2	64	0.001255	0.06451	1.5421	0.092253331	0.91
M3	114	0.000717	0.09459	1.4229	0.154471332	0.90
M4	88	0.000682	0.11580	1.7376	0.195103810	0.92
M5	51	0.000970	0.10533	2.3797	0.204839520	0.91
M6	90	0.000753	0.01330	1.6421	0.174438313	0.99

Table 5. The optimal testing expenditure of each module

Module	w_i	No. of fault detection
M1	2780.20	89
M2	1875.91	64
M3	3373.33	114
M4	2929.85	88
M5	1441.70	51
M6	2598.42	90
Total	1499.41	496

Table 6, is the third phase of testing, ideally the model parameters are re-estimated, based on the fault detection data up to the 2^{nd} testing period. The remaining number of faults in the system is 161; the resource allocated to this testing period is 10,000. The total number of faults that can be removed through this allocation is 137.

Table 6. Parameter Estimates used for resource allocation problem of 3rd Testing Period

Module	a_i	b_i	α_i	β_i	v_i	p_i
M1	29	0.001741	0.03340	1.9826	0.105915519	0.95
M2	15	0.001255	0.06431	1.5321	0.226626951	0.91
M3	25	0.000717	0.09459	1.4329	0.408761176	0.90
M4	31	0.001982	0.11480	1.7376	0.073421292	0.92
M5	33	0.001970	0.10533	2.3697	0.073242288	0.91
M6	28	0.001753	0.01230	1.6314	0.112032774	0.99

Table 7. The optimal testing expenditure of each module

Module	w_{i3}	No. of fault detection
M1	1781.54	26
M2	1070.28	8
M3	1766.61	13
M4	1804.73	32
M5	1947.64	34
M6	1628.89	24
Total	9999.69	161

The optimal allocation to each module and their corresponding number of faults detection for each testing period 1st, 2nd & 3rd shown in Table 3, 5 and 7 respectively.

6 Conclusion

This paper discusses an optimization problem occurring during the module testing phase of the software development life cycle. We have proposed a dynamic resource allocation model for software module testing, which considers the number of faults detected in each module. As module testing proceeds, it re-estimates the model parameters using all the available fault detection data, and adjusts the resource allocation dynamically. We have observed how the testing resources have been dynamically allocated to the modules in different testing periods. Higher weights can be attached to the modules having large number of faults. They are then allocated more resources. The process of allocation of resources stops either when pre-specified numbers of testing period ends or remaining faults meet developer's aspiration. Differential Evolution approach has been used to find the optimal solution of the proposed model.

References

[1] Basili, V.R., Zelkowitz, M.V.: Analyzing medium scale software development. In: Proceedings of the 3rd International Conference on Software Engineering, pp. 116–123 (1979)

[2] Bittanti, et al.: A Flexible Modeling Approach for Software Reliability Growth. In: Bittanti, S. (ed.) Software Reliability Modelling and Identification. LNCS, vol. 341, pp. 101–140. Springer, Heidelberg (1988)

[3] Goel, A.L.: Software Reliability Models: Assumptions, limitations and applicability. IEEE Trans. on Software Engineering 11, 1411–1423 (1985)

[4] Goel, A.L., Okumoto, K.: Time dependent error detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability 28(3), 206–211 (1979)

[5] Huang, et al.: Optimal allocation of testing- resources considering cost, reliability, and testing-effort. In: Proceedings of 10th IEEE Pacific International Symposium on Dependable Computing, pp. 103–112 (2004)

[6] Huang, C.Y., Kuo, S.Y., Chen, J.Y.: Analysis of a software reliability growth model with logistic testing effort function. In: Proceeding ISSRE 1997, pp. 378–388 (1997)

[7] Huo, et al.: Needed resources for software module test, using the hyper-geometric software reliability growth model. IEEE Trans. on Reliability 45(4), 541–549 (1996)

[8] Ichimori, T., Yamada, S., Nishiwaki, M.: Optimal allocation policies for testing-resource based on a Software Reliability Growth Model. In: Proceedings of the Australia–Japan Workshop on Stochastic Models in Engineering, Technology and Management, pp. 182–189 (1993)

[9] Kapur, P.K., Jha, P.C., Bardhan, A.K.: Optimal allocation of testing resource for modular software. Asia Pacific Journal of Operational Research 21(3), 333–354 (2004)

[10] Kapur, P.K., Garg, R.B.: Cost reliability optimum release policies for a software system with testing effort. OPSEARCH 27(2), 109–118 (1990)

[11] Kapur, P.K., Garg, R.B., Kumar, S.: Contributions to Hardware and Software Reliability. World Scientific, Singapore (1999)

[12] Kapur, et al.: Dynamic programming approach to testing resource allocation problem for modular software. Ratio Mathematica 14, 27–40 (2003)

[13] Kapur, P.K., Garg, R.B.: A software reliability growth model for an error removal phenomenon. Software Engineering Journal 7, 291–294 (1992)

[14] Musa, J.D., Iannino, A., Okumoto, K.: Software reliability: Measurement, Prediction, Applications. Mc-Graw Hill, New York (1987)

[15] Ohba, M.: Software reliability analysis models. IBM Journal of Research and Development 28, 428–443 (1984)

[16] Ohetera, H., Yamada, S.: Optimal allocation and control problems for software testing resources. IEEE Transactions on Reliability 39(2), 171–176 (1990)

[17] Price, K.: An introduction to Differential Evolution. In: Corne, D., Marco, D., Glover, F. (eds.) New Ideas in Optimization, pp. 78–108. McGraw-Hill, London (1999)

[18] Putnam, L.: A general empirical solution to the macro software sizing and estimating problem. IEEE Transactions on Software Engineering 4, 345–361 (1978)

[19] Storn, R., Price, K.: DE-a simple and efficient adaptive scheme for global optimization over continuous space. TR-95-012, ICSI (1995), http://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95-012.ps.Z

[20] Xie, M.: Software reliability modeling. World Scientific, Singapore (1991)

[21] Yamada, et al.: S-shaped software reliability growth modeling for software error detection. IEEE Trans. on Reliability 32(5), 475–484 (1983)

[22] Yamada, S., Ohtera, H., Narihsia, H.: Software Reliability Growth Model with testing effort. IEEE Trans. on Reliability 35, 19–23 (1986)

[23] Yamada, S., Hishitani, J., Osaki, S.: Software Reliability Growth Model with Weibull testing effort: A model and application. IEEE Trans. on Reliability 42, 100–105 (1993)

[24] Yiu-Wing, L.: Dynamic Resource Allocation for Software Module Testing. J. Systems Software 37, 129–139 (1997)

Fuzzy Multi-objective Build-or-Buy Approach for Component Selection of Fault Tolerant Modular Software System under Consensus Recovery Block Scheme

Shivani Bali[1], Anshu Gupta[2], and U. Dinesh Kumar[3]

[1] Lal Bahadur Shastri Institute of Management, Dwarka, Delhi, India
[2] Department of Operational Research, University of Delhi, Delhi, India
[3] Indian Institute of Management, Banglore, India
lbsshivani@gmail.com, anshu_or@yahoo.com, dineshk@iimb.ernet.in

Abstract. Computers are used in diverse areas for various applications including air traffic control, nuclear reactors, industrial process control, hospital health care etc., affecting millions of people. As the functionality of computer operations become more essential and yet more critical, there is a great need for the development of modular software system. Component-Based Software Engineering (CBSE) is concerned with composing, selecting and designing components to satisfy a set of requirements while minimizing cost and maximizing reliability of the software system. This paper discusses the "Build-or-Buy" strategy for component selection in designing a software structure. A fuzzy multi objective optimization model for selecting components based on the dual objective of maximizing reliability and minimizing the overall cost of the software system under delivery time constraint is formulated. Numerical illustrations are provided to demonstrate the model developed.

Keywords: Modular Software, Software Reliability, Fault Tolerant Software, Consensus Recovery Block Scheme, Build-or-Buy.

1 Introduction

Software permates every aspect of modern society. Government, transportation, manufacturing, utilities, and almost every other sector that influences our way of life is affected directly or indirectly by software systems. Without software, many of our modern conveniences would be virtually impossible. Despite its widespread use, software is hardly ever "perfect". For a myriad of reasons, it is extremely difficult to produce a flawless piece of software. Software engineering is the discipline concerned with the establishment and application of sound engineering

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1025–1036.
springerlink.com

practices to the development of reliable and efficient software. Because absolute certainty of design correctness is rarely achieved, software fault tolerance techniques are sometimes employed to meet design dependability requirements. Software fault tolerance refers to the use of techniques to increase the likelihood that the final design embodiment will produce correct and/or safe outputs.

Component-Based Software Engineering (CBSE) is concerned with composing, selecting and designing components [5]. In Component-based Software Engineering (CBSE), the construction of cost-reliability -optimal component systems is not a trivial task. It requires not only to optimally select components but also to take their interplay into account. While developing software, components can be both bought as COTS products, and probably adapted to work in the software system, or they can be developed in-house. This decision is known as "build or buy" decision. In this paper, we address the problem of (automatic) component selection. Informally, our problem is to select a set of components from available component set which can satisfy a given set of requirements while minimizing sum of the costs and maximizing the reliability of selected components. Each COTS component is assigned a cost which is the overall cost of acquisition, delivery time which is the duration of acquiring a component and also the reliability of the component. For each build component we estimate the parameters of cost, delivery time and reliability of the component, whereas the parameters of the COTS components are given by its vendor. This paper discusses a framework that helps developers to decide whether to build or to buy software components while designing a software system on the base of cost and non functional factors. The novelty of this paper is that we have applied fuzzy multi – objective approach to optimal decision "build-or-buy" for component selection for a fault tolerant modular software system under Consensus Recovery Block scheme.

2 Notations

R : System quality measure

C : Overall cost of the system

f_l : Frequency of use, of function l

s_l : Set of modules required for function l

R_i : Reliability of module i

L : Number of functions, the software is required to perform

n : Number of modules in the software.

m_i : Number of alternatives available for module i

t_1 : Probability that next alternative is not invoked upon failure of the current alternative

t_2 : Probability that the correct result is judged wrong.

t_3 : Probability that an incorrect result is accepted as correct.

X_{ij} : Event that output of alternative j of module i is rejected.

Y_{ij} : Event that correct result of alternative j of module i is accepted.

r_{ij} : Reliability of alternative j of module i

C_{ij} : Cost of alternative j for module i of a COTS component

s_{ij} : Reliability of alternative j of module i of a COTS component

d_{ij} : Delivery time of alternative j of module i of a COTS component

c_{ij} : Unitary development cost for alternative j of module i

t_{ij} : Estimated development time for alternative j of module i

τ_{ij} : Average time required to perform a test case for alternative j of module i

π_{ij} : Probability that a single execution of software fails on a test case chosen from a certain input distribution

y_{ij} : $\begin{cases} 1 & \text{if the } j\text{th alternative of } i\text{th module is in-house developed.} \\ 0 & \text{otherwise} \end{cases}$

x_{ij} : $\begin{cases} 1, & \text{if the } j^{th} \text{ COTS alternative of the } i^{th} \text{ module is chosen} \\ 0, & \text{otherwise} \end{cases}$

z_{ij} : $\begin{cases} 1, & \text{if alternative } j \text{ is present in module } i \\ 0, & \text{otherwise} \end{cases}$

3 Optimization Models

Large software system has modular structure to perform set of functions with different modules having different alternatives for each module. A schematic representation of the software system is given in figure 1, in section 4.3. We are selecting the components (in-house or COTS) for modules to maximize the system reliability by simultaneously minimizing the cost. The frequency with which the functions are used is not same for all of them and not all the modules are called during the execution of a function, the software has in its menu. The first optimization model is developed under the following assumptions, which also holds good for the second model, but with additional assumptions related to compatibility among alternatives of a module. The following assumptions common for both the optimization models are as follows:

1. A software system consists of a finite number of modules.
2. A software system is required to perform a known number of functions. The program written for a function can call a series of modules $(\leq n)$. A failure occurs if a module fails to carry out an intended operation.
3. Codes written for integration of modules don't contain any bug.

4. Several alternatives are available for each module. Fault tolerant architecture is desired in the modules (it has to be within the specified budget). Independently developed alternatives (primarily COTS or In-house build components) are attached in the modules and work similar to the consensus recovery block scheme discussed in [2, 3].
5. The cost of an alternative is the development cost, if developed in house; otherwise it is the buying price for the COTS product. Reliability for all the COTS components are known and no separate testing is done.
6. Different COTS alternatives with respect to cost, reliability and delivery time of a module are available.
7. Different In-house alternatives with respect to unitary development cost, estimated development time, average time and testability of module are available.
8. Cost and reliability of an in-house component can be specified by using basic parameters of the development process, e.g. a component cost may depend on a measure of developer skills, or the component reliability depends on the amount of testing.

3.1 Model Formulation

Let S be a software architecture made of n modules, with a maximum number of m_i alternatives available for each module.

3.1.1 Build Versus Buy Decision

For each module i, if an alternative is bought (i.e. some $x_{ij} = 1$) then there is no in-house development (i.e. $y_{ij} = 0$) and vice versa.

$$y_{ij} + x_{ij} \leq 1;\ i = 1,2,....,n \text{ and } j = 1,2,....,m_i$$

3.1.2 Redundancy

The equation stated below guarantees that redundancy is allowed for both the build and buy components (i.e. in-house and COTS components).

$$y_{ij} + x_{ij} = z_{ij} \qquad \sum_{j=1}^{m_i} z_{ij} \geq 1;\ i = 1,2,....n$$

3.1.3 Probability of Failure Free In-House Developed Component

The possibility of reducing the probability that the j^{th} alternative of i^{th} module fails by means of a certain amount of test cases (represented by the variable N_{ij}^{tot}). Cortellessa *et al* [8] define probability of failure on demand of an in-house developed j^{th} alternative of i^{th} module, under the assumption that the on-field users' operational profile is the same as the one adopted for testing [4]. Basing on the

testability definition, we can assume that the number N_{ij}^{suc} of successful (i.e. failure free) tests performed on j^{th} alternative of same module.

$$N_{ij}^{suc} = \left(1-\pi_{ij}\right) N_{ij}^{tot} \quad ; i = 1,2,....,n \text{ and } j = 1,2,....,m_i$$

Let A be the event " N_{ij}^{suc} failure – free test cases have been performed "and B be the event "the alternative is failure free during a single run ".If ρ_{ij} is the probability that in-house developed alternative is failure free during a single run given that N_{ij}^{suc} test cases have successfully performed, from the Bayes Theorem we get

$$\rho_{ij} = P(B/A) = \frac{P(A/B)P(B)}{P(A/B)P(B) + P(A/\overline{B})P(\overline{B})}$$

The following equalities come straightforwardly:

$$\bullet\ P(A/B) = 1 \qquad \bullet\ P(B) = 1-\pi_{ij} \qquad \bullet\ P(A/\overline{B}) = (1-\pi_{ij})^{N_{ij}^{suc}} \qquad \bullet\ P(\overline{B}) = \pi_{ij}$$

Therefore, we have $\rho_{ij} = \dfrac{1-\pi_{ij}}{\left(1-\pi_{ij}\right) + \pi_{ij}\left(1-\pi_{ij}\right)^{N_{ij}^{suc}}}$; $i = 1,2,....,n$ and $j = 1,2,....,m_i$

3.1.4 Reliability Equation of Both In-House and COTS Components

The reliability (r_{ij}) of j^{th} alternative of i^{th} module of the software

$$r_{ij} = \rho_{ij} y_{ij} + x_{ij} s_{ij};\ i = 1,2,....,n \text{ and } j = 1,2,....,m_i$$

3.1.5 Delivery Time Constraint

The maximum threshold T has been given on the delivery time of the whole system. In case of a COTS product the delivery time is simply given by d_{ij} , whereas for an in- house developed alternative the delivery time shall be expressed as

$$(t_{ij} + \tau_{ij} N_{ij}^{tot}):\qquad \sum_{i=1}^{n}\left(\sum_{j=1}^{m_i} y_{ij}\left(t_{ij} + \tau_{ij} N_{ij}^{tot}\right) + \sum_{j=1}^{m_i} d_{ij} x_{ij}\right) \le T$$

3.2 Objective Function

3.2.1 Reliability Objective Function

Reliability objective function maximizes the system quality (in terms of reliability) through a weighted function of module reliabilities. Reliability of modules that are invoked more frequently during use is given higher weights. Analytic Hierarchy Process (AHP) can be effectively used to calculate these weights.

$$\text{Maximize } R = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i$$

where R_i is the reliability of module i of the system under consensus recovery block scheme which is stated as follows:

$$R_i = 1 + \left[\sum_{j=1}^{m_i} \frac{1}{(1-r_{ij})^{z_{ij}}} \left[\prod_{k=1}^{m_i} (1-r_{ik})^{z_{ik}} \right] \left[1 - (1-r_{ij})^{z_{ij}} \right] + \prod_{j=1}^{m_i} (1-r_{ij})^{z_{ij}} \right] \left[\sum_{j=1}^{m_i} z_{ij} \left[\prod_{k=1}^{j-1} P(X_{ik})^{z_{ik}} \right] P(Y_{ij})^{z_{ij}} - 1 \right]; \; i = 1,2,.....n$$

3.2.2 Cost Objective Function

Cost objective function minimizes the overall cost of the system. The sum of the cost of all the modules is selected from "build – or - buy" strategy. The in-house development cost of the alternative j of module i can be expressed as $c_{ij}\left(t_{ij} + \tau_{ij} N_{ij}^{tot}\right)$:

$$\text{Minimize } \sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} c_{ij}\left(t_{ij} + \tau_{ij} N_{ij}^{tot}\right) y_{ij} + \sum_{j=1}^{m_i} C_{ij} x_{ij} \right)$$

3.3 Optimization Model I

In the optimization model it is assumed that the alternatives of a module are in a Consensus Recovery Block. Consensus Recovery Block achieving fault tolerance is to run all the attached independent alternatives simultaneously and selecting the output by the voting mechanism. It requires independent development of independent alternatives of a program, which the COTS components satisfy and a voting procedure. Upon invocation of the consensus recovery block all alternatives are executed and their output is submitted by a voting procedure. Since it is assumed that there is no common fault, if two or more alternatives agree on one output then that alternative is designated as correct. Otherwise the next stage is entered. At this stage the best version is examined by the acceptance test. If the output is accepted, it is treated as the correct one. However, if the output is not accepted, the next best version is subjected to testing. This process continues until an acceptable output is found or all outputs are exhausted.

$$\text{Maximize } R = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i \tag{1}$$

$$\text{Minimize } \sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} c_{ij}\left(t_{ij} + \tau_{ij} N_{ij}^{tot}\right) y_{ij} + \sum_{j=1}^{m_i} C_{ij} x_{ij} \right) \tag{2}$$

Subject to $X \in S = \{x_{ij}$ and y_{ij} are binary variable/

$$R_i = 1 + \left[\sum_{j=1}^{m_i} \frac{1}{(1-r_{ij})^{z_{ij}}} \left[\prod_{k=1}^{m_i} (1-r_{ik})^{z_{ik}} \right] \left[1-(1-r_{ij})^{z_{ij}} \right] + \prod_{j=1}^{m_i} (1-r_{ij})^{z_{ij}} \right] \left[\sum_{j=1}^{m_i} z_{ij} \left[\prod_{k=1}^{j-1} P(X_{ik})^{z_{ik}} \right] P(Y_{ij})^{z_{ij}} - 1 \right]; \; i = 1,2,.....n \tag{3}$$

$$\begin{matrix} P(X_{ij}) \\ P(Y_{ij}) \end{matrix} = \begin{matrix} (1-t_1) \\ r_{ij}(1-t_2) \end{matrix} \left[(1-r_{ij})(1-t_3) + r_{ij} t_2 \right] \quad \textbf{(P 1)}$$

$$N_{ij}^{suc} = (1-\pi_{ij}) N_{ij}^{tot} \quad , i = 1,2,....,n \text{ and } j = 1,2,....,m_i \tag{4}$$

$$\rho_{ij} = \frac{1-\pi_{ij}}{(1-\pi_{ij}) + \pi_{ij}(1-\pi_{ij})^{N_{ij}^{suc}}}; \quad i = 1,2,....,n \text{ and } j = 1,2,....,m_i \tag{5}$$

$$r_{ij} = \rho_{ij} y_{ij} + s_{ij} x_{ij}; \; i = 1,2,....,n \text{ and } j = 1,2,....,m_i \tag{6}$$

$$y_{ij} + x_{ij} \le 1; \; i = 1,2,....,n \text{ and } j = 1,2,....,m_i \tag{7}$$

$$y_{ij} + x_{ij} = z_{ij}; \quad i = 1,2,....,n \text{ and } j = 1,2,....,m_i \tag{8}$$

$$\sum_{j=1}^{m_i} z_{ij} \ge 1; \; i = 1,2,....,n \tag{9}$$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} y_{ij} \left(t_{ij} + \tau_{ij} N_{ij}^{tot} \right) + \sum_{j=1}^{m_i} d_{ij} x_{ij} \right) \le T \tag{10}$$

3.4 Optimization Model II

It is observed that some alternatives of a module may not be compatible with alternatives of another module. The next optimization model II addresses this problem. It is done, incorporating additional constraints in the optimization models. This constraint can be represented as $x_{gs} \le x_{hu_t}$, which means that if alternative s for module g is chosen, then alternative $u_t, t = 1,....., z$ have to be chosen for module h.

$$x_{gs} - x_{hu_t} \le M y_t \tag{11}$$

$$\sum_{t=1}^{z} y_t = z - 1 \quad \textbf{(P2)} \tag{12}$$

Constraint (11) and (12) make use of binary variable y_t to choose one pair of alternatives from among different alternative pairs of modules. If more than one alternative compatible component is to be chosen for redundancy, constraint (12) can be relaxed as

$$\sum_{t=1}^{z} y_t \leq z - 1 \tag{13}$$

Since optimization model II is an extension to model I, so above additional constraints of compatibility can be added to (P1) and hence problem (P2) is formed.

4 Fuzzy Multi-objective Optimization Model for Software Component Selection

Most of our traditional tools of modeling are crisp, deterministic, and precise in character. But for many practical problems there are incompleteness and unreliability of input information. This is caused to use fuzzy multi-objective optimization method with fuzzy parameters. Crisp mathematical programming approaches provide no such mechanism to quantify these uncertainties. Fuzzy optimization is a flexible approach that permits more adequate solutions of real problems in the presence of vague information, providing the well defined mechanisms to quantify the uncertainties directly. The idea of fuzzy programming was first given by Bellman and Zadeh [1] in 1970 and then developed by Tanaka [6] in 1974, Zimmermann [10] in the year 1976, and Wiedey & Zimmerman [9] 1979. Therefore, we formulate fuzzy multi-objective optimization model for software components selection.

4.1 Problem Solution

Following algorithm specifies the sequential steps to solve the fuzzy mathematical programming problems.

Step 1. Compute the crisp equivalent of the fuzzy parameters using a defuzzification function. Same defuzzification function is to be used for each of the parameters. Here we use the defuzzification function of the type where a^1, a^2, a^3 are the triangular fuzzy numbers. $F_2(A) = \left(a^1 + 2a^2 + a^3\right)/4$

Step 2. Incorporate the objective function of the fuzzifier min (max) as a fuzzy constraint with a restriction (aspiration) level. The above problem (P1) can be rewritten as

Find X

Subject to $R(X) = \sum_{l=1}^{L} f_l \prod_{i \in s_l} R_i \underset{\sim}{\geq} R_0$

$$C(X) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} c_{ij} \left(t_{ij} + \tau_{ij} N_{ij}^{tot} \right) y_{ij} + \sum_{j=1}^{m_i} C_{ij} x_{ij} \right) \underset{\sim}{\leq} C_0 \qquad \textbf{(P3)}$$

$X \in S$

where R_0 and C_0 are defizzified aspiration levels of system reliability and cost.

Step 3. Define appropriate membership functions for each fuzzy inequalities as well as constraint corresponding to the objective function. The membership function for the fuzzy less than or equal to and greater than or equal to type are given as

$$\mu_R(X) = \begin{cases} 1 & ; R(X) \geq R_0 \\ \dfrac{R(X) - R_0^*}{R_0 - R_0^*} & ; R_0^* \leq R(X) < R_0 \\ 0 & ; R(X) < R_0^* \end{cases}$$

where R_0 is the aspiration level and R_0^* is the tolerance levels to the fuzzy reliability objective function constraint.

$$\mu_C(X) = \begin{cases} 1 & ; C(X) \leq C_0 \\ \dfrac{C_0^* - C(X)}{C_0^* - C_0} & ; C_0 \leq C(X) < C_0^* \\ 0 & ; C(X) > C_0^* \end{cases}$$

where C_0 is the restiction and C_0^* is the tolerance levels to the fuzzy budget constraint.

Step 4. Employ extension principle to identify the fuzzy decision. While solving the problem its objective is treated as a constraint. Each constraint is considered to be an objective for the decision maker and the problem can be looked as a fuzzy multiple objective mathematical programming problem. Further each objective can have different level of importance and can be assigned weight to measure the relative importance. On substituting the values for $\mu_R(X)$ and $\mu_C(X)$ the problem becomes

Maximize α

Subject to

$$\begin{aligned} & R(X) \geq R_0 - (1 - w_1\alpha)(R_0 - R_0^*) \\ & C(X) \leq C_0 + (1 - w_2\alpha)(C_0^* - C_0) \\ & \alpha \in [0,1] \\ & X \in S \\ & w_1,\ w_2 \geq 0,\ w_1 + w_2 = 1 \end{aligned} \qquad \textbf{(P4)}$$

4.2 Illustrative Example

Consider a software system having three modules with more than one alternative for each module. The data sets for COTS and in-house developed components are given in Table-1 and Table 2, respectively.

Let $L = 3$, $s_1 = \{1,2,3\}$, $s_2 = \{1,3\}$, $s_3 = \{2\}$, $f_1 = 0.5$, $f_2 = 0.3$ and $f = 0.2$. It is also assumed that $t_1 = .01$, $t_2 = .05$ and $t_3 = .01$

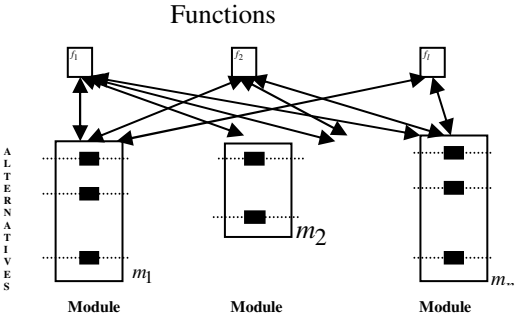


Fig. 1. Structure of Software

Table 1. Data Set for COTS Components

Module	COTS Alternative	Cost	Reliability	Delivery Time
	1	14	*0.89*	*2*
1	2	12.5	*0.90*	*3*
	3	15	*0.86*	*2*
2	1	13	*0.88*	*3*
	2	11	*0.98*	*4*
	3	20	*0.87*	*1*
	4	13	*0.88*	*4*
3	1	16	*0.87*	*2*
	2	16	0.91	2

Table 2. Data Set for In-House Components

Module	Alternatives	t_{ij}	τ_{ij}	c_{ij}	π_{ij}
	1	8	0.05	5	0.002
1	2	6	0.05	4	0.002
	3	7	0.05	4	0.002
2	1	9	0.05	5	0.002
	2	5	0.05	2	0.002
	3	6	0.05	4	0.002
	4	5	0.05	3	0.002
3	1	6	0.05	4	0.002
	2	5	0.05	3	0.002

4.2.1 Assignment of Weights

The assignment of weights is based on the expert's judgment for the reliability and cost. Weights assigned for reliability and cost are 0.6 and 0.4 respectively.

4.2.2 Minimum and Maximum level of Reliability and Cost

Firstly, the triangular fuzzy reliability, and cost values are computed using fuzzy values of these parameters and then defuzzfied using Heilpern' s defuzzfier. If the available reliability and cost are specified as TFN, then the aspiration and tolerance level can be given as follows:

Table 3. Aspiration and Tolerance level

Delivery Time	TFN	Aspiration Level	Tolerance Level
12	$R=(0.79,0.82,0.85)$ $C=(66,71,76)$	$R_0=0.82$ $C_0=71$	$R_0^*=0.78$ $C_0^*=76$
15	$R=(0.990,0.995,0.999)$ $C=(60,64,68)$	$R_0=0.995$ $C_0=64$	$R_0^*=0.97$ $C_0^*=75$
17	$R=(0.990,0.995,0.999)$ $C=(67,71,75)$	$R_0=0.995$ $C_0=71$	$R_0^*=0.97$ $C_0^*=75$

4.3 Optimization Model - I

The problem is solved using software package LINGO, Thiriez [7]. Following solution is obtained. It can be seen from the table that from delivery time 12 to delivery time 15, there is a significant reduction in the cost and increase in the reliability. Therefore, solution at delivery time 15 can be considered as the best decision.

Table 4. Solution of Optimization Model - I

Delivery Time	COTS Components	In-House Components	Optimal Cost	Optimal Reliability
12	$x_{11}, x_{12}, x_{22}, x_{23}, x_{32}$	-	73.5	0.80
15	$x_{12}, x_{13}, x_{24}, x_{32}$	y_{22}	66.5	0.992
17	x_{12}, x_{13}, x_{23}	y_{22}, y_{32}	72.5	0.994

4.4 Optimization Model – II

Since the solution at delivery time 15 is considered to be the best solution, therefore the compatibility condition is checked for the same. We assume third alternative of first module is compatible with first and third alternatives of second module.

Table 5. Solution of Optimization Model - II

Delivery Time	COTS Components	In-House Components	Optimal Cost	Optimal Reliability
15	$x_{12}, x_{13}, x_{23}, x_{32}$	y_{22}	73.5	0.992

It is observed that due to the compatibility condition, third alternative of second module is chosen as it is compatible with third alternative of first module.

5 Conclusion

This paper presents fuzzy multi – objective approach to optimal decision "build-or-buy" for component selection for a fault tolerant modular software system. The component selection problem is formulated as a multi-objective programming problem and fuzzy programming technique is used to provide a feasible solution. Optimization model I deals with the optimal selection of COTS components for a modular software system. In Optimization model II issue of compatibility amongst the alternatives of the modules is discussed. The model is illustrated with the numerical example and it is observed that sensitivity to the model has also been performed. We have observed that at shorter delivery times only COTS components were selected, as the delivery time increases the in- house components also get selected. The optimal cost and reliability values are given in the numerical example.

References

[1] Bellman, R.E., Zadeh, L.: Decision-making in a fuzzy environment. Management Science 17, B141–B164 (1970)

[2] Berman, O., Dinesh Kumar, U.: Optimization models for reliability of modular software system. IEEE Trans. Soft. Engg. 19/11, 1119–1123 (1993)

[3] Berman, O., Kumar, U.D.: Optimization models for complex recovery block schemes. Computers & Operations Research 26, 525–544 (1999)

[4] Bertolino, A., Strigini, L.: On the use of testability measures for dependability assessment. IEEE Transactions on Software Engg. 22/2, 97–108 (1996)

[5] Crnkovic, Larsson, M.: Building Reliable Component- Based Software Systems, pp. xxix–xxx. Artech House publisher (2002)

[6] Tanaka, H., Okuda, T., Asai, K.: On fuzzy mathematical programming. Journal of Cybernetics 3, 37–46 (1974)

[7] Thiriez, H.: OR software LINGO. European Journal of Operation Research 124, 655–656 (2000)

[8] Cortellessa, V., Marinelli, F., Potena, P.: An optimization framework for "build-or-buy" decisions in software architecture. Computers and Operations Research 35, 3090–3106 (2008)

[9] Wiedey, Zimmermann, H.S.: Media selection & fuzzy linear programming. J. Opl. Res. Soc. 29, 1071–1084 (1979)

[10] Zimmermann, H.J.: Description and optimization of fuzzy systems. International Journal of General Systems 2, 209–215 (1976)

A Solution Procedure for a Linear Fractional Programming Problem with Fuzzy Numbers

Mukesh Kumar Mehlawat, and Santosh Kumar

Department of Operational Research, University of Delhi, Delhi, India
mukesh0980@yahoo.com, santoshaor@gmail.com

Abstract. In this paper, we study a linear fractional programming problem in which the objective function coefficients, technological coefficients and the right-hand-side coefficients are fuzzy numbers. We present a vertex-following solution method using a linear ranking function. In fact, the proposed method is similar to the simplex method used for solving crisp linear fractional programming problems.

Keywords: Fuzzy numbers, linear fractional programming, ranking function, simplex method.

1 Introduction

Mathematical programming finds an extensive use in facilitating managerial decision situations in a large number of domains. An important class of mathematical programming problems is fractional programming which deals with situations where a ratio between physical and/or economical functions, is maximized (or minimized). There are certain decision situations that necessitate consideration of uncertainties in working environment best captured by fuzzy set theory. The concept of decision making in fuzzy environment was first proposed by Bellman and Zadeh [2]. Subsequently, Tanaka et al. [18] made use of this concept in mathematical programming. The use of fuzzy set theory concepts in fractional programming has been discussed by many authors, e.g. Chakraborty and Gupta [4], Dutta et al. [7], Gupta and Bhatia [8], Gupta and Mehlawat [9, 10], Luhandjula [11], Mehra et al. [13], Pop and Stancu-Minasian [14], Ravi and Reddy [15], Schaible and Shi [16], Stancu-Minasian and Pop [17].

Fuzzy fractional programming problem can be considered in different forms, that is, it is possible that some coefficients of the problem in the objective function, technological coefficients, the right-hand-side coefficients or decision variables are fuzzy numbers. In this paper, we concentrate on a linear fractional programming problem with fuzzy numbers in the objective function coefficients, technological coefficients and the right-hand-side coefficients termed as fuzzy number linear fractional programming problem (FNLFPP). We apply the concept of comparison of fuzzy numbers by using a linear ranking function from the literature. We describe

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1037–1049.
springerlink.com

basic feasible solution for the FNLFPP and establish optimality conditions. Using these optimality conditions, we propose a simplex type method for solving such problems.

This rest of the paper is organized as follows. In Section 2, we present some basic definitions and notations along with some preliminary results to be used in subsequent sections. In Section 3, we discuss fuzzy number linear fractional programming problem and present details of the solutions method to solve it. This section also devoted to a numerical illustration of the proposed solution method. In Section 4, we present an extension of the solution approach developed in Section 3 to the case of fuzzy variable linear fractional programming problem. Finally, some concluding remarks are made in Section 5.

2 Definitions and Notations

First, we review the fundamental notions of fuzzy set theory, initiated by Bellman and Zadeh [2], to be used in the subsequent sections (see also [6] or [17]).

Definition 1. Let X be the universal set. $\widetilde{A}$ is called a fuzzy set in X if $\widetilde{A}$ is a set of ordered pairs

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) \,|\, x \in X\},$$

where $\mu_{\widetilde{A}}(x)$ is the membership function of x in $\widetilde{A}$.

Remark 1. The membership of $\widetilde{A}$ specifies the degree of membership of element x in fuzzy set $\widetilde{A}$ (in fact, $\mu_{\widetilde{A}}$ shows the degree that x belongs to $\widetilde{A}$).

Definition 2. The α-level set of $\widetilde{A}$ is the set

$$\widetilde{A}_\alpha = \{x \in R \,|\, \mu_{\widetilde{A}}(x) \geq \alpha\},$$

where $\alpha \in (0,1]$. The lower and upper bounds of any α-level set $\widetilde{A}_\alpha$ are represented by finite numbers $\inf_{x \in \widetilde{A}_\alpha}$ and $\sup_{x \in \widetilde{A}_\alpha}$.

Definition 3. The support of a fuzzy set $\widetilde{A}$ is a set of elements in X for which $\mu_{\widetilde{A}}(x)$ is positive, that is, $\text{supp}\widetilde{A} = \{x \in X \,|\, \mu_{\widetilde{A}}(x) > 0\}$.

Definition 4. A fuzzy set $\widetilde{A}$ is convex if

$$\mu_{\widetilde{A}}(\lambda x + (1-\lambda)y) \geq \min\{\mu_{\widetilde{A}}(x), \mu_{\widetilde{A}}(y)\}, \quad \forall\, x \in X \text{ and } \lambda \in [0,1].$$

A fuzzy number $\widetilde{a}$ is a convex normalized fuzzy set on the real line R such that

1. there exist at least one $x_0 \in R$ with $\mu_{\widetilde{a}}(x_0) = 1$.
2. $\mu_{\widetilde{a}}(x_0)$ is piecewise continuous.

Among the various types of fuzzy numbers, triangular and trapezoidal fuzzy numbers are most commonly used.

Definition 5. Let $\widetilde{a} = (a^L, a^U, \alpha, \beta)$ denote the trapezoidal fuzzy number, where $[a^L - \alpha, a^U + \beta]$ is the support of $\widetilde{a}$ and $[a^L, a^U]$ its modal set. (See Figure 1.)

Remark 2. We denote the set of all trapezoidal fuzzy numbers by $F(R)$. If $a = a^L = a^U$ then we obtain a triangular fuzzy number, and we denote it by $\widetilde{a} = (a, \alpha, \beta)$. (See Figure 2.)

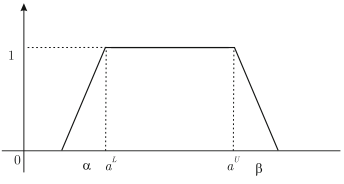


Fig. 1. Trapezoidal fuzzy number

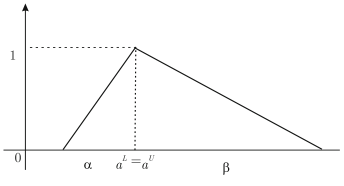


Fig. 2. Triangular fuzzy number

Next, we define arithmetic on fuzzy numbers. Let $\widetilde{a} = (a^L, a^U, \alpha, \beta)$ and $\widetilde{b} = (b^L, b^U, \gamma, \theta)$ be two trapezoidal fuzzy numbers. Let $\widetilde{0} = (0,0,0,0)$ denote the zero trapezoidal fuzzy number. Then

$$
\begin{aligned}
&x > 0,\ x \in R;\ x \cdot \widetilde{a} = (xa^L, xa^U, x\alpha, x\beta),\\
&x < 0,\ x \in R;\ x \cdot \widetilde{a} = (xa^U, xa^L, -x\beta, -x\alpha),\\
&\widetilde{a} + \widetilde{b} = (a^L + b^L, a^U + b^U, \alpha + \gamma, \beta + \theta),\\
&\widetilde{a} - \widetilde{b} = (a^L - b^U, a^U - b^L, \alpha - \theta, \beta - \gamma).
\end{aligned}
$$

We now consider ranking of fuzzy numbers. Many ranking methods can be found in literature. Bass and Kwakernaak [1] are among the pioneers in this area. Bortolan and Degani [3], Wang and Kerre [19] have reviewed different ordering methods. One can refer to Chen and Hwang [5] for the description of various ordering methods.

Define a ranking function $\Re : F(R) \to R$ which maps each fuzzy number into the real line, where a natural order exists.

We define orders on $F(R)$ by

$$\widetilde{a} \underset{\Re}{\geq} \widetilde{b} \text{ if and only if } \Re(\widetilde{a}) \geq \Re(\widetilde{b}), \tag{1}$$

$$\widetilde{a} \underset{\Re}{>} \widetilde{b} \text{ if and only if } \Re(\widetilde{a}) > \Re(\widetilde{b}), \tag{2}$$

$$\widetilde{a} \underset{\Re}{=} \widetilde{b} \text{ if and only if } \Re(\widetilde{a}) = \Re(\widetilde{b}), \tag{3}$$

where $\widetilde{a}$ and $\widetilde{b}$ are in $F(R)$. Also we write $\widetilde{a} \underset{\Re}{\leq} \widetilde{b}$ if and only if $\widetilde{b} \underset{\Re}{\geq} \widetilde{a}$.

The following lemma is now immediate.

Lemma 1. *Let $\Re$ be any linear ranking function. Then*

1. $\widetilde{a} \underset{\Re}{\geq} \widetilde{b}$ *iff* $\widetilde{a} - \widetilde{b} \underset{\Re}{\geq} 0$ *iff* $-\widetilde{b} \underset{\Re}{\geq} -\widetilde{a}$.
2. *if* $\widetilde{a} \underset{\Re}{\geq} \widetilde{b}$ *and* $\widetilde{c} \underset{\Re}{\geq} \widetilde{d}$, *then* $\widetilde{a} + \widetilde{c} \underset{\Re}{\geq} \widetilde{b} + \widetilde{d}$.

We restrict our attention to linear ranking functions, that is, a ranking function $\Re$ such that

$$\Re(k\widetilde{a} + \widetilde{b}) = k\Re(\widetilde{a}) + \Re(b) \tag{4}$$

for any $\widetilde{a}$ and $\widetilde{b}$ belonging to $F(R)$ and any $k \in R$.

In this paper, we use the following criteria for comparison of trapezoidal fuzzy numbers by using a linear ranking function discussed in [5]

$$\Re(\widetilde{a}) = \int_0^1 (\inf \widetilde{a}_\alpha + \sup \widetilde{a}_\alpha) d\alpha. \tag{5}$$

which reduces to

$$\Re(\widetilde{a}) = a^L + a^U + \frac{1}{2}(\beta - \alpha). \tag{6}$$

For trapezoidal fuzzy numbers $\widetilde{a}$ and $\widetilde{b}$, we have

$$\widetilde{a} \underset{\Re}{\geq} \widetilde{b} \text{ if and only if } \; a^L + a^U + \frac{1}{2}(\beta - \alpha) \geq b^L + b^U + \frac{1}{2}(\theta - \gamma). \tag{7}$$

3 Fuzzy Number Linear Fractional Programming

A fuzzy number linear fractional programming problem (FNLFPP) is stated as under

$$\begin{aligned} \max \widetilde{z} &\underset{\Re}{=} \frac{\widetilde{c}^t x + \widetilde{c}_0}{\widetilde{d}^t x + \widetilde{d}_0} \\ \textit{subject to} & \\ & \widetilde{A}x \underset{\Re}{=} \widetilde{b} \\ & x \geq 0 \end{aligned}$$

where '$\underset{\Re}{=}$' mean equality with respect to the ranking function $\Re$, $\widetilde{A} = (\widetilde{a}_{ij})_{m\times n}$, $\widetilde{c} = (\widetilde{c}_1, \widetilde{c}_2, \ldots, \widetilde{c}_n)^t$, $\widetilde{d} = (\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_n)^t$, $\widetilde{b} = (\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_m)^t$, $x = (x_1, x_2, \ldots, x_n)^t$, $\widetilde{c}_0, \widetilde{d}_0$ are scalars and $\widetilde{a}_{ij}$, $\widetilde{b}_i$, $\widetilde{c}_j$, $\widetilde{c}_0, \widetilde{d}_0 \in F(R)$ for $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$.

Definition 6. Any x which satisfies the set of constraints of FNLFPP is called a feasible solution. Let S be the set of all feasible solutions of FNLFPP. We say $x^* \in S$ is an optimal feasible solution for FNLFPP if $\dfrac{\widetilde{c}^t x^* + \widetilde{c}_0}{\widetilde{d}^t x^* + \widetilde{d}_0} \underset{\Re}{\geq} \dfrac{\widetilde{c}^t x + \widetilde{c}_0}{\widetilde{d}^t x + \widetilde{d}_0}$ for all $x \in S$.

Definition 7. We say that the real number **a** corresponds to the fuzzy number $\widetilde{a}$, with respect to a given linear ranking function $\Re$, if $\mathbf{a} = \Re(\widetilde{a})$.

3.1 Solution Method for FNLFPP

We rewrite FNLFPP as

$$\max \widetilde{z} = \frac{\widetilde{c}^t x + \widetilde{c}_0}{\widetilde{d}^t x + \widetilde{d}_0} = \frac{(\widetilde{c}_B^t x_B + \widetilde{c}_N^t x_N) + \widetilde{\alpha}}{(\widetilde{d}_B^t x_B + \widetilde{d}_N^t x_N) + \widetilde{\beta}}$$

subject to

$$\widetilde{A}x \underset{\Re}{=} \widetilde{b},$$
$$x \geq 0,$$

with the additional assumption that the $\widetilde{d}^t x + \widetilde{d}_0 \underset{\Re}{>} \widetilde{0}$ for all feasible solutions and the domain of feasible solutions $S = \{x \,|\, \widetilde{A}x \underset{\Re}{=} \widetilde{b}, x \geq 0\}$ represents a regular set.

• **Fuzzy basic feasible solution**

Consider the system $\widetilde{A}x \underset{\Re}{=} \widetilde{b}$ and $x \geq 0$. Let $A = [a_{ij}]_{m\times n} = [\Re(\widetilde{a}_{ij})] = \Re(\widetilde{A})$. Assume $\text{rank}(A) = m$. Partition A as $[B,N]$ where B, $m \times m$, is nonsingular. It is obvious that $\text{rank}(B) = m$ and $B = \Re(\widetilde{B})$, where $\widetilde{B}$ is the fuzzy matrix in $\widetilde{A}$ corresponding to B. The solution $x = (x_B^t, x_N^t)^t$ where

$$x_B = (x_{B_1}, x_{B_2}, \ldots, x_{B_m})^t = B^{-1}b, \quad x_N = 0$$

is a basic solution of $Ax = b$, where $b = \Re(\widetilde{b})$. If $x_B \geq 0$, then x is called a BFS of FNLFPP. Here B is called the basis matrix and N is called the non-basis matrix. The components of x_B are called basic variables, and the components of x_N are called non basic variables. If $x_B > 0$, then x is called a non degenerate BFS, and if atleast one component of x_B is zero, then x is called a degenerate BFS.

Let x_B be the initial BFS such that

$$Bx_B = b \text{ or } x_B = B^{-1}b, \quad x_B \geq 0,\ x_N = 0$$

where $b = \Re(\widetilde{b})$, $B = (\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_m) = (\Re(\widetilde{b}_1), \Re(\widetilde{b}_2), \ldots, \Re(\widetilde{b}_m))$. Further let

$$\widetilde{z}^1 \underset{\Re}{=} \widetilde{c}_B^t x_B + \widetilde{c}_0 \quad \textit{and} \quad \widetilde{z}^2 \underset{\Re}{=} \widetilde{d}_B^t x_B + \widetilde{d}_0$$

where $\widetilde{c}_B^t$ and $\widetilde{d}_B^t$ are the vectors having their components as the fuzzy numbers associated with the basic variables in the numerator and the denominator of the objective function respectively. In addition, we assume that for this BFS.

$$y_j = B^{-1}a_j, \quad \widetilde{z}_j^1 = \widetilde{c}_B^t y_j \quad and \quad \widetilde{z}_j^2 = \widetilde{d}_B^t y_j$$

are known for every column a_j of A not in B.

We now wish to examine the possibility of finding another BFS with improved value of $\widetilde{z} = \frac{\widetilde{z}^1}{\widetilde{z}^2}$. We shall confine our attention to those basic feasible solutions in which only one column of B is changed. Now if the new basic feasible solution is denoted by $\widehat{x}_B$, then $\widehat{x}_B = \widehat{B}^{-1}b$ where $\widehat{B} = (\widehat{\mathsf{b}}_1, \widehat{\mathsf{b}}_2, \ldots, \widehat{\mathsf{b}}_m)$ i.e. a new nonsingular matrix obtained from B by removing one of its column, say, b_r, and replacing it by a_j. The column of the new matrix $\widehat{B}$ are given by

$$\widehat{\mathsf{b}}_i = \widetilde{b}_i (i \neq r) \quad and \quad \widehat{\mathsf{b}}_r = a_j .$$

We obtain the value of the new basic variables in terms of the original ones and the y_{ij}, as

$$\widehat{x}_{B_i} = x_{B_i} - x_{B_r}(y_{ij}/y_{rj}), \qquad (i \neq r),$$
$$\widehat{x}_{B_r} = x_{B_r}/y_{rj} = \theta \ \ (say)$$

where $a_j = \sum_{i=1}^{m} y_{ij} b_i$. Since we are interested in finding a new BFS with an improved value of the objective function, we must determine whether '$\widetilde{z}$' is improved or not. The value of the objective function for the original BFS is $\widetilde{z} = \frac{\widetilde{z}_1}{\widetilde{z}^2}$. Let the new value of the objective function be $\widetilde{\bar{z}} = \frac{\widetilde{\bar{z}}^1}{\widetilde{\bar{z}}^2}$.

Now we have

$$\widetilde{\bar{z}}^1 = \widetilde{z}^1 + \theta(\widetilde{c}_j - \widetilde{z}_j^1) \quad and \quad \widetilde{\bar{z}}^2 = \widetilde{z}^2 + \theta(\widetilde{d}_j - \widetilde{z}_j^2)$$

where $\widetilde{z}_j^1$ and $\widetilde{z}_j^2$ corresponds to the original BFS. The value of the objective function will improve if

$$\widetilde{\bar{z}} \gtrsim \widetilde{z}, \quad or \quad \frac{\widetilde{z}^1 + \theta(\widetilde{c}_j - \widetilde{z}_j^1)}{\widetilde{z}^2 + \theta(\widetilde{d}_j - \widetilde{z}_j^2)} \gtrsim \frac{\widetilde{z}^1}{\widetilde{z}^2}$$

or

$$\frac{\widetilde{z}^1 + \theta(\widetilde{c}_j - \widetilde{z}_j^1)}{\widetilde{z}^2 + \theta(\widetilde{d}_j - \widetilde{z}_j^2)} - \frac{\widetilde{z}_1}{\widetilde{z}^2} \gtrsim \widetilde{0},$$
$$\widetilde{z}^2(\widetilde{z}^1 + \theta(\widetilde{c}_j - \widetilde{z}_j^1)) - \widetilde{z}^1(\widetilde{z}^2 + \theta(\widetilde{d}_j - \widetilde{z}_j^2)) \gtrsim \widetilde{0},$$

($\widetilde{z}^2, \widetilde{\bar{z}}^2$ are positive, since the denominator of the objective function is positive for all feasible solutions)

or

$$\widetilde{z}^2(\widetilde{c}_j - \widetilde{z}_j^1) - \widetilde{z}^1(\widetilde{d}_j - \widetilde{z}_j^2) \gtrsim \widetilde{0}$$

(θ being positive in the non degenerate cases; if $\theta = 0\ \widetilde{\bar{z}} \underset{\Re}{=} \widetilde{z}$).

Let $\widetilde{\Delta}_j \underset{\Re}{=} \widetilde{z}^2(\widetilde{z}_j^1 - \widetilde{c}_j) - \widetilde{z}^1(\widetilde{z}_j^2 - \widetilde{d}_j) \underset{\Re}{\leq} \widetilde{0}$.

Now $\widetilde{\Delta}_j$ is less than zero if

Case 1. $\widetilde{z}_j^2 - \widetilde{d}_j \underset{\Re}{>} \widetilde{0}$ and $(\widetilde{z}_j^1 - \widetilde{c}_j)/(\widetilde{z}_j^2 - \widetilde{d}_j) \underset{\Re}{<} \widetilde{z}^1/\widetilde{z}^2$.

Case 2. $\widetilde{z}_j^2 - \widetilde{d}_j \underset{\Re}{<} \widetilde{0}$ and $(\widetilde{z}_j^1 - \widetilde{c}_j)/(\widetilde{z}_j^2 - \widetilde{d}_j) \underset{\Re}{>} \widetilde{z}^1/\widetilde{z}^2$.

Case 3. $\widetilde{z}_j^2 - \widetilde{d}_j \underset{\Re}{=} \widetilde{0}$ and $(\widetilde{z}_j^1 - \widetilde{c}_j) \underset{\Re}{=} \widetilde{0}$.

We conclude that given a BFS $x_B = B^{-1}b$, if for any column a_j in A but not in B, $\widetilde{\Delta}_j \underset{\Re}{<} \widetilde{0}$ holds and if atleast one $y_{ij} > 0$ $(i = 1,2,\ldots,m)$ then it is possible to obtain a new BFS by replacing one of the column in B by a_j and the new value of the objective function satisfies $\widetilde{\bar{z}} \underset{\Re}{\geq} \widetilde{z}$. But if $y_{ij} \leq 0$, then the optimal objective is unbounded.

The solution method described as above is summarized in the form of the following theorems. Proof of these theorems follows on similar lines as in [12] and are hence omitted.

Theorem 1. *If in a simplex table, an ℓ exists such that $\widetilde{\Delta}_\ell \underset{\Re}{=} \widetilde{z}^2(\widetilde{z}_\ell^1 - \widetilde{c}_\ell) - \widetilde{z}^1(\widetilde{z}_\ell^2 - \widetilde{d}_\ell) \underset{\Re}{<} \widetilde{0}$ and there exists a basic index i such that $y_{i\ell} > 0$, then a pivoting row r can be found so that pivoting on $y_{r\ell}$ will yield a BFS with a corresponding non decreasing fuzzy objective value.*

Theorem 2. *Assume that FNLFPP is non degenerate. A BFS $x_B = B^{-1}b$, $x_N = 0$ is an optimal solution to FNLFPP if and only if $\widetilde{\Delta}_j \underset{\Re}{\geq} \widetilde{0}$ for all $1 \leq j \leq n$.*

Theorem 3. *If for any BFS to the FNLFPP there is some column, say, ℓ not in basis for which $\widetilde{\Delta}_\ell \underset{\Re}{<} \widetilde{0}$ and $y_{i\ell} \leq 0$, $i = 1,2,\ldots,m$, then the FNLFPP has an unbounded solution.*

• Step-wise algorithm to solve the FNLFPP

A detailed step-wise algorithm is presented here to solve the FNLFPP under the assumption that currently we have a fuzzy basic feasible solution with basis B.

1. The basic feasible solution is given by $x_B = B^{-1}b$ and $x_N = 0$. The fuzzy objective value is $\widetilde{z} \underset{\Re}{=} \dfrac{\widetilde{z}^1}{\widetilde{z}^2}$.
2. Calculate $\widetilde{\Delta}_j \underset{\Re}{=} \widetilde{z}^2(\widetilde{z}_j^1 - \widetilde{c}_j) - \widetilde{z}^1(\widetilde{z}_j^2 - \widetilde{d}_j), 1 \leq j \leq n$. If $\widetilde{\Delta}_j \underset{\Re}{\geq} \widetilde{0}$ for all $1 \leq j \leq n$, then stop. The current solution is optimal. But if $\widetilde{\Delta}_j \underset{\Re}{<} \widetilde{0}$ for at least one j, then an ℓ exists such that $\widetilde{\Delta}_\ell \underset{\Re}{=} \min\limits_{j=1,\ldots,n} \widetilde{\Delta}_j$ and go to step 3.

3. If $y_{i\ell} \leq 0$, then stop. The problem is unbounded. Otherwise, find an index r corresponding to a basic variable x_{B_r}, leaving the basis as follows:
$$\frac{x_{B_r}}{y_{r\ell}} = \min_{i=1,\ldots,m} \{\frac{x_{B_i}}{y_{i\ell}} : y_{i\ell} > 0\}.$$
4. Pivot on $y_{r\ell}$ and update the simplex tableau. Go to step 2.

3.2 Numerical Illustration

In this section, we present an illustration of the proposed method. We solve an FNLFPP using the proposed solution method.

$$\max \widetilde{z} = \frac{(5,8,2,5)x_1 + (3,5,1,2)x_2}{(5,8,2,5)x_1 + (2,4,1,2)x_2 + (1,2,1,1)}$$
$$\textit{subject to}$$
$$(1,2,1,1)x_1 + (2,3,1,1)x_2 \underset{\Re}{\leq} (5,8,2,6)$$
$$(2,3,1,1)x_1 + (1,1,1,1)x_2 \underset{\Re}{\leq} (4,5,1,3)$$
$$x_1, x_2 \geq 0$$

It may be noted that each fuzzy number used in the above problem is a trapezoidal fuzzy number. Applying the ranking function (6), we can write the constraints of the FNLFPP in the standard form as under:

$$3x_1 + 5x_2 + x_3 = 15$$
$$5x_1 + 2x_2 + x_4 = 10$$
$$x_1, x_2, x_3, x_4 \geq 0$$

where x_3 and x_4 are slack variables. The first feasible simplex table is as follows:

Table 1. Simplex method iteration-I

$\widetilde{c}_j \rightarrow$			$(5,8,2,5)$	$(3,5,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_j \rightarrow$			$(5,8,2,5)$	$(2,4,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_B$	$\widetilde{c}_B$	x_B	x_1	x_2	x_3	x_4
$\widetilde{0}$	$\widetilde{0}$	$x_3 = 15$	3	5	1	0
$\widetilde{0}$	$\widetilde{0}$	$x_4 = 10$	5	2	0	1
$z^2 = \Re(\widetilde{z}^2) = 3$	$z^1 = \Re(\widetilde{z}^1) = 0$	$z = \Re(\widetilde{z}) = 0$				
		$\widetilde{z}_j^1 - \widetilde{c}_j$	$(-8,-5,5,2)$	$(-5,-3,2,1)$	$\widetilde{0}$	$\widetilde{0}$
		$\widetilde{z}_j^2 - \widetilde{d}_j$	$(-8,-5,5,2)$	$(-4,2,2,1)$	$\widetilde{0}$	$\widetilde{0}$
$\Delta_j = \Re(\widetilde{\Delta}_j)$			-43.5	-25.5	0	0

Since $(\Delta_1, \Delta_2) = (R(\widetilde{\Delta}_1), R(\widetilde{\Delta}_2)) = (-43.5, -25.5)$, then x_1 enters the basis and the leaving variable is x_4. The new table is

Table 2. Simplex method iteration-II

$\widetilde{c}_j \rightarrow$			$(5,8,2,5)$	$(3,5,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_j \rightarrow$			$(5,8,2,5)$	$(2,4,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_B$	$\widetilde{c}_B$	x_B	x_1	x_2	x_3	x_4
$\widetilde{0}$	$\widetilde{0}$	$x_3 = 9$	0	$\frac{19}{5}$	1	$-\frac{3}{5}$
$(5,8,2,5)$	$(5,8,2,5)$	$x_1 = 2$	1	$\frac{2}{5}$	0	$\frac{1}{5}$
$z^2 = \Re(\widetilde{z}^2) = 32$	$z^1 = \Re(\widetilde{z}^1) = 29$	$z = \Re(\widetilde{z}) = \frac{29}{32}$				
		$\widetilde{z}_j^1 - \widetilde{c}_j$	(-3,3,7,7)	$\left(-3, \frac{1}{5}, \frac{14}{5}, 3\right)$	$\widetilde{0}$	$\left(1, \frac{8}{5}, \frac{2}{5}, 1\right)$
		$\widetilde{z}_j^2 - \widetilde{d}_j$	(-3,3,7,7)	$\left(-2, \frac{6}{5}, \frac{14}{5}, 3\right)$	$\widetilde{0}$	$\left(1, \frac{8}{5}, \frac{2}{5}, 1\right)$
$\Delta_j = \Re(\widetilde{\Delta}_j)$			0	$\frac{-661}{10}$	0	$\frac{87}{10}$

Now, from $(\Delta_2, \Delta_4) = (R(\widetilde{\Delta}_2), R(\widetilde{\Delta}_4)) = (-66.1, 8.7)$, then x_2 enters the basis and the leaving variable is x_3. The new table is

Table 3. Simplex method iteration-III

$\widetilde{c}_j \rightarrow$			$(5,8,2,5)$	$(3,5,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_j \rightarrow$			$(5,8,2,5)$	$(2,4,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_B$	$\widetilde{c}_B$	x_B	x_1	x_2	x_3	x_4
$(2,4,1,2)$	$(3,5,1,2)$	$x_2 = \frac{45}{19}$	0	1	$\frac{5}{19}$	$\frac{-3}{19}$
$(5,8,2,5)$	$(5,8,2,5)$	$x_1 = \frac{20}{19}$	1	0	$\frac{-2}{19}$	$\frac{5}{19}$
$z^2 = \Re(\widetilde{z}^2) = \frac{1279}{38}$	$z^1 = \Re(\widetilde{z}^1) = \frac{1345}{38}$	$z = \Re(\widetilde{z}) = \frac{1345}{1279}$				
		$\widetilde{z}_j^1 - \widetilde{c}_j$	(-3,3,7,7)	(-2,2,3,3)	$\left(\frac{-1}{19}, \frac{15}{19}, \frac{15}{19}, \frac{14}{19}\right)$	$\left(\frac{10}{19}, \frac{31}{19}, \frac{16}{19}, \frac{28}{19}\right)$
		$\widetilde{z}_j^2 - \widetilde{d}_j$	(-3,3,7,7)	(-2,2,3,3)	$\left(\frac{-6}{19}, \frac{10}{19}, \frac{15}{19}, \frac{14}{19}\right)$	$\left(\frac{13}{19}, \frac{34}{19}, \frac{16}{19}, \frac{28}{19}\right)$
$\Delta_j = \Re(\widetilde{\Delta}_j)$			0	0	$\frac{25118}{1442}$	$\frac{-22344}{1442}$

Since, $(\Delta_3,\Delta_4)=(R(\widetilde{\Delta}_3),R(\widetilde{\Delta}_4))=\left(\frac{25118}{1444},\frac{-22344}{1444}\right)$, therefore, x_4 enters the basis and the leaving variable is x_1. The new table is

Table 4. Simplex method iteration-IV

$\widetilde{c}_j\rightarrow$			$(5,8,2,5)$	$(3,5,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_j\rightarrow$			$(5,8,2,5)$	$(2,4,1,2)$	$\widetilde{0}$	$\widetilde{0}$
$\widetilde{d}_B$	$\widetilde{c}_B$	x_B	x_1	x_2	x_3	x_4
$(2,4,1,2)$	$(3,5,1,2)$	$x_2=3$	$\frac{3}{5}$	1	$\frac{1}{5}$	0
$\widetilde{0}$	$\widetilde{0}$	$x_4=4$	$\frac{19}{5}$	0	$\frac{-2}{5}$	1
$z^2=\Re(\widetilde{z}^2)$ $=\frac{45}{2}$	$z^1=\Re(\widetilde{z}^1)$ $=\frac{51}{2}$	$z=\Re(\widetilde{z})$ $=\frac{51}{45}$				
		$\widetilde{z}_j^1-\widetilde{c}_j$	$\left(\frac{-31}{5},-2,\frac{28}{5},\frac{16}{5}\right)$	$(-2,2,3,3)$	$\left(\frac{3}{5},1,\frac{1}{5},\frac{2}{5}\right)$	$\widetilde{0}$
		$\widetilde{z}_j^2-\widetilde{d}_j$	$\left(\frac{-34}{5},\frac{-13}{5},\frac{28}{5},\frac{16}{5}\right)$	$(-2,2,3,3)$	$\left(\frac{2}{5},\frac{4}{5},\frac{1}{5},\frac{2}{5}\right)$	$\widetilde{0}$
$\Delta_j=\Re(\widetilde{\Delta}_j)$			$\frac{588}{10}$	0	$\frac{51}{10}$	0

Now, $(\Delta_2,\Delta_4)=(R(\widetilde{\Delta}_2),R(\widetilde{\Delta}_4))=\left(\frac{633}{10},\frac{51}{10}\right)>0$, and $\widetilde{\Delta}_3\underset{\Re}{=}\widetilde{\Delta}_1\underset{\Re}{=}\widetilde{0}$.

The optimality condition of Theorem 2 is satisfied. Therefore the current basis is optimal. The optimal solution is $x_1^*=0$, $x_2^*=3$ and $z^*=\Re(\widetilde{z}^*)=\frac{25.5}{22.5}=1.133$.

4 Fuzzy Variable Linear Fractional Programming Problem

A fuzzy variable linear fractional programming problem (FVLFPP) is defined as follows:

$$\begin{aligned}&\max\widetilde{z}\underset{\Re}{=}\frac{c^t\widetilde{x}+c_0}{d^t\widetilde{x}+d_0}\\&subject\ to\\&\quad A\widetilde{x}\underset{\Re}{=}\widetilde{b},\\&\quad\widetilde{x}\underset{\Re}{\geq}\widetilde{0},\end{aligned}$$

where $\widetilde{b}\in(F(R))^m$, $A\in R^{m\times n}$, $c,d\in R^n$, c_0,d_0 are scalars, and $\widetilde{x}\in(F(R))^n$.

Definition 8. We say that fuzzy vector $\widetilde{x} \in (F(R))^n$ is a fuzzy feasible solution to FVLFPP if $\widetilde{x}$ satisfies constraints of the problem. Let S be the set of all fuzzy feasible solutions of FVLFPP. We say $\widetilde{x}^* \in S$ is a fuzzy optimal solution for FVLFPP if

$$\frac{c^t\widetilde{x}^* + c_0}{d^t\widetilde{x}^* + d_0} \underset{\Re}{\geq} \frac{c^t\widetilde{x} + c_0}{d^t\widetilde{x} + d_0} \quad for\ all\ x \in S.$$

Let $A = [a_{ij}]_{m\times n}$. Assume $\text{rank}(A) = m$. Partition A as $[B\ N]$ where B, $m \times m$, is nonsingular. It is obvious that $\text{rank}(B) = m$. Let y_j be the solution to $By = a_j$. It is apparent that the basic solution

$$\widetilde{x}_B = (\widetilde{x}_{B_1}, \widetilde{x}_{B_2}, \ldots, \widetilde{x}_{B_m})^t \underset{\Re}{=} B^{-1}\widetilde{b}, \quad \widetilde{x}_N \underset{\Re}{=} \widetilde{0}$$

is a solution of $A\widetilde{x} \underset{\Re}{=} \widetilde{b}$. We call $\widetilde{x}$, accordingly partitioned as $(\widetilde{x}_B^t\ \widetilde{x}_N^t)^t$, a fuzzy basic solution corresponding to basis B. If $\widetilde{x}_B \underset{\Re}{\geq} \widetilde{0}$, then $\widetilde{x}$ is called fuzzy basic feasible solution (FBFS) of the system. If $x_B \underset{\Re}{>} \widetilde{0}$ then $\widetilde{x}$ is called a non-degenerate FBFS, and if atleast one component of $\widetilde{x}_B$ is zero, then $\widetilde{x}$ is called a degenerate FBFS.

Theorem 4. *Assume that the FVLFPP is non-degenerate. A fuzzy basic feasible solution* $\widetilde{x}_B \underset{\Re}{=} B^{-1}\widetilde{b}$, $\widetilde{x}_N \underset{\Re}{=} \widetilde{0}$ *is optimal to FVLFPP if and only if* $\widetilde{\Delta}_j \underset{\Re}{\geq} \widetilde{0}$ *for all* $j, 1 \leq j \leq n$.

• Step-wise algorithm to solve the FVLFPP

A detailed step-wise algorithm is presented here to solve the FVLFPP under the assumption that currently we have a fuzzy basic feasible solution with basis B.

1. The basic feasible solution is given by $\widetilde{x}_B = B^{-1}b$ and $\widetilde{x}_N = \widetilde{0}$. The fuzzy objective value is $\widetilde{z} \underset{\Re}{=} \dfrac{\widetilde{z}^1}{\widetilde{z}^2}$.
2. Calculate $\widetilde{\Delta}_j \underset{\Re}{=} \widetilde{z}^2(\widetilde{z}_j^1 - \widetilde{c}_j) - \widetilde{z}^1(\widetilde{z}_j^2 - \widetilde{d}_j), 1 \leq j \leq n$. If $\widetilde{\Delta}_j \underset{\Re}{\geq} \widetilde{0}$ for all $1 \leq j \leq n$, then stop. The current solution is optimal. But if $\widetilde{\Delta}_j \underset{\Re}{<} \widetilde{0}$ for at least one j, then an ℓ exists such that $\widetilde{\Delta}_\ell \underset{\Re}{=} \min\limits_{j=1,\ldots,n} \widetilde{\Delta}_j$ and go to step 3.
3. If $y_{i\ell} \leq 0$, then stop. The problem is unbounded. Otherwise, find an index r corresponding to a basic variable $\widetilde{x}_{B_r}$, leaving the basis as follows:
$$\frac{x_{B_r}}{y_{r\ell}} = \min_{i=1,\ldots,m} \{\frac{x_{B_i}}{y_{i\ell}} : y_{i\ell} > 0\},$$
where $x_{B_i} = \Re(\widetilde{x}_{B_i})$.
4. Pivot on $y_{r\ell}$ and update the simplex tableau. Go to step 2.

5 Concluding Remarks

We considered a linear fractional programming problem in which the objective function coefficients, technological coefficients and the right-hand-side coefficients are

fuzzy numbers and introduced the concept of fuzzy basic feasible solution for the problem. Applying the optimality conditions for the non-degenerate problem, we proposed a new algorithm using a linear ranking function for comparison of fuzzy numbers. The proposed solution method is similar to the simplex method for solving crisp linear fractional programming problem. The presented solution method allows the use of linear fuzzy sets, however, for nonlinear membership functions it is necessary to use numerical methods to obtain linear approximations.

References

1. Bass, S.M., Kwakernaak, H.: Rating and ranking of multiple aspent alternative using fuzzy sets. Automatica 13, 47–58 (1977)
2. Bellman, R.E., Zadeh, R.E.: Decision making in a fuzzy environment. Manag. Sci. 17, 141–164 (1970)
3. Bortolan, G., Degani, R.: A review of some methods for ranking fuzzy subsets. Fuzzy Set. Syst. 15, 1–19 (1985)
4. Chakraborty, M., Gupta, S.: Fuzzy mathematical programming for multi-objective linear ractional programming problem. Fuzzy Set. Syst. 125, 335–342 (2002)
5. Chen, S.J., Hwang, C.L.: Fuzzy Multiple Attribute Decision Making, Methods and Applications. Springer, Berlin (1992)
6. Dubios, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
7. Dutta, D., Tiwari, R.N., Rao, J.R.: Multiple objective linear fractional programming problem-a fuzzy set theoretic approach. Fuzzy Set. Syst. 52, 39–45 (1992)
8. Gupta, P., Bhatia, D.: Sensitivity analysis in fuzzy multiobjective linear fractional programming problem. Fuzzy Set. Syst. 122, 229–236 (2001)
9. Gupta, P., Mehlawat, M.K.: Duality for a convex fractional programming under fuzzy environment. International Journal of Optimization: Theory, Methods and Applications 3, 291–301 (2009)
10. Gupta, P., Mehlawat, M.K.: Duality in fuzzy linear fractional programming. International Journal of Optimization: Theory, Methods and Applications 3, 21–44 (2011)
11. Luhandjula, M.K.: Fuzzy approaches for multiple objective linear fractional optimization. Fuzzy Set. Syst. 13, 11–23 (1984)
12. Madhavi-Amiri, N., Nasseri, S.H.: Duality in fuzzy number linear programming by use of a certain linear ranking function. Appl. Math. Comput. 180, 206–216 (2006)
13. Mehra, A., Chandra, S., Bector, C.R.: Acceptable optimality in linear fractional programming with fuzzy coefficients. Fuzzy Optim. Decis. Making 6, 5–16 (2007)
14. Pop, B., Stancu-Minasian, I.M.: A method of solving fully fuzzified linear fractional programming problem. J. Appl. Math. Comput. 27, 227–242 (2008)
15. Ravi, V., Reddy, P.J.: Fuzzy linear fractional goal programming applied to reffinery operations planning. Fuzzy Set. Syst. 96, 173–182 (1998)
16. Schaible, S., Shi, J.: Recent developments in fractinal programming: single ratio and max-min case. In: Takahasi, W., Tanoka, T. (eds.) Proceedings of the 3rd International Conference on Nonlinear and Convex Analysis, August 25-29, Yokohama Publishing, Tokyo (2003)

17. Stancu-Minasian, I.M., Pop, B.: Fuzzy fractional programmin some recent results. Paper Presented at 2nd International Conference on Symmetry and Antisymmetry in Mathematics, Formal Languages and Computer Science, BraUsov, Romania, June 29-July 1 (2000)
18. Tanaka, H., Okuda, T., Asai, K.: On fuzzy mathematical programming. J. Cybern. 3, 37–46 (1984)
19. Wang, X., Kerre, E.: On the classification and the dependencies of the ordering methods. In: Ruan, D. (ed.) Fuzzy Logic Foundations and Industrial Applications, pp. 73–90. Kluwer (1996)

A Neural Network Approach to Distinguish Parkinsonian Tremor from Advanced Essential Tremor

A. Hossen[1], M. Muthuraman[2], J. Raethjen[2], G. Deuschl[2], and U. Heute[3]

[1] Department of Electrical and Computer Engineering, Sultan Qaboos University, Oman
abhossen@squ.edu.om, {m.muthuraman,j.raethjen, g.deutschl}@neurologie.uni-kiel.de,
[2] Department of Neurology, College of Medicine, University of Kiel, Germany
[3] Institute of Circuit and System Theory, Faculty of Engineering, University of Kiel, Germany
uh@tf.uni-kiel.de

Abstract. A new technique for discrimination between Parkinsonian tremor and essential tremor is investigated in this paper. The method is based on spectral analysis of both accelerometer and surface EMG signals with neural networks. The discrimination system consists of two parts: feature extraction part and classification (distinguishing) part. The feature extraction part uses the method of approximate spectral density estimation of the data by implementing the wavelet-based soft decision technique. In the classification part, a machine learning approach is implemented using back-propagation supervised neural network. The data has been recorded for diagnostic purposes in the Department of Neurology of the University of Kiel, Germany. Two sets of data are used. The training set, which consists of 21 essential-tremor (ET) subjects and 19 Parkinson-disease (PD) subjects, is used to obtain the important features used for distinguishing between the two subjects. The test data set, which consists of 20 ET and 20 PD subjects, is used to test the technique and evaluate its performance.

Keywords: Wavelet-Decomposition, Soft-Decision Technique, Parkinsonian Tremor, Essential Tremor, EMG, Accelerometer, Artificial Neural Networks.

1 Introduction

Essential tremor (ET) and the tremor in Parkinson's disease (PD) are the two most common pathological tremor forms encountered in clinical neurology [1]. Differential diagnosis between the two tremors is usually achieved clinically. But there is a certain overlap in the clinical presentation between the two diseases that can make the differentiation on purely clinical grounds difficult [2]. In such cases, functional imaging of the dopaminergic deficit as the hallmark of PD is considered the diagnostic gold standard [3-4]. However, this requires SPECT (Single Photon Emission Computer Tomography) technology, injection of a radioactivity-labeled

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1051–1060.
springerlink.com

dopamine transporter ligand into the patients (DAT-Scan), and needs a considerable amount of time. Thus more readily available and easier diagnostic tests are desirable [5]. Spectral analysis of tremor time-series recorded by accelerometry and surface EMG is a common approach [6]. It has proven useful to distinguish between physiological and pathological tremor [7], but is not superior to the clinical assessment in the distinction of ET from PD in its present form [8]. Therefore methods beyond the standard spectral analysis of the recorded tremor time-series have been applied to safely separate ET and PD [9-14]. A new approach of spectral analysis is investigated in [15]. This approach is based on a soft-decision wavelet-decomposition technique and it succeeds in obtaining 85% accuracy of discrimination of ET from PD. In this paper the wavelet-based soft-decision approach is combined with a neural network. The organization of the paper is as follows:

In section 2, both the trial data and test data are described. Section 3 contains the main idea of the soft-decision wavelet-based technique. The results of its implementation on test data using neural networks and discussion of the results are given in section 4. Conclusions are given in section 5.

2 Data

In this study, a total of 39 PD and 41 ET subjects were analyzed. The training set consists of 21 ET and 19 PD subjects, while the test set consists of 20 ET and 20 PD subjects. The mean age, sex and disease duration of the PD patients were compared with the ET patients for the trial and test data in Tables 1 & 2, respectively.

Table 1. Description of trial data of both PD and ET subjects

	PD	**ET**
Number of Patients	19	21
Mean Age (Range)	64.54 (40-90) Years	63.24 (27-94) Years
Gender (Male/Female)	11/8	12/9
Mean Disease Duration	16.4 Years	34 Years

Table 2. Description of test data T of both PD and ET subjects

	PD	ET
Number of Patients	20	20
Mean Age (Range)	68.22 (52-85) Years	64.52 (32-86) Years
Gender (Male/Female)	12/8	11/9
Mean Disease Duration	15.3 Years	29 Years

A piezoelectric accelerometer of about 2g was fixed to the dorsum of the more affected hand (selected clinically) in the middle of the third metacarpal bone, and bipolar surface-EMG recordings with silver-silver-chloride electrodes from forearm flexors (EMG1) and extensors (EMG2) were taken. All data were sampled at 800 Hz. The EMG was band-pass filtered between 50 and 350 Hz and full-wave rectified. The relatively high sampling frequency was useful for the EMG recordings as within the bursts there are frequency components up to 350 Hz and can only be fully picked up with such a sampling frequency to satisfy the Nyquist theorem.

3 Soft-Decision Wavelet-Decomposition

3.1 Wavelet-Decomposition

The block-diagram of a one-stage wavelet-decomposition is shown in Fig.1.

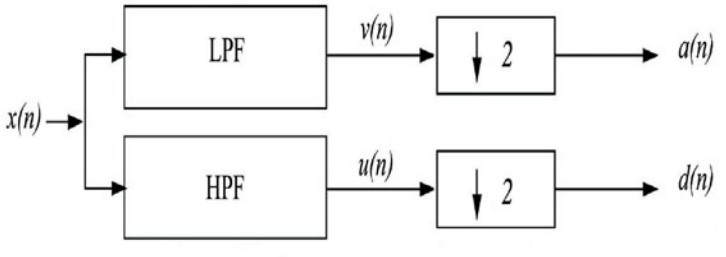


Fig. 1. A Single Stage of Wavelet Decomposition

The input signal $x(n)$ of length-N is first filtered by low-pass (LPF) and high-pass (HPF) filters and then down-sampled by a factor of 2 to produce both the "approximation" $a(n)$ and the "details" $d(n)$. Assuming Hadamard-filters are used, a(n) and $d(n)$ can be obtained by:

$$a(n) = \frac{1}{2}[x(2n) + x(2n+1)],$$

$$d(n) = \frac{1}{2}[x(2n) - x(2n+1)] \tag{1}$$

If there is no information about the energy distribution of the input sequence, a band-selection algorithm [16] can be used to decide (as a hard decision) which band is to be computed or kept for more processing. This method depends on the energy comparison between the low- and high-frequency subsequences after the down sampling in Fig.1.

$$B = \sum_{n=0}^{\frac{N}{2}-1} (a(n))^2 - (d(n))^2 \tag{2}$$

According to the sign of B, the decision is taken: If B is positive, the low-frequency band is considered, and if B is negative, the high-frequency band is considered. Since we are not interested in the value of B, but only in its sign, a more-simpler equation than Eq.2 can be obtained approximately as [17]:

$$\mathrm{sgn}(B) = \mathrm{sgn} \sum_{n=0}^{\frac{N}{2}-1} (|a(n)| - |d(n)|) \tag{3}$$

3.2 Soft-Decision Algorithm

The soft-decision algorithm can be summarized in the following steps:

1) The one stage decomposition (Fig.1) is computed with all branches up to a certain pre-selected stage.
2) All estimator results up to this stage are stored, and their outputs are given a probabilistic interpretation by assigning a probability measure to each path.
3) If J(L) is the assigned probability of the input signal being primarily low-pass, the number J(H) = 1- J(L) is the probability that the signal is primarily high-pass.

At the following stage, the resulting estimate can be interpreted as the conditional probability of the new input sequence containing primarily low (high) frequency components, given that the previous branch was predominantly low (high)-pass. Using this reasoning and laws of probability, the assignments for the probability measure of the resulting subbands should be made equal to the product of the previous branch probability and the conditional probability estimated at a given stage (see Fig.2). The above probabilities derived from the estimator outputs may be interpreted themselves as a coarse measurement of the power spectral density

[17]. The higher the probability value of a particular band, the higher is its power-spectral content! So, after m-stage decomposition, a staircase approximation of the PSD is obtained, when the 2^m probabilities are plotted. For m=8, 256-subbands are resulted, each covering 400/256 Hz of the spectrum (0-400) Hz.

4 The Artificial Neural Networks

Before discussing the classification network used in this work, let us define the key features used in classification. The power spectral density of the first 16 bands out of 256 bands of Accelerometer, EMG1 and EMG2 signals is used as key features of the neural network. The selection of the 16 bands was on a prior information that the frequency of both tremors and their important harmonics are allocated in those bands.

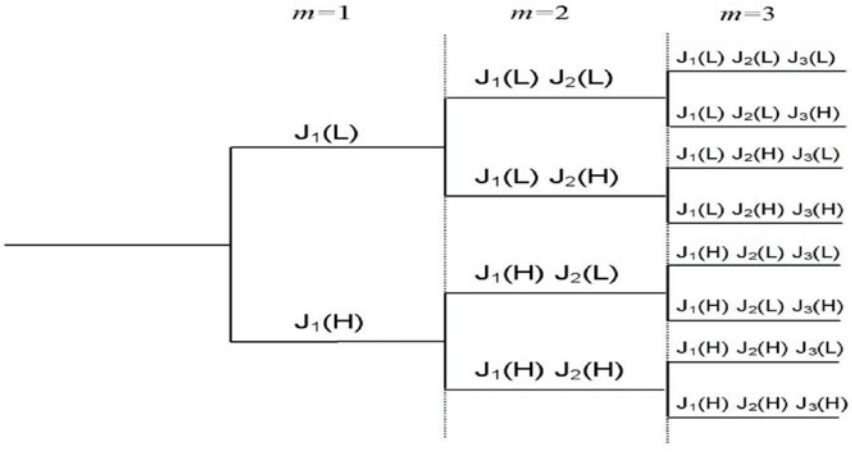


Fig. 2. The Soft-Decision Algorithm

4.1 Supervised Neural Network

In supervised learning, the training input data is composed of feature vector and the corresponding target output values. This approach is commonly described as learning with a teacher, since the desired output of a given input vectors is known and used during the learning process. The implementation of the supervised classification network is done according to the following two steps:

4.1.1 Training Stage of the Supervised ANN

A neural network of the type feed-forward back-propagation [18] (referred to as a multi-layer perceptron) is used in this approach. This network consists of three layers. The first layer (input layer) accepts 16 input signals (B1 to B16) from the

outside world and redistributes these signals to all neurons in the second layer. Actually, the input layer does not include computing neurons. The second layer (hidden layer) with a size of (3, 1) has three hyperbolic tangent sigmoid "*tansig*" neurons. The selection of "*tansig*" neurons was due to the nature of the difference of input features (PSD of the 16 bands) between the two classes (PD and ET) under investigation. Any PSD value is located between a maximum and a minimum, which can be easily simulated by a "*tansig*" function. The Neurons in the hidden layer detect the features; the weights of the neurons represent the features hidden in the input patterns. These features are then used by the third layer (output layer) in determining the output pattern. This third layer has one linear "*purelin*" neuron in our approach since the output is one out of two cases (PD or ET). The whole network has a single output that corresponds to one out of the two types under classification (ET or PD). The features extracted are the outputs of the properly designed hidden layer.

Fig.3 shows the three-layer back-propagation neural network used in the training. Since the neural network needs large set of data for training, a 2000 data set (1000 PD and 1000 ET) is simulated randomly to satisfy the spectral ranges of the 16 bands obtained from 21 ET and 19 PD (the original training data). Any new data is simulated by assignments of 16 PSD values corresponding to the 16 frequency bands in such a way that the PSD of any region is selected randomly between the minimum PSD and the maximum PSD of that region in the original set of data. The assigned PSD value is a random number having a mean value equals the average between the minimum PSD and the maximum PSD and a standard deviation with 10% to 20% of the maximum PSD.

4.1.2 Testing Stage of the Supervised ANN

At this stage the power spectral densities of the test data (1000 ET and 1000 PD), simulated randomly to satisfy the spectral ranges of the 16 bands obtained from 20 ET and 20 PD subjects (of the original test data set), are fed to the 16 inputs of the neural network. The assumed output is either 1 for PD or 2 for ET. In this work, binary classification is considered, e.g. classification between two different cases, positive (PD) and negative cases (ET). The performance of a classifier is evaluated by three main metrics: Specificity, Sensitivity and Accuracy as follows [19]:

$$Specificity\ (\%) = \frac{TN}{TN + FP}.100 \tag{4}$$

$$Sensitivity\ (\%) = \frac{TP}{TP + FN}.100 \tag{5}$$

$$Accuracy\,(\%) = \frac{TP + TN}{T}.100 \tag{6}$$

where the entities in the above equations are defined in confusion matrix shown in Table 3, and T is the total number of data under test.

Table 3. The Confusion Matrix

	Predicted Class [a]	
Actual Class [a]	**Positive (P)**	**Negative (N)**
Positive (P)	TP	FN
Negative (N)	FP	TN

[a]Positive = PD, Negative = ET, T=True, F=False

Results are shown in Tables 4 and 5 using each signal (Accel., EMG1, EMG2) individually and using all three signals with 48-inputs to the neural network . In Table 5, the training and testing sets are interchanged to test the consistency of the algorithm and its data independency.

Table 4. Results obtained from test data

Signal	**Specificity**	**Sensitivity**	**Accuracy**
Accel.	59%	70.6%	64%
EMG1	98.5%	98.3%	98.4%
EMG2	55.2%	95.6%	75.4%
All Signals	88.2%	95%	91.6%

Table 5. Results obtained from training data

Signal	**Specificity**	**Sensitivity**	**Accuracy**
Accel.	85.8%	86.9%	86.3%
EMG1	80%	89.6%	84.8%
EMG2	92.5%	76.9%	84.7%
All Signals	94.9%	94%	94.4%

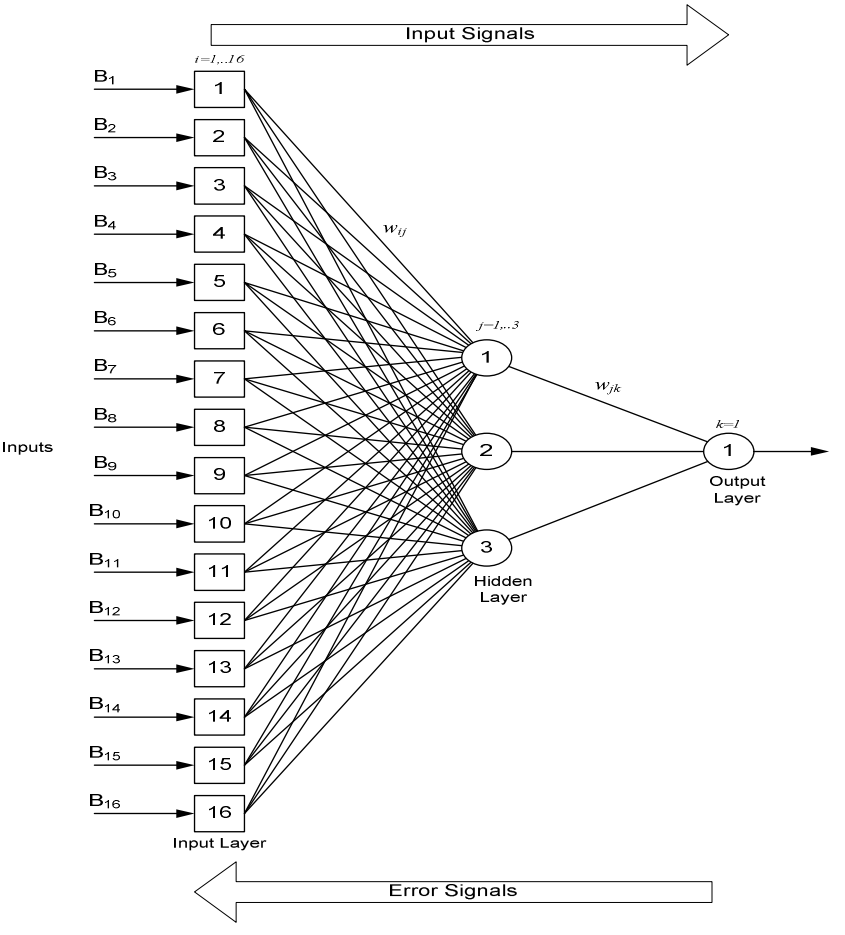


Fig. 3. The Artificial Neural Network Structure

5 Conclusions

A new identification method for PD from ET subjects is investigated. The method is based on the soft-decision wavelet-decomposition power spectral estimation and neural network. The first 16-bands out of 256 bands are used to represent the power-spectral density that forms the classification features. The accuracy of classification approaches 91.6% and 94.4% by testing the designed supervised neural network on test data set and training data set, respectively. In brief, the technique used in this paper, is a complete distinguishing system between ET and PD, that is data-independent, simple, efficient with automatic results.

References

[1] Harris, M.K., Shneyder, N., Borazanci, A., Korniychuk, E., Kelley, E., Minagar, A.: Movement Disorders. Med. Clin. North America 93, 371–388 (2009)

[2] Deuschl, G., Bain, P., Brin, M.: Consensus statement of the movement disorder society on tremor. Ad Hoc Scientific Committee, Mov. Disorder 13(suppl. 3), 2–23 (1998)

[3] Marshall, V., Reininger, C.B., Marquardt, M., Patterson, J., Hadley, D.M., Oertel, W.H., Benamer, H.T., Kemp, P., Burn, D., Tolosa, E., Kulisevsky, J., Cunha, L., Costa, D., Booij, J., Tatsch, K., Chaudhuri, K.R., Ulm, G., Pogarell, O., Hoffken, H., Gerstner, A., Grosset, D.G.: Parkinson's disease is overdiagnosed clinically at baseline in diagnostically uncertain cases: A 3-year European multicenter study with repeat [(123)I] FP-CIT SPECT. Mov. Disord. 24, 499–507 (2008)

[4] Djaldetti, R., Nageris, B.I., Lorberboym, M., Treves, T.A., Melamed, E., Yaniv, E.: [(123)I]- FP-CIT SPECT and oldfaction test in patients with combined postural and rest tremor. J. Neural Transm. 115, 469–472 (2008)

[5] Antonini, A., Berto, P., Lopatriello, S., Tamma, F., Annemans, L., Chambers, M.: Cost-effectiveness of [(123)I]-FP-CIT- SPECT in the differential diagnosis of essential tremor and Parkinson's disease in Italy. Mov. Disord. 23, 2202–2209 (2008)

[6] Deuschl, G., Krack, P., Lauk, M., Timmer, J.: Clinical neurophysiology of tremor. J. Clin. Neurophysiol. 13, 110–121 (1996)

[7] Raethjen, J., Lauk, M., Koster, B., Fietzek, U., Friege, L., Timmer, J., Lucking, C.H., Deuschl, D.: Tremor analysis in two normal cohorts. Clin. Neurophysiol. 115, 2151–2156 (2004)

[8] Bain, P., Brin, M., Deuschl, G., Elble, R., Jankovic, J., Findley, L., Koller, W.C., Pahwa, R.: Criteria for the diagnosis of essential tremor. Neurology 54, S7 (2000)

[9] Deuschl, G., Lauk, M., Timmer, J.: Tremor classification and tremor time series analysis. CHAOS 5(1), 48–51 (1995)

[10] Spieker, S., Jentgens, C., Boose, A., Dichgans, J.: Reliability, specificity and sensitivity of long-term tremor recordings. Electroencephalography and Clinical Neurophysiology 97, 326–331 (1995)

[11] Sapir, N., Karasik, R., Havlin, S., Simon, E., Hausdorff, J.M.: Detecting scaling in the period of dynamics of multimodal signals: Application to Parkinsonian tremor. Physical Review E 67(031903), 1–8 (2003)

[12] Breit, S., Spieker, S., Schulz, J.B., et al.: Long-term EMG recordings differentiate between parkinsonian and essential tremor. J. Neurol. 255, 103–111 (2008)

[13] Elble, R.J.: Essential tremor frequency decreases with time. Neurology 55, 1547–1551 (2000)

[14] Cohen, O., Pullman, S., Jurewicz, E., et al.: Rest tremor in patients with essential tremor: prevalence, clinical correlates, and electrophysiologic characteristics. Arch. Neurol. 60, 405 (2003)

[15] Hossen, A., Muthuraman, M., Raethjen, J., Deuschl, G., Heute, U.: Discrimination of Parkinsonian tremor from Essential tremor by implementation of a wavelet-based soft-decision technique on EMG and Accelerometer signals. Biomedical Signal Processing and Control 5, 181–188 (2010)

[16] Hossen, A., Heute, U.: Fully Adaptive Evaluation of SB-DFT. In: Proceedings of IEEE Int. Symp. on Circuits and Systems, Chicago, Illinois (1993)
[17] Hossen, A.: Power spectral density estimation via wavelet decomposition. Electronics Letters 40(17), 1055–1056 (2004)
[18] Principe, J., Euliano, N., Lefebvre, W.: Neural and Adaptive Systems: Fundamentals Through Simulations. John Wiley & Sons (2000)
[19] Rangayyan, R.M.: Biomedical Signal Analysis: A Case-Study Approach. IEEE Press, NJ (2001)

Application of Binary Particle Swarm Optimization in Cryptanalysis of DES

Shimpi Singh Jadon, Harish Sharma, Etesh Kumar,
and Jagdish Chand Bansal

ABV-Indian Institute of Information Technology and Management, Gwalior
{shimpisingh2k6,eteshkumar,jcbansal}@gmail.com,
harish0107@rediffmail.com

Abstract. Feistel cipher based cryptographic algorithms like DES are very hard for the cryptanalysts as their internal structures are based on high non-linearity and low autocorrelation. It has been shown that the traditional and brute-force type attacks are insignificant for the cryptanalysis of this type of algorithms. Swarm intelligence is an exciting new research field and shown their effectiveness, robustness to solve a wide variety of complex problems. Therefore, in this paper, Binary Particle Swarm Optimization (BPSO) strategy is used for cryptanalysis of DES symmetric key cryptographic algorithm. The reported results show that it is very promising to solve block cipher based cryptographic optimization problem through meta heuristic techniques.

Keywords: Meta Heuristics, Particle swarm optimization, Data Encryption Standard, Cryptanalysis, Evolutionary computation.

1 Introduction

Swarm Intelligence is a meta-heuristic method in the field of artificial intelligence that is used to solve optimization problems. It is based on the collective behavior of social insects, flocks of birds, or schools of fish. However, a swarm can be considered as any collection of interacting agents or individuals. These animals can solve complex tasks without centralized control.

Researchers have analyzed such behaviors and designed algorithms that can be used to solve combinatorial and numerical optimization problems in many science and engineering domains. Previous research [16, 8, 12] has shown that algorithms based on Swarm Intelligence have great potential. The algorithms that have emerged in recent years include Ant Colony Optimization (ACO) [16] based on the foraging behavior of ants, and Particle Swarm Optimization (PSO) [8] based on the behaviors of bird flocks and fish schools.

Particle Swarm Optimization is one of the stochastic optimization technique and in this paper it is used for cryptanalysis of a feistel cipher based algorithm. The main reason for using PSO here is its global solution finding

K. Deep et al. (Eds.): Proceedings of the International Conference on SocProS 2011, AISC 130, pp. 1061–1071.
springerlink.com

property, without any need to give initial approximation to unknown parameters. Particle Swarm optimization (PSO) technique is considered as one of the modern heuristic algorithms for optimization introduced by James Kennedy and Eberhart in 1995 [8]. It is based on the social behavior metaphor [8]. It is a population based optimization technique, which is sometimes regarded as an alternative tool to Genetic Algorithm (GAs) and other Evolutionary Algorithms (EAs) and gained a lot of attention in the recent years. As compared to EAs, PSO is a stochastic search technique with reduced memory requirement, computationally effective and easier to implement. Also PSO has a more global searching ability at the beginning of the run and has greater local search ability near the end of the run [1].

Cryptanalysis is the study of methods for obtaining the meaning of encrypted information. The aim of this paper is to use PSO and its variant Binary PSO for cryptanalysis of Feistel-cipher based cryptosystems. More specifically, in this paper, Binary PSO is applied for finding some missing bits of the key of data encryption standard (DES) symmetric key cryptographic algorithm for four rounds. Furthermore, a comparative study also carried out for Binary PSO and PSO in terms of efficiency and accuracy.

The rest of the paper is organized as follows: In Section 2, PSO algorithm is briefly described. In section 3, DES cryptosystem and differential cryptanalysis are briefly reviewed and the optimization problem is formulated. The experimental results are reported in section 4. Finally, in section 5, conclusions are derived and directions for future work are suggested.

2 Brief Overview of Particle Swarm Optimization

The field of nature-inspired metaheuristics is mostly constituted by the evolutionary algorithms [comprising of GAs, EP, ESs, differential evolution (DE), and so on] as well as the swarm intelligence algorithms [e.g., ant colony optimization (ACO) [6], particle swarm optimization (PSO) [8] and bacterial foraging optimization (BFO) [11]]. From the aforementioned algorithms, in this paper, Binary version of PSO is used for cryptanalysis of four-rounded DES.

Particle Swarm optimization (PSO) technique is considered as one of the modern heuristic algorithms for optimization introduced by James Kennedy and Eberhart in 1995 [8]. The idea behind PSO is based on the simulation of the social behavior of bird flock and fish schools. PSO is a swarm intelligence method for global optimization problems. It differs from well-known evolutionary algorithms as in evolutionary algorithms a population of potential solutions is used to probe the search space, but no operators, inspired by evolution procedures, are applied on the population to generate new promising solutions. Instead in PSO, each individual, namely *particle*, of the population, called *swarm*, adjusts its trajectory towards its own previous best position (pbest), and towards the previous best position of any member of its topological neighborhood (gbest). Two variants of the PSO have been

developed, one with a global neighborhood and the other with a local neighborhood. According to the global variant, each particle moves towards its best previous position and towards the best particle in the whole swarm. On the other hand, in the local variant, each particle moves towards its best previous position and towards the best particle in its restricted neighborhood. Working of PSO may be briefly described as under:

Suppose the search space is D-dimensional, then the i^{th} particle of the swarm can be represented by a D-dimensional vector, $X_i = (x_{i1}, x_{i2}, ..., x_{iD})^T$. The velocity (position change) of this particle can be represented by another D-dimensional vector $V_i = (v_{i1}, v_{i2}, ..., v_{iD})^T$. The best previously visited position of the i^{th} particle is denoted as $P_i = (p_{i1}, p_{i2}, ..., p_{iD})^T$. Defining g as the index of the best particle in the swarm, the swarm is manipulated according to the following two equations:

Velocity Update Equation:

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{1}$$

Position Update Equation:

$$x_{id} = x_{id} + v_{id} \tag{2}$$

where $d = 1, 2, ..., D$; $i = 1, 2, ..., S$, where S is the size of the swarm; c_1 and c_2 are constants, called cognitive and social scaling parameters respectively (usually, $c_1 = c_2; r_1, r_2$ are random numbers, uniformly distributed in [0, 1]). Equations (1) and (2) define the initial version of PSO algorithm. A constant, $Vmax$, was used to arbitrarily limit the velocities of the particles and improve the resolution of the search. The pseudo code of PSO is shown below:

Algorithm PSO:
for t= 1 to the max. bound of the number on iterations **do**
  for i=1 to the swarm size **do**
    for d=1 to the problem dimensionality **do**
      Apply the velocity update equation:
      Update Position
    end for
    Compute fitness of updated position;
    if Needed **then**
      update historical information for P_i and P_g;
    end if
  end for
  Terminate if P_g meets problem requirements;
end for

The maximum velocity $Vmax$, serve as a constraint to control the global exploration ability of particle swarm. A larger $Vmax$ facilitates global exploration, while a smaller $Vmax$ encourages local exploitation. The concept of an

inertia weight was also developed to better control exploration and exploitation. The motivation was to be able to eliminate the need for $Vmax$. The inclusion of an inertia weight in the particle swarm optimization algorithm was first reported in the literature in 1998 [13, 14].

After some experience with the inertia weight, it was found that although the maximum velocity factor, $Vmax$, could not always be eliminate, the particle swarm algorithm works well if $Vmax$ set to the value of the dynamic range of each variable (on each dimension). The resulting velocity update equation becomes:

$$v_{id} = w \times v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{3}$$

Eberhart and Shi [7] indicate that the optimal strategy is to initially set w to 0.9 and reduce it linearly to 0.4, allowing initial exploration followed by acceleration toward an improved global optimum. Maurice Clerc [3] has introduced a constriction factor, χ , which improves PSO's ability to constrain and control velocities. χ is computed as:

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi(\phi - 4)}|} \tag{4}$$

where $\phi = c_1 + c_2$, $\phi > 4$, and the velocity update equation is then

$$v_{id} = \chi \times (v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})) \tag{5}$$

Eberhart and Shi [7] found that χ , combined with constraints on $Vmax$, significantly improved the PSO performance.

Discrete binary version of PSO was proposed by Kennedy and Eberhart for binary problems [9]. In their model a particle will decide on "yes" or " no", "include" or "not to include" etc. These binary values can be a representation of a real value in binary search space. Same, as in continuous version, the particle's personal best and global best is updated in binary PSO. The major difference between binary and continuous version of PSO is that in binary PSO, velocities of the particles are rather defined in terms of probabilities that a bit will change to one. According to above definition a velocity must be within the range [0,1]. So a sigmoid function defined as $(sig(v) = (1 + e^{-v_{id}})^{-1})$ is used to map all real valued numbers of velocity to the range [0,1] [9]. Also to update the velocity vector of the particle, equation (3) is used. The equation (6) is used for obtaining the new position of the particle.

$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } U(0,1) < sig(v_{ij}(t+1)) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Where U(0,1) is the uniform random number.

3 DES as an Optimization Problem

3.1 Data Encryption Standard

The data encryption standard (DES) is basically a Feistel iterated block cipher. For DES, data are encrypted in 64-bit blocks using a 56- bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output.To reverse the encryption same steps, with the same key are used but in reverse order. Input to the DES cipher is plaintext P and output is ciphertext C, with each block 64 bits. Through a 16-round process, the current 64-bit word is divided into two parts 32-bit each, the left part L_i and the right part R_i. Then round i, $1 \leq i \leq 16$ is defined as follows
$L_i = R_{i-1}$,
$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$,
Where K_i is derived from the cipher key K via the key scheduling algorithm, K_i is the 48 bits subkey used in ith round, and F is a round function.

The round function F is the main component of DES. Mainly three operations are involved in round function i.e substitution, permutation and XOR operations (denoted by $\oplus$). The S-boxes are the only nonlinear part of DES and basically S-boxes are nonlinear mapping. There are eight S-boxes each of which is a substitution mapping 6 to 4 bits. Detailed description of DES is given in [5].

Differential cryptanalysis is a chosen plaintext attack meaning that the attacker must be able to obtain encrypted ciphertexts for some set of plaintexts of his choosing. The two plaintexts can randomly be chosen, as long as they satisfy the difference condition. In this method the effect of particular differences in plaintext pairs on the differences of the resultant ciphertext pairs is analysed. Using these differences, probabilities can be assigned to the possible keys and locate the most probable key. The difference is chosen as a fixed XORed value of the two plaintexts for cryptosystems similar to DES.

Characteristic is the most important component in differential cryptanalysis. The definition of a characteristic is the following. "Associated with any pair of encryptions are the XOR value of its two plaintexts, the XOR of its ciphertexts, the XORs of the input of each round in the two executions and the XORs of the outputs of each round in the two executions. These XOR values form an r-round characteristic . A characteristic has a probability, which is the probability that a random pair with the chosen plaintext XOR has the round and ciphertext XORs specified in the characteristic" [2].

3.2 Formulation of the Problem

Biham and Shamir [2] use a one-round characteristic with probability 1 for the differential cryptanalysis of DES reduced to four rounds, and at the first step differential cryptanalysis provides 42 bits of the subkey of the last round. Subkeys are calculated using the DES key scheduling algorithm, differential

cryptanalysis gives 42-bit which are actual key bits of the 56 key bits and still there are 14 key bits missing. One of the method for finding these key bits was to try all the 2^{14} possibilities in decrypting the given ciphertexts, using the resulting keys. Known plaintext XOR value for all the pairs that are used by differential cryptanalysis should be satisfied with the right key. The probability for remaining $2^{14} - 1$ values of the key will be only 2^{-64} to satisfy the pairs condition [2].

We formulate the problem of finding the missing 14 bits to an optimization one, as follows. each one of the 14 bits is considered as a component of a 14^{th} dimensional vector. This vector represents a possible solution of the problem. Now, assume that the right 42 key bits which were obtained by differential cryptanalysis were suggested using *np* pairs. These *np* pairs can be used to evaluate the possible solutions provided by the optimization method.For each possible solution, X_i , obtained by the optimization algorithm, we construct the 56 bits of the key comprising 42 bits which are known by differential cryptanalysis and the 14 components of X_i in proper order. All *np* ciphertext pairs that were used by differential cryptanalysis decrypted with the resulting key and number of decrypted pairs that satisfy the known plaintext XOR value is counted, denoted as cnp_{X_i}. Thus, the evaluation function *f*, is the bitwise difference between the desired output *np* and the actual output cnp_{X_i} given by [10]:

$$f(X_i) = np - cnp_{X_i} \tag{7}$$

For the above function f, zero is the global minimum and actual key will be the global minimizer with probability $P = 1 - 2^{-64}$.

4 Experimental Results and Discussion

The PSO and BPSO methods, are applied on the considered problem with two different examples. Subsequent subsections contain the experimental settings and results with proper statistical analysis using Performance indices [4] and boxplots [15].

4.1 Experimental Settings

The experimental settings of various parameters for BPSO and PSO, are shown as follows. This is the most recommended parameter setting.

1. Iterations/Generations=$Swarm \times 10$,
2. Total Runs=100,
3. Acceleration constants (c_1, c_2)=(2.05,2.05),
4. Inertia Weigh= Linearly decreasing from (0.8 to 0.1),
5. Maximum Velocity (V_{max})=2.

4.2 Experimental Results

In order to compare the performance of Binary PSO (BPSO) and PSO, two examples are taken into consideration (Listed in Table 1.1 and Table 1.2). In example-1, 5 test pairs and in example-2, 10 test pairs are considered for finding the missing 14 key bits. Numerical results obtained by BPSO and PSO are tabulated in Table 1.1 and Table 1.2. In Table 1.1 and 1.2, success rate (SR) and average function evaluations (AFE) are shown for both the algorithms.

Table 1. Comparison of the Methods for 5 Test Pairs

Key	PSO		Binary PSO	
	SR	AFE	SR	AFE
k_1	100	2815	100	1540
k_2	97	2516	99	1598
k_3	98	2704	100	1645
k_4	100	2393	100	1324
k_5	98	2861	100	1554
k_6	99	2414	100	1462

Table 2. Comparison of the Methods for 10 Test Pairs

Key	PSO		Binary PSO	
	SR	AFE	SR	AFE
k_1	100	2613	100	1465
k_2	99	2105	100	1286
k_3	100	2463	100	1436
k_4	100	2198	100	1232
k_5	98	2487	100	1327
k_6	100	2253	100	1121

4.3 Statistical Analysis

In order to extract the best strategy for finding the missing 14 key bits, a comparative analysis is done for PSO and $BPSO$. Statistical comparison is carried out using performance index [4] and boxplot.

In order to compare the consolidated performance of $BPSO$ and PSO search heuristics, the value of a performance index PI [4] is computed. This index gives a weighted importance to the success rate and the average number of function evaluations. The value of this performance index for a computational algorithm under comparison is given by

$$PI = \frac{1}{N_p} \sum_{i=1}^{N_p} (k_1 \alpha_1^i + k_2 \alpha_2^i)$$

Where $\alpha_1^i = \frac{Sr^i}{Tr^i}$; $\alpha_2^i = \begin{cases} \frac{Mf^i}{Af^i}, & \text{if } Sr^i > 0. \\ 0, & \text{if } Sr^i = 0. \end{cases}$;

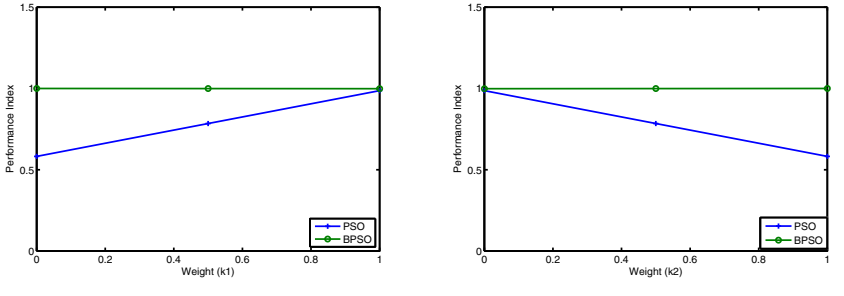
- Sr^i = Number of successful runs of i^{th} problem
- Tr^i = Total number of runs of i^{th} problem
- Mf^i = Minimum of average number of function evaluations of successful runs used by all algorithms in obtaining the solution of i^{th} problem
- Af^i = Average number of function evaluations of successful runs used by an algorithm in obtaining the solution of i^{th} problem
- N_p = Total number of problems analyzed.

k_1 and k_2 ($k_1 + k_2 = 1$ and $k_1, k_2 \leq 1$) are the weights assigned to success rate and average number of function evaluations of successful runs respectively. From above definition it is clear that PI is a function of k_1 and k_2. Since $k_1 + k_2 = 1$, one of $k_i, i = 1, 2$ could be eliminated to reduce the number of dependent variables from the expression of PI. We adopt the same methodology as given in [4] i.e. equal weights are assigned to two terms at a time in the PI expression. This way PI becomes a function of one variable. The resultant cases are as follows:

1. $k_1 = W, k_2 = \frac{1-W}{2}, 0 \leq W \leq 1$;
2. $k_2 = W, k_1 = \frac{1-W}{2}, 0 \leq W \leq 1$;

For example-1 (Table 1.1), graphs for cases (1) and (2) are shown in Fig. 1.1(a) and 1.1(b) respectively and for example-2 (Table 1.2), graphs for cases (1) and (2) are shown in Fig. 1.2(a) and 1.2(b) respectively. In these figures the horizontal axis represents the weight W and the vertical axis represents the performance index PI.

PIs of both algorithms ($PSO, BPSO$) are superimposed in the Figures 1.1 and 1.2 for comparison and to get a ranked order of the performance. It is observed that for $BPSO$, the value of PI is more than PSO. It is concluded from PI that $BPSO$ is proved to be the a better choice compared to PSO to solve these type of optimization problems.



(a) (b)

Fig. 1. Performance index analysis for e.g. 1; (a) for case (1) and (b) for case (2)

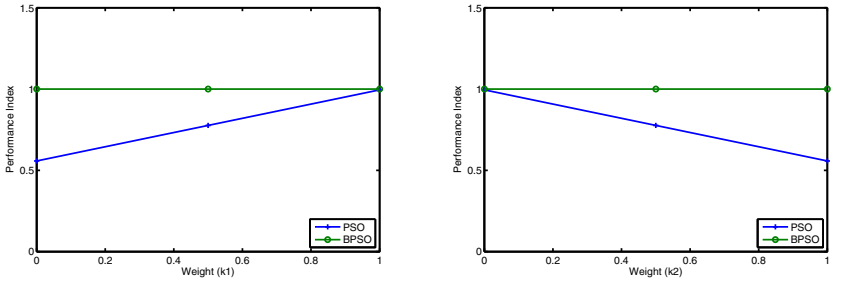


Fig. 2. Performance index analysis for e.g. 2; (a) for case (1) and (b) for case (2)

For the purpose of comparison in terms of performance, boxplot analysis is carried out for both the algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [15]. Analysis of univariate expression, where the variability of measurements may be affected by many parameters is effectively done by the boxplot tool. Degree of dispersion and skewness in the data are easily analyzed by measuring the spacings between the different parts of the box. The boxplot based on average function evaluation for example-1 and example-2 for both the algorithms are shown in Fig. 1.3-1.4. For example-1, it is clear from Fig. 1.3 that the performance of BPSO is better than PSO. For example-2, it is observed from Fig. 1.4 that BPSO performs better than PSO.

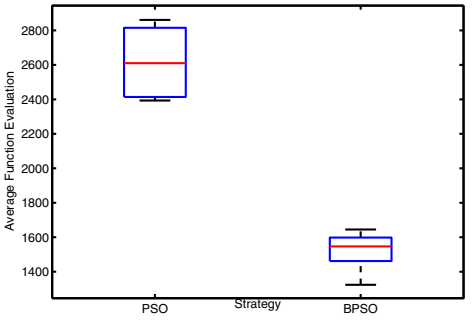


Fig. 3. Boxplot graph of average function evaluation for example 1

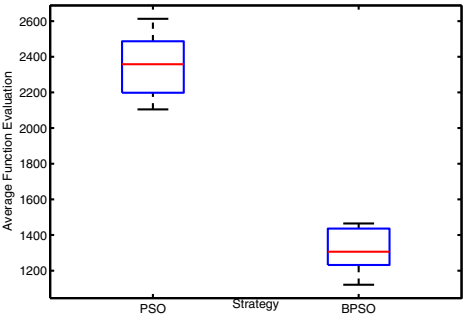


Fig. 4. Boxplot graph of average function evaluation for example-2

5 Conclusion

In this paper, Binary Particle Swarm Optimization (BPSO) method is analyzed for the cryptanalysis of DES block ciphers. Differential cryptanalysis can locate 42 actual key bits of the 56 key bits of DES and then BPSO is applied for finding 14 missing key bits. Since its inception, PSO has gained wide popularity due to its ability to solve complex optimization problems in a very efficient and simple manner. The results produced in this paper show that the use of BPSO is a better choice for the cryptanalysis of feistel cipher based cryptographic algorithms. A comparative analysis of BPSO and PSO also carried out. In future, it will be very interesting to see the use of evolutionary algorithms for cryptanalysis of these type of optimization problems with higher intensity.

Acknowledgement. First, second and third Author acknowledge ABV- Indian Institute of Information Technology and Management Gwalior for providing research grant under MHRD scheme to carry out this work.

References

[1] Arumugam, M.S., Rao, M.V.C., Tan, A.W.C.: A novel and effective particle swarm optimization like algorithm with extrapolation technique. Applied Soft Computing 9(1), 308–320 (2009)
[2] Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. Journal of CRYPTOLOGY 4(1), 3–72 (1991)
[3] Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 3. IEEE (1999)
[4] Thakur, M., Deep, K.: A new crossover operator for real coded genetic algorithms. Applied Mathematics and Computation 188(1), 895–911 (2007)

[5] Diffie, W., Hellman, M.E.: Special feature exhaustive cryptanalysis of the nbs data encryption standard. Computer 10(6), 74–84 (1977)
[6] Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 2. IEEE (1999)
[7] Eberhart, R.C., Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 84–88. IEEE (2000)
[8] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
[9] Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics, 'Computational Cybernetics and Simulation', vol. 5, pp. 4104–4108. IEEE (1997)
[10] Laskari, E.C., Meletiou, G.C., Stamatiou, Y.C., Vrahatis, M.N.: Evolutionary computation based cryptanalysis: A first study. Nonlinear Analysis 63(5-7), e823–e830 (2005)
[11] Passino, K.M.: Bacterial foraging optimization. International Journal of Swarm Intelligence Research (IJSIR) 1(1), 1–16 (2010)
[12] Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)
[13] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, pp. 69–73. IEEE (1998)
[14] Shi, Y., Eberhart, R.: Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 591–600. Springer, Heidelberg (1998)
[15] Williamson, D.F., Parker, R.A., Kendrick, J.S.: The box plot: a simple visual method to interpret data. Annals of Internal Medicine 110(11), 916 (1989)
[16] Yagmahan, B., Yenisey, M.M.: Ant colony optimization for multi-objective flow shop scheduling problem. Computers & Industrial Engineering 54(3), 411–420 (2008)

Author Index