Studies in Computational Intelligence 575

Ahmad Taher Azar
Sundarapandian Vaidyanathan *Editors*

Computational Intelligence Applications in Modeling and Control

Springer

Studies in Computational Intelligence

Volume 575

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Ahmad Taher Azar · Sundarapandian Vaidyanathan
Editors

Computational Intelligence Applications in Modeling and Control

Springer

Editors
Ahmad Taher Azar
Faculty of Computers and Information
Benha University
Benha
Egypt

Sundarapandian Vaidyanathan
Research and Development Centre
Vel Tech University
Chennai
India

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-11016-5 ISBN 978-3-319-11017-2 (eBook)
DOI 10.1007/978-3-319-11017-2

Library of Congress Control Number: 2014957150

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2015
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

About the Subject

The development of Computational Intelligence (CI) systems was inspired by observable and imitable aspects of intelligent activity of human beings and nature. The essence of the systems based on computational intelligence is to process and interpret data of various nature, so that that CI is strictly connected with the increase in available data as well as their capabilities of processing mutually supportive factors. Without them, the development of this field would be almost impossible, and its application practically marginal. That is why these techniques have especially developed in recent years. Developed theories of computational intelligence have been quickly applied in many fields of engineering, data analysis, forecasting, biomedicine, and others. They are used in image and sound processing and identifying, signal processing, multidimensional data visualization, steering of objects, analysis of lexicographic data, requesting systems in banking, diagnostic systems, expert systems, and many other practical implementations.

Intelligent control systems are very useful when no mathematical model is available a priori and intelligent control itself develops a system to be controlled. Intelligent control is inspired by the intelligence and genetics of living beings. Some important types of intelligent control are fuzzy logic, artificial neural networks, genetic algorithms, ant colony optimization (ACO), particle swarm optimization (PSO), support vector machines, etc. Intelligent control systems, especially fuzzy logic control systems, have found great applications in engineering and industry.

About the Book

The new Springer book, *Computational Intelligence Applications in Modeling and Control*, consists of 16 contributed chapters by subject experts specialized in the various topics addressed in this book. The special chapters have been brought out in this book after a rigorous review process in the broad areas of Control Systems, Power Electronics, Computer Science, Information Technology, modeling, and engineering applications. Special importance is given to chapters offering practical solutions and novel methods for recent research problems in the main areas of this book, viz., Control Systems, Modeling, Computer Science, IT and engineering applications.

Intelligent control methods can be broadly divided into the following areas:

- Neural network control
- Fuzzy logic control
- Neuro-fuzzy control
- Genetic control
- Expert Systems
- Bayesian control
- Intelligent agents

This book discusses trends and applications of computational intelligence in modeling and control systems engineering.

Objectives of the Book

The objective of this book takes a modest attempt to cover the framework of computational intelligence and its applications in a single volume. The book is not only a valuable title on the publishing market, but is also a successful synthesis of computational intelligence techniques in the world literature. Several multidisciplinary applications in Control, Engineering, and Information Technology are discussed in this book, where CI have excellent potentials for use.

Organization of the Book

This well-structured book consists of 16 full chapters.

Book Features

- The book chapters deal with recent research problems in the areas of intelligent control, computer science, information technology, and engineering.
- The chapters contain a good literature survey with a long list of references.
- The chapters are well-written with a good exposition of the research problem, methodology, and block diagrams.

- The chapters are lucidly illustrated with numerical examples and simulations.
- The chapters discuss details of engineering applications and the future research areas.

Audience

This book is primarily meant for researchers from academia and industry, who are working in the research areas—Computer Science, Information Technology, Engineering, and Control Engineering. The book can also be used at the graduate or advanced undergraduate level as a textbook or major reference for courses such as intelligent control, mathematical modeling, computational science, numerical simulation, applied artificial intelligence, fuzzy logic control, and many others.

Acknowledgments

As the editors, we hope that the chapters in this well-structured book will stimulate further research in computational intelligence and control systems and utilize them in real-world applications.

We hope sincerely that this book, covering so many different topics, will be very useful for all readers.

We would like to thank all the reviewers for their diligence in reviewing the chapters.

Special thanks go to Springer, especially the book Editorial team.

Benha, Egypt Ahmad Taher Azar
Chennai, India Sundarapandian Vaidyanathan

Contents

An Investigation into Accuracy of CAMEL Model of Banking Supervision Using Rough Sets 1
Renu Vashist and Ashutosh Vashishtha

Towards Intelligent Distributed Computing: Cell-Oriented Computing 27
Ahmad Karawash, Hamid Mcheick and Mohamed Dbouk

Application of Genetic Algorithms for the Estimation of Ultrasonic Parameters 55
Mohamed Hesham Farouk El-Sayed

A Hybridized Approach for Prioritizing Software Requirements Based on K-Means and Evolutionary Algorithms 73
Philip Achimugu and Ali Selamat

One-Hour Ahead Electric Load Forecasting Using Neuro-fuzzy System in a Parallel Approach 95
Abderrezak Laouafi, Mourad Mordjaoui and Djalel Dib

A Computational Intelligence Optimization Algorithm Based on the Behavior of the Social-Spider 123
Erik Cuevas, Miguel Cienfuegos, Raul Rojas and Alfredo Padilla

Black Hole Algorithm and Its Applications 147
Santosh Kumar, Deepanwita Datta and Sanjay Kumar Singh

Genetic Algorithm Based Multiobjective Bilevel Programming for Optimal Real and Reactive Power Dispatch Under Uncertainty 171
Papun Biswas

A Monitoring-Maintenance Approach Based on Fuzzy Petri Nets in Manufacturing Systems with Time Constraints 205
Anis Mhalla and Mohamed Benrejeb

Box and Jenkins Nonlinear System Modelling Using RBF Neural Networks Designed by NSGAII 229
Kheireddine Lamamra, Khaled Belarbi and Souaad Boukhtini

Back-Propagation Neural Network for Gender Determination in Forensic Anthropology 255
Iis Afrianty, Dewi Nasien, Mohammed R.A. Kadir and Habibollah Haron

Neural Network Approach to Fault Location for High Speed Protective Relaying of Transmission Lines 283
Moez Ben Hessine, Houda Jouini and Souad Chebbi

A New Approach for Flexible Queries Using Fuzzy Ontologies 315
Amira Aloui and Amel Grissa

An Efficient Multi Level Thresholding Method for Image Segmentation Based on the Hybridization of Modified PSO and Otsu's Method 343
Fayçal Hamdaoui, Anis Sakly and Abdellatif Mtibaa

IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm 369
Nizar Rokbani, Alicia Casals and Adel M. Alimi

Computer Aided Intelligent Breast Cancer Detection: Second Opinion for Radiologists—A Prospective Study 397
J. Dheeba and N. Albert Singh

An Investigation into Accuracy of CAMEL Model of Banking Supervision Using Rough Sets

Renu Vashist and Ashutosh Vashishtha

Abstract Application of intelligent methods in banking becomes a challenging issue and acquiring special attention of banking supervisors and policy makers. Intelligent methods like rough set theory (RST), fuzzy set and genetic algorithm contribute significantly in multiple areas of banking and other important segments of financial sector. CAMEL is a useful tool to examine the safety and soundness of various banks and assist the banking regulators to ward off any potential risk which may lead to bank failure. RST approach may be applied for verifying authenticity and accuracy of CAMEL model and this chapter invites reader's attention towards this relatively new and unique application of RST. The results of CAMEL model have been widely accepted by banking regulators for the purpose of assessing the financial health of banks. In this chapter we have considered ten largest public sector Indian banks on the basis of their deposit-base over a five-year period (2008–2009 to 2012–2013). The analysis of financial soundness of banks is structured under two parts. Part I is devoted to ranking of these banks on the basis of performance indices of their capital adequacy (C), asset quality (A), management efficiency (M), earnings (E) and liquidity (L). Performance analysis has been carried out in terms of two alternative approaches so as to bring implications with regard to their rank accuracy. We named these approaches as *Unclassified Rank Assignment Approach* and *Classified Rank Assignment Approach.* Part II presents analysis of accuracy of ranks obtained by CAMEL model for both the approaches that is for Unclassified Rank Assignment Approach and for Classified Rank Assignment in terms of application of Rough Set Theory (RST). The output of CAMEL model (ranking of banks for both approaches) is given as input to rough set for generating rules and for finding the reduct and core. The accuracy of the

---

R. Vashist (✉)
Faculty of Computer Science, Shri Mata Vaishno Devi University Katra,
Katra, Jammu and Kashmir, India
e-mail: vashist.renu@gmail.com

A. Vashishtha
Faculty of Management, Shri Mata Vaishno Devi University Katra,
Katra, Jammu and Kashmir, India
e-mail: ashu.vashishtha@smvdu.ac.in

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_1

ranking generated by the CAMEL model is verified using lower and upper approximation. This chapter demonstrates the accuracy of Ranks generated by CAMEL model and decisions rules are generated by rough set method for the CAMEL model. Further, the most important attribute of CAMEL model is identified as risk-adjusted capital ratio, CRAR under capital adequacy attribute and results generated by rough set theory confirm the accuracy of the Ranks generated by CAMEL Model for various Indian public- sector banks.

Keywords CAMEL model · Rough set · Rules · Reduct · Core · Ranking of banks

1 Introduction

In recent times, the experience of countries worldwide as regard the financial distress and failures of financial institutions, especially banks, has attracted considerable interest of researchers and policy makers. Due to the potential fragility of banks, the need for their effective regulation and periodical review of financial soundness is being increasingly recognized. The objective of these initiatives is to facilitate timely detection of potential financial distress and prevent the catastrophic effects that would otherwise engulf the financial system. For evaluation of financial soundness of a bank, there is perhaps no other measure that is considered more reliable than the CAMEL rating system—a set of financial ratios or indices. As an early warning system, these ratios were developed by the Federal Deposit Insurance Cooperation (FDIC) in the US in the late 1970s. Initially, these ratios were conceived under five broad categories or components: capital adequacy (C), asset quality (A), management efficiency (M), earnings (E) and liquidity (L). Subsequently, a sixth component, namely, sensitivity to market risk (S) was also added. Thus the rating system came to acquire the acronym CAMELS.

Rough set theory is relatively a new mathematical tool for dealing with vagueness, imprecision and uncertainty. This theory is basically used for data analysis for finding hidden patterns in the data. This theory has been successfully applied to solve many real life problems in medicine, pharmacology, engineering, banking, financial and market analysis and other areas. Basic idea of this methodology hinges on classification of objects of interest into similarity classes (clusters) containing objects which are indiscernible with respects to some features, which form basic building blocks of knowledge about reality and are employed to find out patterns in data. Attempt to successfully predicting the financial market has always attracted the attention of researchers, economist, mathematicians and bankers. The use of any computational intelligence method such as neural network, genetic algorithm, fuzzy sets and rough sets for banking system has been widely established. Rough Sets is used to generate the reducts that is the reduced set of significant attributes and core which is the most significant attribute of the dataset, which can not be removed from the dataset without affecting the classification.

2 Related Work

As a response to the financial crises in the recent decades there is a growing volume of literature that is largely devoted to analyzing the sources and effects of financial crises, and predicting and suggesting remedies for their prevention. There is a keen interest of researchers in testing and improving the accuracy of early warning systems for timely detection of prevention of the crisis.

Wheelock and Wilson [29] examine the factors that are believed to be relevant to predict bank failure. The analysis is carried out in terms of competing-risks hazard models. It is concluded that the more efficiently a bank operates, the less likely it is to fail. The possibility of bank failure is higher in respect of banks having lower capitalization, higher ratios of loans to assets, poor quality of loan portfolios and lower earnings.

Estrella and Park [9] examine the effectiveness of three capital ratios (the first based on leverage, the second on gross revenues, and the third on risk-weighted assets) for predicting bank failure. The study is based on 1988–1993 data pertaining to U.S. banks. The results show that over 1 or 2 year time-horizons the simple leverage and gross revenue ratios perform as well as the more complex risk-weighted ratio. But over longer periods the simple ratios are not only less costly to implement but also useful supplementary indicators of capital adequacy.

Canbas et al. [4] demonstrate that an Integrated Early Warning System (IEWS) can be employed to predict bank failure more accurately. The IEWS is conceived in terms of Discriminant Analysis (DA), Logit/Probit regression, and Principal Component Analysis (PCA). In order to test the predictive power of the IEWS, the authors use the data for 40 privately owned Turkish commercial banks. The results show that the IEWS has more predictive power relatively to the other techniques.

Kolari et al. [13] and Lanine and Rudi [14] are other notable studies that have attempted to develop an early warning system based on Logit and the Trait Recognition method The authors test the predictive ability of the two models in terms of their prediction accuracy. The Trait Recognition model is found to outperform the Logit model.

Tung et al. [28] explain and predict financial distress of banks using Generic Self-organizing Fuzzy Neural Network (GenSoFNN) based on the compositional rule of inference (CRI). The study is based on a population of 3,635 US banks observed over a 21 years period, 1980–2000. The authors have found the performance of their bank failure classification and EWS as encouraging.

Mannasoo and Mayes [16] employ logit technique with a five components CAMELS model and structural and macroeconomic factors to demonstrate that besides bank-specific factors, macroeconomic factors and institutional frameworks are also crucial factors responsible for bank distress in East European countries over the period 1996–2003.

Demirguc-Kunt et al. [7] analyze structural and other changes in the banking sector following a bank crisis. The authors observe that individuals and companies take away their funds from inefficient, weaker banks and invest the same in stronger banks. Among other consequences of bank crises, it is concluded that there is no marked decline in aggregate bank deposits relative to GDP of the country; aggregate credit declines; and banks tend to reallocate their assets away from loans and attempt to achieve cost efficiency.

Poghosyan and Cihak [22] analyze the causes of banking distress in Europe using the database relevant to financial distress of banks across the European Union from mid-1990s to 2008. Authors bring out the significance of early identification of financial fragility of banks as late identification renders solutions much more difficult and costly. They also identify indicators and thresholds to facilitate a distinction between sound banks and banks vulnerable to financial distress.

Demyanyk and Hasan [8] present a summary of findings as well as methodologies of empirical studies that attempted to explain and predict financial crises and suggest appropriate remedies in the back-drop of bank failures in the U.S. and many other countries in the recent decades.

Mariathasan and Merrouche [17] compare the relevance of alternative measures of capitalization for bank failure during the 2007–2010 crisis, and search for evidence of manipulated Basel risk-weights. Compared with the unweighted leverage ratio, the authors find the risk-weighted asset ratio to be a superior predictor of bank failure when banks operate under the Basel II regime and the crisis risk is low. When the crisis risk is high, the unweighted leverage ratio is found to be the more reliable predictor. However, when banks do not operate under Basel II rules, both ratios perform comparably, independent of a crisis risk. The authors also observe that banks in trouble tend to manipulate their key ratios so as to conceal the actual magnitude of their financial fragility.

Prasad and Ravinder [23] evaluate the financial performance of nationalized banks in India over a five-year period, 2006–2010. For ranking of these banks, the authors employ the CAMEL parameters assigning equal weights to its five components. Ranks are assigned on the basis of array of financial indices arranged in ascending/descending order.

Apart from traditional model artificial intelligence methods are also used for generating knowledge from empirical data. For predicting the financial crises rough set method along with neural network is widely used. The development and application of artificial intelligence led some researchers to employ inductive learning and neural networks in credit risk domain [31]. Ahn et al. [1] have combined the rough set methodology with neural networks to predict company failure. The rough set methodology has applied by Daubie et al. [6] for the classification of commercial loans. Segovia et al. [26] applied this technique to the prediction of insolvency in insurance companies, and [25] utilized it for the same purpose in a sample of small- and medium-sized enterprises (SMEs). Nursel et al. [18] predicts the bankruptcy of Turkish bank using rough set methodology. Reyes and Maria [24] applied this methodology for modeling credit risks. More applications of rough set theory in banking and stock market prediction can be found in [10–12, 30].

3 Methodology

The main objective of this chapter is to analyze accuracy aspect of CAMEL-based ranking of banks. The issue of financial soundness of a bank is particularly important from the point of view of its depositors. They invest their savings with banks having trust in their financial integrity and prudence. These savings constitute the basis of basic roles of a bank, namely, financial intermediation and asset transmutation. Information relating to financial soundness of a bank assumes importance for facilitating appropriate decision-choice by the depositors. Information must be accurate, as far as possible. The CAMEL ranking system is a widely relied source of information in this regard. It has been used under a variety of scenarios with regard to scope and computational methodology. There can be alternative perceptions as regard the inclusion and weighting of components and sub-components for constructing indices of financial health for various banks. Weighting is a sensitive issue; it must not leave any scope for rank-reversibility, that is, sensitivity of ranking to a change in weighting. Similarly, financial parameters included under various components must be relevant and basic to the purpose in question; due care should be exercised to ensure that no financial parameter is included that is, in fact, redundant. Likewise, choice of criterion for ranking of banks on the basis of various indices is another important methodological issue which needs to be carefully looked into, and which represents the analytic thrust of this paper.

It may be noted that the CAMEL methodology came into existence actually as EWS to ascertain the need for, and the extent of, regulatory intervention for timely detection and prevention of financial distress in the banking and other financial sectors. It provides for a rating scale from 1 (good) to 5 (bad) for performance assessment. An overall rating of 3–5 implies a signal to alert the regulators for necessary regulatory action. However, the use of CAMEL rating in the present study is for an entirely different purpose, namely, ranking banks so that depositors have information about relative financial performance of various banks.

For illustrative purpose, we have considered ten largest public sector banks on the basis of their deposit-base over a five-year period (2008–2009 to 2012–2013). The analysis of financial soundness of banks is structured under two parts. Part I is devoted to ranking of these banks on the basis of performance indices of their capital adequacy (C), asset quality (A), management efficiency (M), earnings (E) and liquidity (L). Relevant data have been drawn from two sources—RBI (MS-Excel) and Annual Balance Sheets of banks. Performance analysis has been carried out in terms of two alternative approaches so as to bring implications with regard to their rank accuracy. We prefer to term these approaches as *Unclassified Rank Assignment Approach* and *Classified Rank Assignment Approach.* Part II presents analysis of accuracy of ranks obtained by CAMEL model for both the approaches that is for Unclassified Rank Assignment Approach and for Classified Rank Assignment in terms of application of Rough Set Theory (RST). The output of CAMEL model (Ranking of banks for both approaches) is given as input to rough set for generating rules and for finding the reduct and core. The accuracy of the ranking generated by the CAMEL model is verified using lower and upper approximation.

3.1 Unclassified Rank Assignment Approach (URAA)

When ranks are assigned simply on the basis of array of financial indices arranged in ascending/descending order (Unclassified Rank Assignment Approach) some absurd outcomes may follow. Two such possibilities are discussed here. A major problem with rank assignment occurs when the ranks fail to represent the relativity among magnitudes of financial indices on which they are based. Consider the ranks, assigned to various banks on the basis of rates-of-return (on equity) achieved by them, as specified in Table 1 (col. 2.1). In this case, improvements in rate-of-return positions are not proportionally seen in the corresponding ranks. For instance, a difference of 1.83 in rates-of-return as between the bank at Sr. No. 8 (having a 11.76 % rate-of-return) and the bank at Sr. No. 7 (having a 9.93 % rate-of-return), which amounts to an increase of 18.43 %, induces only a one-step improvement in rank from 10 to 9. In contrast, difference between the bank at Sr. No. 9 (having

Table 1 Unclassified rank assignment approach

Bank Sr. No.	Financial performance (t = 1)		Financial performance (t = 2)		Remarks
	Return on equity (%)	Rank	Return on equity (%)	Rank	
1	2.1	2.2	3.1	3.2	4
1	15.12	8	12.10 (−20 %)	8	No change in rank despite substantial decline in earnings position
2	19.93	2	19.93	2	
3	21.03	1	21.03	1	
4	15.91	7	15.91	7	
5	18.27	3	18.27	4	
6	17.53	5	18.40 (+5 %)	3	Improvement in rank induced by a small increase in earnings position
7	9.93	10	9.93	10	
8	11.76	9	11.76	9	
9	17.65	4	17.65	5	
10	16.43	6	17.50 (+6.5 %)	6	No change in rank despite improvement in earnings position

Note rank 1 is assigned to the bank commanding the highest earnings position

7.65 % rate-of-return) and the bank at Sr. No. (having 15.91 % rate-of-return) is even lesser, 1.83 (which amounts to an increase of 10.94 %). Yet it leads to a much greater improvement in ranks, that is, a three-step improvement from rank 7 to rank 4. Other rank figures may also reveal similar anomalies. Apparently, this is an absurd outcome.

For accuracy of ranking, it is also necessary that overtime changes in financial performance of banks are duly reflected in the ranks assigned to them. Suppose in due course there is improvement in earnings position in case of one bank and deterioration in the case of another. If the ranks assigned to these banks are still found to be unchanged we may reasonably infer that ranking methodology does not provide for accuracy. It is also possible that in one case a relatively small increase in earnings of a bank may lead to improvement in its rank position, whereas a relatively substantial decline in the earnings of another bank may leave the rank unchanged. This is another instance of an absurd outcome. For illustration, consider the behavior of earnings of banks during two time-periods, t = 1 and t = 2 as depicted in Table 1.

3.2 Classified Rank Assignment Approach (CRAA)

For ensuring a fair degree of rank accuracy, we must have an approach that provides for a more or less systematic relationship between the behavior of ranks and financial indices on which the ranks are based. This may be expected under the *Classified Rank Assignment Approach*. Computation of ranks under this approach involves the following steps. Consider, for instance, col. 2.1 of Table 2:

1. There are in all 10 (N) indices relating to return on equity. Take the difference (D) between the maximum and the minimum indices, and divide it by 9 (that is N − 1) in order to classify the given indices under 10 class intervals each having the same class width, with minimum and maximum indices falling at mid point of the 1st and the 10th class intervals, respectively. In the present case, we have D = 21.03 − 9.93 = 11.1. Dividing D by 9 we have 11.1/9 = 1.233 as the width of each of the 10 class intervals.
2. In order to determine the first class interval, divide 1.233 by 2. We have 1.233/2 = 0.616 which is the difference between the mid value and lower/upper class limits. Accordingly, for the 1st class interval lower class limit is 9.31 (that is, 9.93–0.62), and the upper class limit is 10.54 (that is, 9.31 + 1.23). Limit values for the rest of the class intervals can be obtained by adding 1.23 to the lower limit (that is upper limit of the previous class interval) each time.

Class interval	9.31–10.54	10.54–11.77	11.77–13.00	13.00–14.23	14.23–15.46	15.46–16.69	16.69–17.92	17.92–19.15	19.15–20.38	20.38–21.68
Rank	10	9	8	7	6	5	4	3	2	1

Table 2 Classified rank assignment approach

Bank Sr. No.	Financial performance (t = 1)		Financial performance (t = 2)		Remarks
	Return on equity (%)	Rank	Return on equity (%)	Rank	
1	2.1	2.2	3.1	3.2	4
1	15.12	6	12.10 (−20 %)	8	Decline in earning position is duly reflected in rank deterioration
2	19.93	2	19.93	2	
3	21.03	1	21.03	1	
4	15.91	5	15.91	5	
5	18.27	3	18.27	3	
6	17.53	4	18.40 (+5 %)	3	Increase in earning position is duly reflected in rank improvement
7	9.93	10	9.93	10	
8	11.76	9	11.76	9	
9	17.65	4	17.65	4	
10	16.43	5	17.50 (+6.5 %)	4	Increase in earning position is duly reflected in rank improvement

3. Rank the class intervals in ascending/descending order:
4. Classify each of the given indices under relevant class intervals and assign it the corresponding rank as specified in Table 2.

4 Analysis of Financial Soundness of Banks

A considerable variety of choice is often seen in the literature as regard the choice of CAMEL components, and the financial ratios or indices within each component. There is no uniform approach as to how many and which components/indices should be included. The choice is mainly specific to the objective of the study and investigator's critical judgment. The objective of present study is to rank public sector banks in India on the basis of their financial soundness particularly in relation to the concerns of debt holders. The choice of financial indices has been motivated mainly by this consideration. We have considered ten largest Indian public sector banks on the basis of their deposit-base over a five-year period, 2008–2009 through 2012–2013 so as to ensure that these banks constitute more or less a homogeneous group in terms of ownership structure and regulatory and administrative control over their policies. For ranking of these banks, we have employed in all eleven financial indices relating to capital adequacy, asset quality, management efficiency, earnings and liquidity. Each of these financial indices has been specified by a quantity which represents the mean value over the five-year period, 2008–2009 through 2012–2013. The rankings of banks obtained under the two approaches, Unclassified Rank Assignment Approach (URAA) and Classified Rank Assignment Approach (CRAA), are specified in Table 3 through Table 8. For accuracy of ranks it is important to ensure that financial indices included under various components are relevant and basic to the purpose in question and no financial parameter is included which is redundant. For compliance of this requirement, ranks representing overall financial soundness of various banks were regressed on the eleven indices to identify if any of them needed to be excluded. This exercise revealed that Net NPA ratio (in relation to asset quality) and business per employee (in relation to management efficiency) fell under the category of 'excluded variables'. Accordingly, these indices were dropped while computing ranks.

Capital Adequacy: Bank's capital or equity provides protection to the depositors against its possible financial distress. A bank having a shrinking capital base relative to its assets is vulnerable to potential financial distress. This is the main reason that in the literature on analysis of financial distress of banks, this component appears as a critical factor practically in all empirical research. Bank equity ratio is defined in two broad contexts: risk-weighted capital ratio and non-risk-weighted capital ratio. In some of the studies the latter ratio is employed. It is argued that the risk-weighted ratios are open to manipulation. A bank while adjusting its assets for the associated risk may be tempted to apply weighting to different categories of risky assets that may help it conceal the real position with regard to its financial fragility. This possibility is recognized by a number of studies such as [5, 15]. In view of this possibility studies such as [22, 27] employ non-risk-weighted capital ratios. The Indian banking sector is under stringent regulatory regime of the RBI. Possibilities of such manipulations are believed to be practically non-existent. Accordingly, we have employed the risk-adjusted capital ratio, CRAR. It represents bank's qualifying capital as a proportion of risk adjusted (or weighted) assets. The RBI has set the minimum capital adequacy ratio at 9 % for all banks to ensure that banks do not expand their business without having adequate capital. A ratio below the minimum indicates that the bank is not adequately capitalized to expand its operations. As a measure of bank's capital adequacy, we have supplemented the CRAR by including another ratio, namely, D/E ratio (deposit to equity ratio). The capital adequacy of different banks and their rank position based on the same are specified in Table 3.

Asset Quality: The inferior quality of bank assets accentuates its vulnerability to financial distress. Losses resulting from such assets eat into the capital base of the bank and become one of the main causes of bank failure. Since the predominant business of a commercial bank is lending, loan quality is an important factor in the context of its financial soundness. A relatively dependable measure of overall loan quality of a bank is the Net NPA ratio which represents non-performing assets as a proportion of its total loans (advances). According to the RBI, NPAs are those assets for which interest is overdue for more than 90 days (or 3 months). For assessing asset quality besides the Net NPA ratio we have included another ratio, as well, that is, the ratio of government and other approved securities to total assets as

Table 3 Capital adequacy

Name of bank	CRAR[a]	Ranking under alternative approaches		D/E	Ranking under alternative approaches		Average rank		Capital adequacy rank R (C)	
		URAA	CRAA		URAA	CRAA	URAA	CRAA	URAA	CRAA
State Bank of India	13.28	4	5	12.79	1	1	2.5	3	2	2
Bank of Baroda	14.18	1	1	14.85	2	4	1.5	2.5	1	1
Punjab National Bank	13.19	5	5	16.18	7	5	6	5	4	4
Bank of India	12.22	9	10	15.73	6	5	7.5	7.5	6	8
Canara Bank	13.81	2	3	14.93	3	4	2.5	3.5	2	2
Union Bank of India	12.41	8	9	15.70	5	5	6.5	7	5	7
Central Bank of India	12.18	10	10	17.71	8	7	9	8.5	8	9
Indian Overseas Bank	13.54	3	4	14.85	2	4	2.5	4	2	3
Syndicate Bank	12.65	7	8	19.64	9	10	8	9	7	10
Allahabad Bank	12.71	6	8	15.33	4	4	5	6	3	6

[a] CRAR in Table 3 specifies Risk-adjusted capital ratio

these are believed to be nearly risk-free assets. The greater the proportion of theses securities, the greater the provision for depositors concerns. The asset quality of different banks and their rank position based on the same are specified in Table 4.

Management Efficiency: The success of any institution depends largely on competence and performance of its management. The more vigilant and capable the management, the greater the bank's financial soundness. There is no doubt about these assertions. But the problem is with regard to identifying and measuring the effect of management on bank's financial performance. There is no direct and definite measure in this regard. Management influences can be gauged in many ways, for instance, asset quality or return on assets, business expansion and optimal utilization of bank's skills and material resources. In the present study we have attempted to capture the influence of management quality in terms three indices, namely, return on advances (adjusted for cost of funds), business per employee and profit per employee. Table 5 presents relevant information in this regard together with the ranks assigned to various banks on this count.

Earnings: Earnings and profits are important for strengthening capital base of a bank and preventing any financial difficulty. There should be not only high profits but also sustainable profits. In empirical research—for instance, [2, 3, 22, 27] a wide variety of indicators of earnings have been employed, but the relatively more popular measures are rate-of-return on assets (ROA) and rate-of-return on equity (ROE). We have used these indicators to represent earnings and profits of the banks (Table 6).

Liquidity: An essential condition for preserving confidence of depositors in financial ability of the bank is that the latter must have sufficient liquidity for discharging their short-term obligations and unexpected withdrawals. If it is not so, there is a catastrophic effect of loss of confidence in the financial soundness of the bank. The confidence loss may cumulate to even lead to financial distress of the bank [19]. In order to capture the liquidity position of a bank, we have employed two ratios, namely, ratio of liquid assets to demand deposits (LA/DD) and ratio of government and other approved securities to demand deposits (G. Sec./DD). The relevant statistics and results are specified in Table 7.

4.1 Ranking of Overall Financial Soundness of Banks

We have pooled the information for each bank as regard its relative position in respect of the five CAMEL indicators so as to assign to it an overall rank representing its financial soundness relatively to that of other banks. The overall rank is represented by the simple mean value of the ranks assigned to various indices included under the five broad financial indicators. These indicators do not necessarily imply the same significance as regard the financial soundness of a bank; significance may vary depending on the context in which these are viewed as well

Table 4 Asset quality

Name of bank	Net NPA ratio	Ranking under alternative approaches		G.Sec./TA (%)	Ranking under alternative approaches		Average rank		Asset quality rank R (A)	
		URAA	CRAA		URAA	CRAA	URAA	CRAA	URAA	CRAA
State Bank of India	1.81	Excluded variable	Excluded variable	20.26	7	6	8.5	6	7	6
Bank of Baroda	0.56			17.67	10	10	5.5	10	4	10
Punjab National Bank	1.08			22.03	5	3	4.0	3	2	3
Bank of India	1.24			19.73	9	7	7.0	7	6	7
Canara Bank	1.38			23.39	3	1	5.0	1	3	1
Union Bank of India	1.13			20.52	6	6	5.0	6	3	6
Central Bank of India	1.71			23.56	2	1	5.0	1	3	1
Indian Overseas Bank	1.78			23.12	4	1	6.5	1	5	1
Syndicate Bank	0.91			20.06	8	6	5.0	6	3	6
Allahabad Bank	1.27			23.64	1	1	3.5	1	1	1

Table 5 Management efficiency

Name of bank	Return on advances adjusted to COF (%)	Ranking under alternative approaches		Business per employee (Rs. in million)	Ranking under alternative approaches		Profit per employee (Rs. in million)	Ranking under alternative approaches		Average rank		Management efficiency rank R (M)	
		URAA	CRAA		URAA	CRAA		URAA	CRAA	URAA	CRAA	URAA	CRAA
State Bank of India	3.97	4	7	72.78	Excluded variable	Excluded variable	0.50	7	7	7.00	7.00	7	7
Bank of Baroda	3.61	8	9	125.58			0.94	1	1	3.33	5.0	1	5
Punjab National Bank	4.84	1	1	95.56			0.76	2	3	3.33	2.0	1	1
Bank of India	3.43	10	10	121.4			0.62	5	5	5.67	7.5	5	8
Canara Bank	3.57	9	9	115.13			0.75	3	3	5.00	6.00	4	6
Union Bank of India	3.89	5	7	97.5			0.70	4	4	4.67	5.5	3	5
Central Bank of India	3.69	7	8	78.84			0.27	10	10	8.67	9.00	8	10
Indian Overseas Bank	3.79	6	8	97.41			0.36	9	9	7.00	8.5	7	9
Syndicate Bank	4.23	3	5	94.08			0.48	8	7	6.33	6.00	6	6
Allahabad Bank	4.39	2	4	104.08			0.60	6	6	4.00	5.00	2	5

Table 6 Earnings

Name of bank	Return on equity (%)	Ranking under alternative approaches		Return on assets (%)	Ranking under alternative approaches		Average rank		Earnings rank R (E)	
		URAA	CRAA		URAA	CRAA	URAA	CRAA	URAA	CRAA
State Bank of India	15.12	8	6	0.88	5	5	6.5	5.5	6	5
Bank of Baroda	19.93	2	2	1.15	1	1	1.5	1.5	1	1
Punjab National Bank	21.03	1	1	0.86	7	5	4.0	3.0	3	3
Bank of India	15.91	7	5	0.88	6	5	6.5	5.0	6	5
Canara Bank	18.27	3	3	1.10	2	2	2.5	2.5	2	2
Union Bank of India	17.53	5	4	1.03	3	3	4.0	3.5	3	3
Central Bank of India	9.93	10	10	0.50	10	10	10.0	10.0	8	10
Indian Overseas Bank	11.76	9	9	0.63	9	8	9.0	8.5	7	8
Syndicate Bank	17.65	4	4	0.81	8	6	6.0	5.0	5	5
Allahabad Bank	16.43	6	5	0.97	4	3	5.0	4.0	4	4

Table 7 Liquidity

Name of bank	LA/DD[a] (%)	Ranking under alternative approaches		G.Sec./DD (%)	Ranking under alternative approaches		Average rank		Liquidity rank R (L)	
		URAA	CRAA		URAA	CRAA	URAA	CRAA	URAA	CRAA
State Bank of India	23.66	8	8	58.14	10	10	9	9	7	10
Bank of Baroda	40.53	1	1	73.72	8	7	4.5	4.0	4	2
Punjab National Bank	19.23	10	10	68.85	9	8	9.5	9.0	8	10
Bank of India	28.17	4	6	88.94	3	3	3.5	4.5	2	3
Canara Bank	28.48	3	6	101.84	1	1	2	3.5	1	1
Union Bank of India	22.18	9	9	76.86	7	6	8	7.5	6	7
Central Bank of India	24.78	6	8	80.69	5	5	5.5	6.5	5	6
Indian Overseas Bank	26.47	5	7	96.61	2	2	3.5	4.5	2	3
Syndicate Bank	30.51	2	5	78.90	6	6	4	5.5	3	4
Allahabad Bank	23.87	7	8	83.00	4	5	5.5	6.5	5	6

[a] LA/DD in Table 7 specifies Ratio of liquid assets to demand deposits (LA/DD)

as the judgment of the investigator who assigns weighting to them. We have refrained from assigning weighting due mainly to avoid any arbitrariness in this regard. As such, the overall ranks specified in Table 8 are 'indicative' only in the specific context of the present study. The information specified in the table shows that in case of some of the banks the ranks assigned under the two approaches differ which is due mainly to the questionable treatment of differences in performance indices of various banks under the Unclassified Rank Assignment Approach. On the scale of financial soundness, the Canara Bank stands at the top with an overall rank of 1, followed by the Bank of Baroda with rank of 2.

5 Analysis of Ranking Generated by CAMEL Model Using Rough Sets

Since the year of its inception in 1982, rough set theory has been extensively used as an effective data mining and knowledge discovery technique in numerous applications in the finance, investment and banking fields. Rough set theory is a way of representing and reasoning imprecision and uncertain information in data [21]. This theory is basically revolves around the concept of Indiscernibility which means the inability to distinguish between objects or objects which are similar under the given information. Rough set theory deals with the approximation of sets constructed from empirical data. This is most helpful when trying to discover decision rules, important features, and minimization of conditional attributes. There are four important concepts to discuss when talking about rough set theory: information systems, Indiscernibility, reduction of attributes and rule generation.

In the Rough Sets Theory, information systems are used to represent knowledge. The notion of an information system presented here is described in Pawlak [20]. Suppose we are given two finite, non-empty sets U and A, where U is the *universe*, and A, a set *attributes*. With every attribute $a \in A$ we associate a set Va, of its *values*, called the *domain* of a. The pair $S = (U, A)$ will be called a *database* or *information system*. Any subset B of A determines a binary relation $I(B)$ on U, which will be called an *Indiscernibility relation*, and is defined as $(x, y) \in I(B)$ if and only if $a(x) = a(y)$ for every $a \in A$, where $a(x)$ denotes the value of attribute a for element x. The Indiscernibility relation will be used next to define two basic operations in rough set theory, which are defined below:

- The set of all objects which can be with *certainty* classified as members of X with respect to R is called the *R-lower approximation* of a set X with respect to R, and denoted by

$$R(\underline{X}) = \{x \in U : R(x) \subseteq X\}$$

Table 8 Comparative analysis of ranks under the two approaches

Name of bank	Ranks assigned under alternative approaches													
	Capital adequacy (Table 3)				Asset quality (Table 4)		Management efficiency (Table 5)				Earnings (Table 6)			
	CRAR		D/E		G. Sec./TA[a]		ROA[b]		PPE		ROE		ROA[c]	
	URAA	CRAA	URAA	CRAA	URAA	CRAA	URAA	CRAA	URAA	CRAA	URAA	CRAA	URAA	CRAA
SBI	4	5	1	1	7	6	4	7	7	7	8	6	5	5
BOB	1	1	2	4	10	10	8	9	1	1	2	2	1	1
PNB	5	5	7	5	5	3	1	1	2	3	1	1	7	5
BOI	9	10	6	5	9	7	10	10	5	5	7	5	6	5
CB	2	3	3	4	3	1	9	9	3	3	3	3	2	2
UBI	8	9	5	5	6	6	5	7	4	4	5	4	3	3
CBI	10	10	8	7	2	1	7	8	10	10	10	10	10	10
IOB	3	4	2	4	4	1	6	8	9	9	9	9	9	8
SB	7	8	9	10	8	6	3	5	8	7	4	4	8	6
AB	6	8	4	4	1	1	2	4	6	6	6	5	4	3

Name of bank	Liquidity (Table 7)				URAA		CRAA	
	LA/DD		G. Sec/DD		Rank average	Overall rank	Rank average	Overall rank
	URAA	CRAA	URAA	CRAA				
SBI	8	8	10	10	**6.00**	7	**6.11**	7
BOB	1	1	8	7	**3.78**	2	**4.00**	2
PNB	10	10	9	8	**5.22**	4	**4.55**	3
BOI	4	6	3	3	**6.55**	9	**6.22**	7
CB	3	6	1	1	**3.22**	1	**3.55**	1
UBI	9	9	7	6	**5.78**	6	**5.89**	6
CBI	6	8	5	5	**7.55**	10	**7.67**	10
IOB	5	7	2	2	**5.44**	5	**5.78**	6
SB	2	5	6	6	**6.11**	8	**6.33**	7
AB	7	8	4	5	**4.44**	3	**4.89**	4

[a] *TA* specifies total assets

[b] *ROA* specifies return on *advances*

[c] *ROA* specifies return on *assets*

The ranks obtained by Unclassified Rank Assighment Approach (URAA) and Classified Rank Assignment Approach (CRAA) has been made bold so that they can be compared side by side

- The set of all objects which can be only classified as *possible* members of *X* with respect to *R* is called the *R-upper approximation* of a set *X* with respect to *R*, and denoted by

$$R(\overline{X}) = \{x \in U : R(x) \cap X \neq \varphi\}$$

The set of all objects which can be decisively classified neither as members of *X* nor as members of—*X* with respect to *R* is called the *boundary region* of a set *X* with respect to *R*, and denoted by *RN* (*X*) *R,* i.e.

$$RN(X)R = R(\overline{X}) - R(\underline{X})$$

A set *X* is called *crisp* (*exact*) with respect to *R* if and only if the boundary region of *X* is empty.

A set *X* is called *rough* (*inexact*) with respect to *R* if and only if the boundary region of *X* is nonempty.

Let C, $D \subseteq A$, be sets of condition and decision attributes, respectively. We will say that $C' \subseteq C$ is a *D-reduct* (reduct with *respect* to *D*) of *C*, if C' is a minimal subset of *C* such that γ (*C*, *D*) = γ (C', *D*).

Now we define a notion of a core of attributes. Let B be a subset of A. The core of B is a set of all indispensable attributes of B. The following is an important property, connecting the notion of the core and reducts

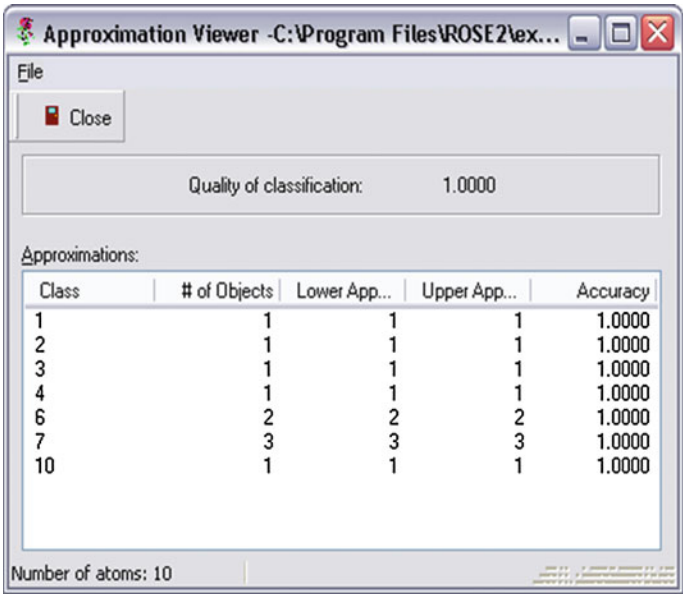
Core(B) = ∩ Red(B),
where Red(B) is the set off all reducts of B.

Table 8 given above is divided into two tables one for Unclassified Rank assignment approach and other for Classified Rank Assignment Approach. Both these tables are given as input to rose2 software for analysis of rough set.

For classified CAMEL approach there is no attributes in core set which means that there is no indispensable attribute and any single attribute in such an information system can be deleted without altering the equivalence-class structure. There are two reducts which are found by Heuristic methods

Core = {}
Reduct1 = {CRAR, PPE}
Reduct2 = {D/E, PPE}

The classification accuracy of ranks as given by lower and upper approximation for classified CAMEL Rank approach is shown in Fig. 1.



Quality of classification: 1.0000

Approximations:

Class	# of Objects	Lower App...	Upper App...	Accuracy
1	1	1	1	1.0000
2	1	1	1	1.0000
3	1	1	1	1.0000
4	1	1	1	1.0000
6	2	2	2	1.0000
7	3	3	3	1.0000
10	1	1	1	1.0000

Number of atoms: 10

Fig. 1 *Lower* and *upper* approximation for classified rank assignment approach

And rules are

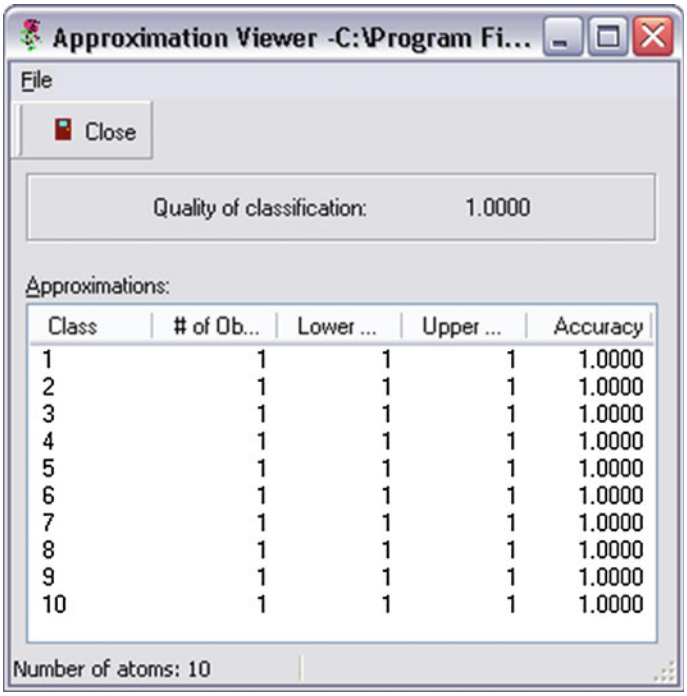
```
#       LEM2
#       C:\Program Files\ROSE2\examples\classified camel.isf
#       objects = 10
#       attributes = 10
#       decision = CRAA
#       classes = {1, 2, 3, 4, 6, 7, 10}
#       Wed Apr 02 11:37:38 2014
#       0 s

rule 1. (CRAR = 3) => (CRAA = 1); [1, 1, 100.00%, 100.00%][1, 0, 0, 0, 0, 0, 0]
[{5}, {}, {}, {}, {}, {}, {}]
rule 2. (CRAR = 1) => (CRAA = 2); [1, 1, 100.00%, 100.00%][0, 1, 0, 0, 0, 0, 0]
[{}, {2}, {}, {}, {}, {}, {}]
rule 3. (GTA = 3) => (CRAA = 3); [1, 1, 100.00%, 100.00%][0, 0, 1, 0, 0, 0, 0]
[{}, {}, {3}, {}, {}, {}, {}]
rule 4. (ROA = 4) => (CRAA = 4); [1, 1, 100.00%, 100.00%][0, 0, 0, 1, 0, 0, 0]
[{}, {}, {}, {10}, {}, {}, {}]
rule 5. (CRAR = 9) => (CRAA = 6); [1, 1, 50.00%, 100.00%][0, 0, 0, 0, 1, 0, 0]
[{}, {}, {}, {}, {6}, {}, {}]
rule 6. (CRAR = 4) => (CRAA = 6); [1, 1, 50.00%, 100.00%][0, 0, 0, 0, 1, 0, 0]
[{}, {}, {}, {}, {8}, {}, {}]
rule 7. (PPE = 7) => (CRAA = 7); [2, 2, 66.67%, 100.00%][0, 0, 0, 0, 0, 2, 0]
[{}, {}, {}, {}, {}, {1, 9}, {}]
rule 8. (GTA = 7) => (CRAA = 7); [1, 1, 33.33%, 100.00%][0, 0, 0, 0, 0, 1, 0]
[{}, {}, {}, {}, {}, {4}, {}]
rule 9. (D/E = 7) => (CRAA = 10); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 0, 1]
[{}, {}, {}, {}, {}, {}, {7}]
**END
```

For Unclassified Rank Assignment approach there is no elements in the core set and there are two reducts set

Core = { }
Reduct1 = {CRAR}
Reduct2 = {GTA}

The classification accuracy of ranks as given by lower and upper approximation for Unclassified Rank Assignment Approach is shown in Fig. 2.



Quality of classification: 1.0000

Approximations:

Class	# of Ob...	Lower ...	Upper ...	Accuracy
1	1	1	1	1.0000
2	1	1	1	1.0000
3	1	1	1	1.0000
4	1	1	1	1.0000
5	1	1	1	1.0000
6	1	1	1	1.0000
7	1	1	1	1.0000
8	1	1	1	1.0000
9	1	1	1	1.0000
10	1	1	1	1.0000

Number of atoms: 10

Fig. 2 *Lower* and *upper* approximation of unclassified rank assignment approach

And the rules generated for Unclassified Rank Assignment Approach are

```
#         LEM2
#         C:\Program Files\ROSE2\examples\unclassified camel.isf
#         objects = 10
#         attributes = 10
#         decision = URAA
#         classes = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
#         Wed Apr 02 11:43:22 2014
#         0 s


rule 1. (CRAR = 2) => (URAA = 1); [1, 1, 100.00%, 100.00%][1, 0, 0, 0, 0, 0, 0,
0, 0, 0]
[{5}, {}, {}, {}, {}, {}, {}, {}, {}, {}]
rule 2. (CRAR = 1) => (URAA = 2); [1, 1, 100.00%, 100.00%][0, 1, 0, 0, 0, 0, 0,
0, 0, 0]
[{}, {2}, {}, {}, {}, {}, {}, {}, {}, {}]
rule 3. (CRAR = 6) => (URAA = 3); [1, 1, 100.00%, 100.00%][0, 0, 1, 0, 0, 0, 0,
0, 0, 0]
[{}, {}, {10}, {}, {}, {}, {}, {}, {}, {}]
rule 4. (CRAR = 5) => (URAA = 4); [1, 1, 100.00%, 100.00%][0, 0, 0, 1, 0, 0, 0,
0, 0, 0]
[{}, {}, {}, {3}, {}, {}, {}, {}, {}, {}]
rule 5. (CRAR = 3) => (URAA = 5); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 1, 0, 0,
0, 0, 0]
[{}, {}, {}, {}, {8}, {}, {}, {}, {}, {}]
rule 6. (CRAR = 8) => (URAA = 6); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 1, 0,
0, 0, 0]
[{}, {}, {}, {}, {}, {6}, {}, {}, {}, {}]
rule 7. (CRAR = 4) => (URAA = 7); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 0, 1,
0, 0, 0]
[{}, {}, {}, {}, {}, {}, {1}, {}, {}, {}]
rule 8. (CRAR = 7) => (URAA = 8); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 0, 0,
1, 0, 0]
[{}, {}, {}, {}, {}, {}, {}, {9}, {}, {}]
rule 9. (CRAR = 9) => (URAA = 9); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 0, 0,
0, 1, 0]
[{}, {}, {}, {}, {}, {}, {}, {}, {4}, {}]
rule 10. (CRAR = 10) => (URAA = 10); [1, 1, 100.00%, 100.00%][0, 0, 0, 0, 0, 0,
0, 0, 0, 1]
[{}, {}, {}, {}, {}, {}, {}, {}, {}, {7}]
**END
```

6 Result and Discussions

The objective of present study is to rank public sector banks in India on the basis of their financial soundness particularly in relation to the concerns of debt holders. The choice of financial indices has been motivated mainly by this consideration.

Ten largest Indian public sector banks on the basis of their deposit-base over a five-year period, 2008–2009 through 2012–2013 have been considered so as to ensure that these banks constitute more or less a homogeneous group in terms of ownership structure and regulatory and administrative control over their policies. For ranking of these banks, we have employed in all eleven financial indices relating to capital adequacy, asset quality, management efficiency, earnings and liquidity. Each of these financial indices has been specified by a quantity which represents the mean value over the five-year period, 2008–2009 through 2012–2013. The rankings of banks obtained under the two approaches, Unclassified Rank Assignment Approach (URAA) and Classified Rank Assignment Approach (CRAA), are specified. For accuracy of ranks it is important to ensure that financial indices included under various components are relevant and basic to the purpose in question and no financial parameter is included which is redundant. For compliance of this requirement, ranks representing overall financial soundness of various banks were regressed on the eleven indices to identify if any of them needed to be excluded. This exercise revealed that Net NPA ratio (in relation to asset quality) and business per employee (in relation to management efficiency) fell under the category of 'excluded variables'. Accordingly, these indices were dropped while computing ranks. Various public sector banks are ranked from 1–10 under classified and unclassified rank assignment approach and the ranks obtained by these two approaches are almost same. Ranks of banks obtained by these two approaches are given as input to the rough set for checking the accuracy of ranks by lower and upper approximation. Since for both the approaches there is no element in the core or core is empty this means that which means that there is no indispensable attribute any single attribute in such an information system can be deleted without altering the equivalence-class structure. In such cases, there is no *essential* or necessary attribute which is required for the class structure to be represented. There are attributes in the information systems (attribute-value table) which are more important to the knowledge represented in the equivalence class structure than other attributes. Often, there is a subset of attributes which can, by itself, fully characterize the knowledge in the database; such an attribute set is called a *reduct*. A reduct can be thought of as a *sufficient* set of features—sufficient, that is, to represent the category structure. For both approaches we have risk-adjusted capital ratio CRAR under capital adequacy attribute as one of the element of reduct. This means that CRAR alone is self sufficient to define the ranking of different banks. On the basis of the values of CRAR ranks of different banks can be generated as shown in the rules generated for unclassified rank assignment approach. Ranks of different banks as generated by Classified and Unclassified Rank approach are 100 % accurate as shown in Figs. 1 and 2 by lower and upper approximation. On the scale of financial soundness, the Canara Bank stands at the top with an overall rank of 1, followed by the Bank of Baroda with rank of 2 whereas Central Bank of India is at the bottom with rank 10.

7 Conclusion

Banking supervision has assumed enormous significance after 2008 global financial meltdown which has caused a wide spread distress and chaos among various constituents of financial system across the globe. Subsequently, an overhaul of banking supervision and regulatory practices took place which put special emphasis on supervisory tools like CAMEL model with certain reformulations. Another, parallel but related development in banking supervision is Basel-III norms which have been framed by the Basel Committee on Banking Supervision, a global forum of Central Bankers of major countries for effective banking supervision. Basel-III has also put maximum focus on risk based capital of banks which is also the findings of our analysis as Capital to Risk weighted Asset Ratio (CRAR) appears as single most important element as reduct of RST analysis. This research has applied RST approach to CAMEL model and future research may use RST analysis for Basel-III norms as well. It will be an interesting exercise for the researcher to find out the authenticity and effectiveness of CAMEL model in combination with Basel-III norms.

References

1. Ahn, B.S., Cho, S.S., Kim, C.Y.: The integrated methodology of rough set theory and artificial neural network for business failure prediction. Expert Syst. Appl. **18**(2), 65–74 (2000)
2. Arena, M.: Bank failures and bank fundamentals: A comparative analysis of Latin America and East Asia during the nineties using bank-level data. J. Bank. Finan. **32**(2), 299–310 (2008)
3. Avkiran, N.K., Cai, L.C.: Predicting Bank Financial Distress Prior to Crises, Working Paper. The University of Queensland, Australia (2012)
4. Canbas, S., Cabuk, A., Kilic, S.B.: Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case. Eur. J. Oper. Res. **166**(2), 528–546 (2005)
5. Das, S., Sy, A.N.R.: How Risky Are Banks' Risk Weighted Assets? Evidence from the Financial Crisis, IMF Working Paper, 12/36 (2012)
6. Daubie, M., Leveck, P., Meskens, N.: A comparison of the rough sets and recursive partitioning induction approaches: An application to commercial loans. Int. Trans. Oper. Res. **9**, 681–694 (2002)
7. Demirguc-Kunt, A., Detragiache, E., Gupta, P.: Inside the crisis: An empirical analysis of banking systems in distress. J. Int. Money Finan. **25**(5), 702–718 (2006)
8. Demyanyk, Y., Hasan, I.: Financial crises and bank failures: A review of prediction method. OMEGA **38**(5), 315–324 (2010)
9. Estrella, A., Park, S.: Capital ratios as predictors of bank failure. Econ. Policy Rev. **6**(2), 33–52 (2000)
10. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multicriteria decision analysis. Eur. J. Oper. Res. **129**, 1–47 (2001)
11. Hassanien, A.Q., Zamoon, S., Hassanien, A.E., Abrahm, A.: Rough set generating prediction rules for stock price movement. In: Computer Modeling and Simulation, EMS '08. Second UKSIM European Symposium, pp. 111–116 (2008)

12. Khoza, M., Marwala, T.: A rough set theory based predictive model for stock prices. In: Proceeding of IEEE 12th International Symposium on Computational Intelligence and Informatics, pp. 57–62. Budapest (2011)
13. Kolari, J., Glennon, D., Shin, H., Caputo, M.: Predicting large US commercial bank failures. J. Econ. Bus. **54**(4), 361–387 (2002)
14. Lanine, G., Rudi, V.V.: Failure predictions in the Russian bank sector with logit and trait recognition models. Expert Syst. Appl. **30**(3), 463–478 (2006)
15. Le Lesle, V., Avramova, S.: Revisiting Risk-Weighted Assets, IMF Working Paper, 12/90 (2012)
16. Mannasoo, K., Mayes, D.G.: Investigating the Early Signals of Banking Sector Vulnerabilities in Central and East European Emerging Markets, Working Paper of Eesti Pank, p. 8 (2005)
17. Mariathasan, M., Merrouche, O.: The Manipulation of Basel Risk-Weights. Evidence from 2007–2010. University of Oxford, Department of Economics, Discussion Paper, p. 621 (2012)
18. Nursel, S.R., Fahri, U., Bahadtin, R.: Predicting bankruptcies using rough set approach: The case of Turkish bank. In: Proceeding of American Conference on Applied Mathematics (Math ′08), Harvard, Massachusetts, USA, 24–26 Mar 2008
19. Ooghe, H., Prijcker S.D.: Failure Processes and Causes of Company Bankruptcy: A Typology, Working Paper, Steunpunt OOI (2006)
20. Pawlak, Z.: Rough set approach to knowledge-based decision support. Eur. J. Oper. Res. **99**, 48–57 (1997)
21. Pawlak, Z.: Rough sets. Int. J. Comput. Int. Sci. **11**(3), 341–356 (1982)
22. Poghosyan, T., Cihák, M.: Distress in European Banks: An Analysis Based on a New Dataset. IMF Working Paper, 09/9 (2009)
23. Prasad, K.V.N., Ravinder, G.: A camel model analysis of nationalized banks in India. Int. J. Trade Commer. **1**(1), 23–33 (2012)
24. Reyes, S.M., Maria, J.V.: Modeling credit risk: An application of the rough set methodology. Int. J. Bank. Finan. **10**(1), 34–56 (2013)
25. Rodriguez, M., Díaz, F.: La teoría de los rough sets y la predicción del fracaso empresarial. Diseño de un modelo para pymes, Revista de la Asociación Española de Contabilidad y Administración de Empresas **74**, 36–39 (2005)
26. Segovia, M.J., Gil, J.A., Vilar, L., Heras, A.J.: La metodología rough set frente al análisis discriminante en la predicción de insolvencia en empresas aseguradoras. Anales del Instituto de Actuarios Españoles 9 (2003)
27. Tatom, J., Houston, R.: Predicting Failure in the Commercial Banking Industry. Networks Financial Institute at Indiana State University. Working Paper, p. 27 (2011)
28. Tung, W.L., Quek, C., Cheng, P.: Genso-Ews: A novel neural-fuzzy based early warning system for predicting bank failures. Neural Netw. **17**(4), 567–587 (2004)
29. Wheelock, D.C., Wilson, P.W.: Why do banks disappear? The determinants of U.S. bank failures and acquisitions. Rev. Econ. Stat. **82**(1), 127–138 (2000)
30. Xu, J.N., Xi, B.: AHP-ANN based credit risk assessment for commercial banks. J. Harbin Univ. Sci. Technol. **6**, 94–98 (2002)
31. Yu, G.A., Xu, H.B.: Design and implementation of an expert system of loan risk evaluation. Comput. Eng. Sci. **10**, 104–106 (2004)

Towards Intelligent Distributed Computing: Cell-Oriented Computing

Ahmad Karawash, Hamid Mcheick and Mohamed Dbouk

Abstract Distributed computing systems are of huge importance in a number of recently established and future functions in computer science. For example, they are vital to banking applications, communication of electronic systems, air traffic control, manufacturing automation, biomedical operation works, space monitoring systems and robotics information systems. As the nature of computing comes to be increasingly directed towards intelligence and autonomy, intelligent computations will be the key for all future applications. Intelligent distributed computing will become the base for the growth of an innovative generation of intelligent distributed systems. Nowadays, research centres require the development of architectures of intelligent and collaborated systems; these systems must be capable of solving problems by themselves to save processing time and reduce costs. Building an intelligent style of distributed computing that controls the whole distributed system requires communications that must be based on a completely consistent system. The model of the ideal system to be adopted in building an intelligent distributed computing structure is the human body system, specifically the body's cells. As an artificial and virtual simulation of the high degree of intelligence that controls the body's cells, this chapter proposes a Cell-Oriented Computing model as a solution to accomplish the desired Intelligent Distributed Computing system.

Keywords Distributed computing · Intelligence · Cell theory

A. Karawash (✉) · H. Mcheick
Department of Computer Science, University of Quebec at Chicoutimi (UQAC),
555 Boulevard de l'Université Chicoutimi, Chicoutimi G7H2B1, Canada
e-mail: ahmad.karawash1@uqac.ca

H. Mcheick
e-mail: hamid_mcheick@uqac.ca

A. Karawash · M. Dbouk
Department of Computer Science, Ecole Doctorale des Sciences et de Technologie (EDST),
Université Libanaise, Hadath-Beirut, Lebanon
e-mail: mdbouk@ul.edu.lb

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_2

1 Introduction

Distributed computing (DC) is the consequence of permanent learning, the improvement of experience and the progress of computing knowledge. It offers advantages in its potential for improving availability and reliability through replication; performance through parallelism; sharing and interoperability through interconnection; and flexibility and scalability through modularity. It aims to identify the distributable components and their mutual interactions that together fulfil the system's requirements. In order to achieve DC goals, the client/server model is undoubtedly the most consolidated and regularly applied paradigm. With the extensive deployment of DC, the management, interoperability and integration of these systems have become challenging problems. Investigators have researched and developed important technologies to cope with these problems. One of the results of the continuous evolution of DC in the last decade is the Service-Oriented Computing (SOC) paradigm, which offers an evolution of the internet-standards based DC model, an evolution in processes of architecting, design and implementation, as well as in deploying e-business and integration solutions. The other key result is the Mobile Agent Computing (MAC) paradigm, which provides an alternative computing paradigm to the traditional client-server paradigm. Moreover, the latest DC technology is expressed by cloud computing, which evolved from grid computing and provides on-demand resource provisioning. Grid computing connects disparate computers to form one large infrastructure, harnessing unused resources.

Trends in the future of the Web require building intelligence into DC; consequently the goal of future research is the Intelligent Distributed Computing (IDC). The emergent field of IDC focuses on the development of a new generation of intelligent distributed systems. *IDC* covers a combination of methods and techniques derived from classical artificial intelligence, computational intelligence and multi-agent systems. The field of DC predicts the development of methods and technology to build systems that are composed of collaborating components.

Building a smart distributed model that controls the whole of Web communications needs to be based on an extremely consistent system. The ideal system that can be adopted in building IDC is the model of the human body system, specifically the body cell. Based on the high degree of intelligence that controls body cells, this chapter proposes a Cell-Oriented Computing (COC) methodology. COC is an artificial simulation of human cell functions that is proposed as a solution to achieve the desired intelligent distributed computing.

All parts of the human body are made up of cells. There is no such thing as a typical cell. Our bodies are composed of different kinds of cells. The diverse types of cells have different, specialized jobs to do. Cell computing simulates the human cell functions in the distributed systems environment. In fact, there are approximately 10 trillion cells in the human body [1]. Cells are the basic structural and functional units of the human body. Each cell has a specialized function and works in collaboration with other cells to perform a job. The cell acts like a mini computer. It is composed of a decision centre (the nucleus), the protein industry (mitochondria), store of human traits (genes) and a defence system (cell membrane). All cells in the body are connected to a giant computer called Intelligence that controls their tasks. The human intelligence works like a super-computer. Indeed, the human cell network is millions of times larger than the communication networks of the whole Web. Each cell has a great capacity to receive and transmit information to every cell in the body; each remembers the past for several generations, stores all the impressions of past and present human lives in its data banks and also evaluates and records possibilities for the future. It has an internal defence system to face intruders when an external attack occurs.

This chapter is arranged as follows: Sect. 2 discusses previous work on intelligent distributed computing, Sect. 3 introduces the Cell computing methodology, Sect. 4 discusses some definitions relating to the proposed model, Sect. 5 shows the Cell components, Sect. 6 describes the strategy of Cell-oriented computing, Sect. 7 discusses the characteristics of the proposed computing type and finally, Sect. 8 summarizes the over-arching ideas of the chapter.

2 Background

Nowadays, most computing procedures are directed towards intelligence and towards decrease processing time and cost, while the main research question today is about how to add intelligence to distributed computing. Service computing and software agent computing are the two dominant paradigms in distributed computing work within the area of Service-Oriented Architecture (SOA). Although the service computing paradigm constituted a revolution in World Wide Web, it is still regarded as a non-autonomous pattern. With the support of mobile agent's computing aptitude, the service computing model may be improved to be more efficient and dynamically prototyped. Furthermore, in the area of SOA, cloud and grid computing have become more popular paradigms and their role in developing distributed computing toward autonomy and intelligence is important. This section discusses several previous researches dealing with distributed computing intelligence from the service, agent, cloud and grid computing perspectives.

2.1 Service Paradigm

Service-oriented computing is the most cross-disciplinary paradigm for distributed computing that is changing the way software applications are designed, architected, delivered and consumed [2]. The paradigm is moving towards intelligent Web services, WSMO (Web Service Modelling Ontology) and towards semantically enhanced information processing empowered by logical inference that will eventually allow for the development of high quality techniques for the automated discovery, composition and execution of services on the Web [3]. On the other hand, Suwanapong et al. [4] propose the Intelligent Web Service (IWS) system as a declarative approach to the construction of semantic Web applications. IWS utilizes a uniform representation of ontology axioms, ontology definitions and instances, as well as application constraints and rules in machine-processable form. In order to improve single service functions and meet complex business needs, Li et al. [5] introduced composite semantic Web services based on an agent. To discover better Web service, Rajendran and Balasubramanie [6] proposed an agent-based architecture that respected QoS constraints. To improve Web service composition, Sun et al. [7] proposed a context-aware Web service composition framework that was agent-based. Their framework brings context awareness and agent-based technology into the execution of Web service composition, thus improving the quality of service composition, while at the same time providing a more suitable service composition to users. Another approach towards better Web service composition was made by Tong et al. [8], who proposed a formal service agent model, DPAWSC, which integrates Web service and software agent technologies into one cohesive entity. DPAWSC is based on the distributed decision making of the autonomous service agents and addresses the distributed nature of Web service composition. From a different perspective, Yang [9] proposed a cloud information agent system with Web service techniques, one of the relevant results of which is the energy-saving multi-agent system.

2.2 *Mobile Agent Paradigm*

The mobile agent paradigm provides many benefits in developments of distributed application, while at the same time introducing new requirements for security issues in these systems [10]. To achieve a collaborative environment, Liu and Chen [11] proposed a role-based mobile agent architecture, in which agents are grouped into a specific group according to their roles. Furthermore, through agent communication, mobile agents of an agent group can collaborate with each other to contribute to group tasks. In accordance with this agent-oriented programming approach, Telecom Italia launched the Java Agent Development Framework (JADE), which supports the following features: (a) a completely distributed situation as an existential platform for the agents, (b) a very effective asynchronous message transport protocol that provides location transparency, (c) implementation of white and yellow pages, providing easy search mechanisms for agents and their services, (d) easy, but still effective agent lifecycle management while monitoring the uniqueness of agent's ID, (e) support for agent mobility that provides a mechanism for agent code transfer to other platforms (by storing the agent's state) and (f) a flexible core that allows programmers to add new features [12]. Elammari and Issa [13] propose using Model Driven Architecture for developing Multi-Agent Systems so as to increase their flexibility and avoid any previously imposed restrictions.

Brazier et al. [14] proposed a compositional multi-agent method, DESIRE, as a methodological perspective, which was based on the software engineering principles of process and knowledge abstraction, compositionality, reuse, specification and verification.

2.3 Cloud Paradigm

Cloud computing has recently emerged as a compel paradigm for managing and delivering services over the Internet. It has rapidly modified the information technology scene and eventually made the goals of utility computing into a reality [15]. In order to provide distributed IT resources and services to users based on context-aware information, Jang et al. designed a context model based on the ontology of mobile cloud computing [16]. In the same area of smart cloud research, Haase et al. [17] discussed intelligent information management in enterprise clouds and introduced eCloudManager ontology in order to describe concepts and relationships in enterprise cloud management . In an equivalent manner, the work of Block et al. [18] establishes an alignment between ontologies in a cloud computing architecture. However, this work did not rely on reasoning among the distributed ontologies. By contrast, a distributed reasoning architecture, DRAGO, has been designed, based on local semantics [19, 20]. It uses a distributed description logics outline [21] to represent multiple semantically connected ontologies. Unlike DRAGO, the model introduced in Schlicht and Stuckenschmidt [22, 23] creates a distributed, comprehensive and terminating algorithm that demonstrates consistency of logical terminologies and promises that the overall semantics will be preserved.

2.4 Grid Paradigm

The aim of grid computing is to enable coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [24]. Shi et al. proposed an intelligent grid computing architecture for transient stable constrains that reassign a potential evaluation of future smart grids. In their architecture, a model of generalized computing nodes with an ‘able person should do more work’ feature is introduced and installed to make full use of each node [25]. GridStat has been introduced as a middleware layer capable of keeping pace with the data collection capabilities of the equipment present in the power grid [26]. Liang and Rodrigues [27] proposed a service-oriented middleware for smart grids. Their solution is capable of tackling issues related to heterogeneous services, which are most common in the smart grid domain.

3 Cell Theory

Cell theory is the modular representation of human cell characteristics from the perspective of computer science. It is a flexible and scalable virtual processing unit that treats complex distributed computing smartly by organized and accurate decisions. A cell is a software object that:

- Is sited within a command/execution environment;
- Holds the following compulsory properties:
  - Collaborative: works in groups to finish a job;
  - Inheritance: serves clients according to their environmental profile if there is no specification in their requests;
  - Shares business processes: each cell business process represents a group of business processes of components with the same goal. However, every cell is open for collaboration with all other cells and can keep up best process quality via dynamic changes in process nodes. Thus, the cell has great processing power since all cells' business processes can be shared by one cell to serve the client;
  - Uniqueness: each cell deals with a specific type of job;
  - Reactive: cell senses modification in the environment and acts in accordance with those changes;
  - Autonomous: has control over its own actions;
  - Optimal: keeps to best functional and non-functional requirements;
  - Federative: each cell has its own information resources;
  - Self-error covering: monitors changes in the computing environment and applies improvements when errors are detected;
  - Dynamic decision making: applies decision alteration based on the change of context;
  - Learning: acclimatizes in accordance with previous experience;

In order to introduce the proposed cell theory, we discuss a new software design style: the cell architecture and then show how cell computing works. For simplicity, we identify the infrastructure of the Cell-Oriented Architecture (COA) and the functionality of Cell-Oriented Computing (COC) model with definability in the mathematical model.

3.1 Cell-Oriented Architecture

COA is a novel software design principle targeted generally at Web resource computing devices. The architecture allows users to engage in smart collaborations among devices during Web resource invocations. COA is based on a centre of intelligence, which collects cells in order to exchange data between participants and manage organized standard communication methods to obtain information.

The architecture is designed to achieve smart Web goals and overcome the limitations of existing Web infrastructures. The cell architecture presented here is device, network and provider independent. This means that COA works across most computing machines and ensures a novel methodology of computing.

COA is designed to cater to smart Web requirements and aims to achieve at last an ambient, intelligent Web environment. Cells in COA are internally secured, sustain autonomic analysis of communications and are able to support the mechanism of collaborations through the following requirements:

[R1] Management and Communication: to establish local and remote sessions, the underlying infrastructure provides the ability to find any other cells in the network and then to establish a session with that cell.

[R2] Context-based Security: to enable secure interactions in the communication spaces among all connected participants.

[R3] Analysis: supporting analysis of data exchange among cells, plus encompassing the interior analysis of cell process infrastructure.

[R4] Validation: to verify cell components and ensure consistent process combinations among cells.

[R5] Output Calculation: to evaluate the suitable output results with less cost and minimal use of resources.

[R6] Trait Maintenance: to avoid and deal spontaneously with all sources of weakness in cells' communications.

To realize these goals, we developed a complete command-execute architecture, designed from the ground up to work over existing Web standards and traditional networks. COA makes it possible to merge the material and digital worlds by incorporating physical and computing entities into *smart spaces*. Put simply, it facilitates the steps to achieving a pervasive form of computing. Figure 1 outlines the components of this COA, its functionality and the operation of the underlying protocols.

COA is composed of three main components: Cell Commander, Executive Cell and Cell Feeding Source. Cell theory is introduced to provide intelligence in distributed computing; however, it combines client/server and peer-to-peer models at once. This is a client/server representation because we have a client component (the

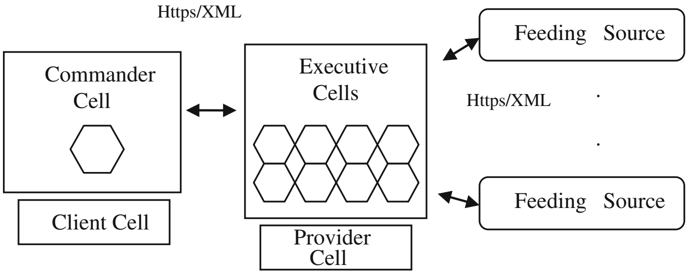


Fig. 1 Cell-oriented architecture

Commander Cell) invoking a server component (the Executer Cells) to solve a problem. On the other hand, virtually, it is an illustration of peer-to-peer applications because we have two types of cells communicating with each other.

Commander (Client) Cell: This is a commander component that looks for a procedural module to accomplish a required function. The commander can be an application, another service, or some other type of software module that needs the service. The Commander Cell works like a brain cell in the human body; it demands a solution for a definite problem and suggests a general view of the solution to be realized by a specific type of executive cells.

Executive (Provider) Cell: This is an intelligence centre consisting of a definite number of cells that are ready to serve commanders. Each cell is characterized by uniqueness of goal, self-governance, federated role, internal security and interoperability. Cell business processes, which are called genes, are built directly by the cell designer or else can be transformed by any type of service business processes. Genes use the ontology of an abstract business process and link different processes with the same purpose into a specific node. Similar to the gene in human body, each artificial gene serves a specific type of job in a different style and no other gene is capable of doing the same job. Based on gene characteristics, an Executive Cell is unique in delivering a specific type of service; for example, if a client cell requires a booking room service, there is only one, replicated, Executive book room cell to be invoked. Cell theory maintains diversity and competition between companies to serve clients; however, it hides complexity issues when selecting or composing Web processes. Solutions are prepared in an autonomic manner without any interference from the client; that is why there is no complex discovery and selection of cells or processes of composition or intervention.

Feeding Source: It represents a pre-built component that forms a base for building genes of Executive Cells. The Feeding Source can be a Web service provider, a company, or any third party that is capable to supplying a process design. A cell's internal system can use a pre-designed business process or demand the building of new designs by process designers, making it suitable to be a cell gene.

4 Structure of COA Components

The abovementioned main components of COA are discussed in detail in this section.

4.1 Commander Cell Structure

The Commander Cell represents the client side in COA and is the main requester of an output. This section discusses the structure of cells from the client side (Fig. 2).

Command Cell Manager (CCM): the client cell's 'head' that is responsible of any external collaboration with the Executive Cells. It receives a client as a list of

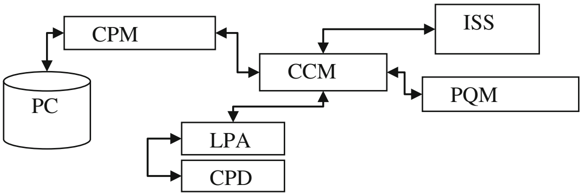


Fig. 2 Structure of client cell

four components: proposed cell input, interval of output of Executive Cell result, proposed Cell process's general design (if available) and the required cell process quality. Some of these components can be inherited from the client cell's environment. The Command Cell Manager monitors the context profile of the Commander Cell via the profile manager. It also manages the access to the client cell by specified rules of internal security.

Internal Security System (ISS): this is protection software that is responsible of giving tickets for Executive Cells to access the Command Cell manager. It depends mainly on the analysis of the outer cell's context profile to ascertain whether it can collaborate with the client Cell.

Process Quality Manager (PQM): software used by the Commander Cell to select the required quality of the cell process. For example, the client may need to specify some qualities such as performance, cost, response time, etc. If there is no selection of specific qualities, these qualities are inherited from the environment's qualities (as an employee may inherit a quality from his company).

Cell Process Designer (CPD): a graphical design interface that is used to build a general cell process flow graph or to select an option from the available process graphs. If there is no graph design or selection, the Executive Cell has the right to pick a suitable gene based on the commander profile.

Logic Process Analyser (LPA): after designing a general proposition for the executive gene design via the process designer, the job of the logic process analyser is to transform the graph design into a logical command to be sent to the executive side.

Context Profile Manager (CPM): this tool is responsible for collecting information about the Commander Cell profile, such as place, type of machine, user properties, etc. Since the commander profile is dynamic, several users may use the same Commander Cell; the profile information is instantaneously provided when needed.

Profile Core (PC): this storage is performed by a special database that stores information about the Commander Cell profile and allows the Executive Cell to tell whether there are several users utilizing the same Commander Cell.

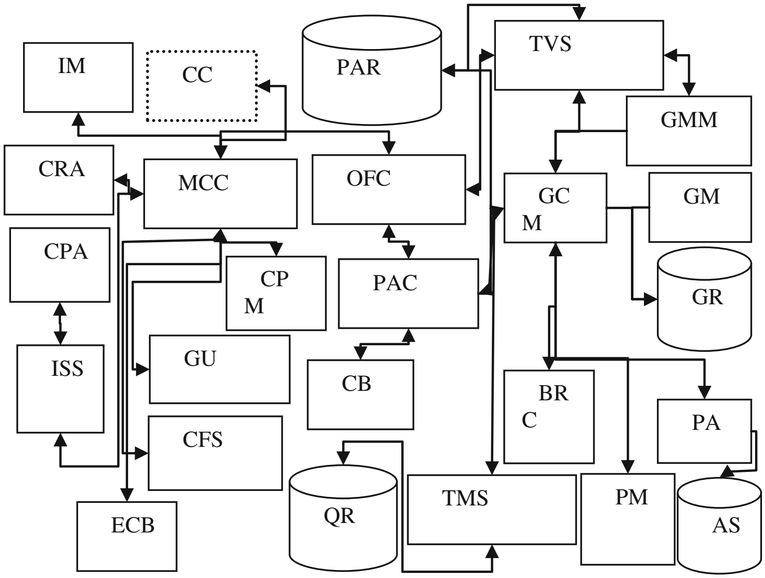


Fig. 3 Structure of cell provider

4.2 Cell Provider Infrastructure

The cell provider represents the supplier side in the COA, which is responsible for building suitable outputs for client invocation. This section discusses the structure of the cell provider shown in Fig. 3.

Management and Control Centre (MCC): Smart software work like an agent and is considered to be similar to the brain of the COA, in which it orchestrates the whole computing infrastructure. It is composed of a virtual processing unit that controls all the internal and external connections. So, Executive Cells are supported and managed according to well-defined cell level agreements. It monitors every connection among cells and prepares all decisions, such as update requirement, communication logics, maintenance facilities, access control management, repository stores and backups, etc. The COA management and control centre have stable jobs inside the cell provider. However, it cannot respond to an external job from other cells without security permission from the internal security system. Since one of the main principles of cell theory is availability, the management and control centre is replicated in order that collaboration can be carried out to serve cells. Each cell uses its Decision System to communicate with the COA management centre.

Testing and Validation System (TVS): the cell testing and validation system describes the testing of cells during the process composition phase of the Executive Cell. This will ensure that new or altered cells are fit for purpose (utility) and fit for use (warranty). Process validation is a vital point within cell theory and has often been the unseen underlying cause of what were in the past seen as inefficient cell management processes. If cells are not tested and validated sufficiently, then their introduction into the operational environment will bring problems such as loops, deadlocks, errors, etc. In a previous book chapter [28] we have discussed a new model of how to validate the business processes of Web service; the concepts of the same validation method can be used to validate the cell business process (Gene). Cell validation and testing's goal means that the delivery of activities adds value in an agreed and expected manner.

Cell Traits Maintenance System (TMS): the challenge is to make cell technology work in a way that meets customer expectations of quality, such as availability, reliability, etc., while still offering Executive Cells the flexibility needed to adapt quickly to changes. Qualities of genes are stored in a *QoG* repository and the maintenance system has permission to access and monitor these qualities. *QoG* can be considered a combination of *QoS* with a set of Web services if the source of the cell is a Web service provider. *QoG* parameters are increasingly important as cell networks become interconnected and larger numbers of operators and providers interact to deliver business processes to Executive Cells.

Process Analyser Core (PAC): since a cell process map can be composed of a set of other components' business processes, there should be a method for selecting the best direction for the cell map. In addition to the context of environment dependency, cell theory uses a deep quality of service analysis to define a best process. This type of process map analysis is summarized by building a quality of process data warehouse to monitor changes in process map nodes. Every process component invokes a set of subcomponents, similar to sub services in a service model, in which all these subcomponents are categorized in groups according to goals. The process analyser core applies analysis to these subcomponents and communicates with the cell broker to achieve the best map of the Executive Cell process. In addition to analysing Executive Cell process, the process analyser core also analyses and maps the invocations from the Commander Cells. This type of dual analysis results in an organized store of collaboration data without the need to re-analyse connections and without major data problems.

Output Fabrication Centre (OFC): depending on the specific output goal, options may be available for executive cells to communicate with the output fabrication centre. This centre provides more control over the building of the executive cell process to serve the client cell. Based on the results of the process analyser core and the consequences of the test and validation system, executive cells, specifically their output builder systems, collaborate with the output fabrication centre to return a suitable output to the commander cell.

Cell Profile Manager (CPM): traditional styles of client/server communications suffer from a weakness: the dominance of the provider. Indeed, a server can request information about client profiles for security purposes, but power is limited in the converse direction. In cell theory, every ell must have a profile to contact other cells. The cell profile manager works to build suitable profiles for executive cells to help in constructing a trusted cell instruction tunnel.

Cell Federation System (CFS): the system coordinates sharing and exchange of information which is organized by the cells, describing common structure and behaviour. The prototype emphasizes the controlled sharing and exchange of information among autonomous components by communicating via commands. The cell federation system ensures the highest possible autonomy for the different cooperating components.

Cells' Core (CC): this forms a centre of Executive Cells. A cell is an item of smart software that performs a specific type of job. All cells have the same structure but different processes. Thus, the executive cell is considered an example of a general cell component. Each executive cell is composed of seven sub-components, as follows: decision system, gene store system, trait maintenance system, output builder system, process validation system, process analyser system, defence system and gene storage. These sub-components communicate with the cell provider subsystems to carry out their jobs.

Inheritance Manager (IM): a client is observed as a Commander Cell so as to decide which types of cell inherit the properties of their environment. For example, if the commander is a professor, they can be seen a part of a university environment by Executive Cells. A commander can be part of more than one environment; and results in a hybrid profile of context. The inheritance manager maps the commander cell to its suitable environment. To serve a commander, the executive cell uses a quality of process compatible with its surroundings or follows the commander's requirements to build a suitable process.

Cell Request Analyser (CRA): cell theory is based on the concept of collaboration to serve the client. However, every client has a different request, so a computing component is needed to detect which cells will work in generating the answer. In general, the job of the cell request analyser is to map the Commander Cell to the appropriate Executer Cells to accomplish a job.

Cell Profile Analyser (CPA): this component is related to the security of cells. One of the main concepts of cell theory is its context-based property. There are sensors for profile context collecting information about the commander at the client side. The cell profile analyser verifies the commander profile by a specific method before allowing access to executer cells.

Internal Security System (ISS): since some commander cells can access sensitive data, stringent protection must be provided from the server side. The available security methods follow two types of protection: network and system protection. In network protection, the data among nodes is encrypted to hide the content from intruders. In system protection, a token (username and password), antivirus application and firewall are used. Cell theory proposes a new type of protection which is specific to the application itself. It is described as an internal system protection that verifies the profile of the user by several methods before allowing access.

Cell Process Modelling (CPM): a procedure for mapping out what the Executive Cell process does, both in terms of what various applications are expected to do and what the Commander Cells in the provider process are expected to do.

Enterprise Cell Bus (ECB): The enterprise cell bus is the interaction nerve core for cells in cell-oriented architecture. It has the propensity to be a controller of all relations, connecting to various types of middleware, repositories of metadata definitions and interfaces for every kind of communication.

Cell Broker (CB): analytical software that monitors changes in cell processes and evaluates quality of processes according to their modifications. The evaluation of quality of process is similar to that of quality of service in the service model. However, the new step can be summarized as the building of a data warehouse for quality of process that permits an advance online process analysis.

QoG Repository (QR): a data warehouse for the quality of cell process. It collects up-to-date information about process properties, such as performance, reliability, cost, response time, etc. This repository has an *OLAP* feature that support an online process analysis.

COA Governance Unit (GU): the COA governance unit is a component of overall IT governance and as such administers controls when it comes to policy, process and metadata management.

Process Analysis Repository (PAR): a data warehouse of all cells' process connections. It stores information about cell processes in the shape of a network graph, in which every sub unit of a process represents a node. The collected data summarizes analytical measures such as centrality.

Gene Core Manager (GCM): software responsible of gene storage, backups and archiving. It receives updates about business processes from sources and alters the gene ontology, backs up the gene when errors occur and archives unused genes.

Gene Mediator (GM): the problem of communication between the gene core manager and the sources of business processes may be complex, so GM defines an object that encapsulates how a set of objects interact. With the gene mediator, communication between cells and their sources is encapsulated by a mediator object. Business process sources and cells do not communicate directly, but instead communicate through the mediation level, ensuring a consistent mapping of different business process types onto the gene infrastructure.

Gene Meta-Data Manager (GMM): genes are complex components that are difficult to analyse, so for analysis and validation purposes, the gene meta-data manger invokes gene meta-data from the gene repository and supplies gene core data through this process.

Gene Repository (GR): ontologies are used as the data model throughout the gene repository, meaning that all resource descriptions, as well as all data interchanged during executive cell usage, are based on ontologies. Ontologies have been identified as the central enabling technology for the Semantic Web. The general use of ontologies allows semantically-enhanced information processing as well as support for interoperability. To facilitate the analysis of the gene map, meta-data about each gene is also stored in the gene repository.

Backup and Recovery Control (BRC): this refers to the different strategies and actions occupied in protecting cell repositories against data loss and reconstructing the database after any kind of such loss.

Process Archiving (PA): the archiving process helps to remove the cell process instances which have been completed and are no longer required by the business. All cell process instances which are marked for archiving will be taken out from the archive set database and archived to a location as configured by the administrator.

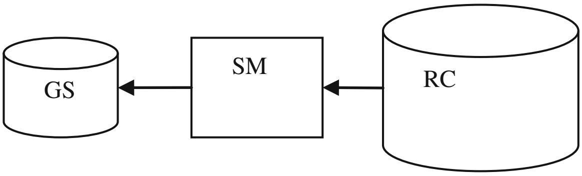


Fig. 4 Structure of cell source

The job of the process archiving component includes the process-, task- and business log-related content from the archive database.

Archive Set (AS): a database for unused genes that is accessed and managed by the process archiving component.

4.3 Cell Source

Cell source can be any kind of code that can be reused and follow specific composition rules. Generally, the first sources of cells are Web service business processes (such as BPEL and OWL-S) or reusable code (Java, C# etc.). This section discusses the structure of the sources that feed Executive Cells (Fig. 4).

Resource Code (RC): a store of cell sources, such as business processes or reusable code. If the cell source is a Web service provider, then its business process may be BPEL, OWL-S, or another. Further, the cell source may be a reusable programming code for a combination of objects (in Java, C#, etc.).

Source Mediator (SM): transformer software that maps the process of a cell's source into a gene. The mediator's job is similar to that of the BPEL parser in a Web service provider, which maps BPEL code into a WSDL code. In COA, every source business process is converted into OWL-S ontology. However, the obtained OWL-S ontology has a special property: the extension of OWL-S' business process.

Gene Store (GS): a store that is composed by mapping the source business process. This is an abstract of a source process in shape of an ontology, organized in a structure compatible with the cell's job.

5 Definitions and Notations

Definition 1 Let $W(P, Q, T)$ be a finite nonempty set that represents Web infrastructure, where: $P = \{p_1, p_2, \ldots, p_n\}$ represents the set of feeding sources of Web applications, $Q = \{q_1, q_2, \ldots, q_m\}$ represents the set of consumers of Web sources and $T = \{t_1, t_2, \ldots, t_k\}$ represents the set of tools that are used by Web providers to serve Web customer, where $n, m, k \in IN$.

Definition 2 Let set $J = \{\bigcup_m^z j_m / j_m \textit{is specific goal and } j_m \neq j_n\}$ and set $S = \{\bigcup_m^z s_m / s_i \textit{ is structure of component}\}$.

As with most things in the business world, the size and scope of the business plan depend on specific practice. A specific practice is the description of an activity that is considered important in achieving the associated specific goal. Set J represents a group of components, each of which supports a specific computing goal based on a particular practice. However, the structure of the studied components is denoted by set S.

Proposition 1 *A set* $\sigma = \{\bigcup_i^n L_i / \sigma_i \textit{denote a Cell}\} \complement T$, *is a finite and ordered set such that* $J_{\sigma_i} \cap J_{\sigma_j} = \emptyset$ *and* $S_{\sigma_i} = S_{\sigma_j}$, *where* $i, j, n \in IN$.

In all other computing models, different components may perform similar jobs. For example, two classes, in the object-oriented model, can utilize similar inputs and return the same type of output but using different coding structures. Furthermore, in the discovery phase of service-oriented computing, service consumers receive a set of services that do the same job before selecting which one of them to invoke. The main advantage of Web service theory is the possibility of creating value-added services by combining existing ones. Indeed, the variety involved in serving Web customers is useful in that it gives several aid choices to each one of them. However, this direction in computing failed since service customers found themselves facing a complex service selection process. One of the main properties of cell methodology is the avoidance of the ‘service selection’ problem. The cell is developed to provide highly focused functionality for solving specific computing problems. Every cell has its own functionality and goal to serve, so one cannot find two different cells which support the same type of job. However, all cells are similar in base and structure: they can sense, act, process data and communicate. That is to say, regarding cell structure there is only one component to deal with, while in function there are several internal components, each with a different computing method and resource.

Definition 3 Let φ be a property that expresses the collaboration relation such that $\alpha\varphi\beta$ where $\alpha, \beta \in \sigma$.

Business collaboration is increasingly taking place on smart phones, computers and servers. Cells in COC are intelligent components that are capable of collecting information, analysing results and taking decisions and identifying critical Web business considerations in a collaborative environment.

Proposition 2 *A collaboration relation* φ *defined on the set* σ *is transitive, in which, if* $\alpha\varphi\beta$ *and* $\beta\varphi\gamma = > \alpha\varphi\gamma$, *where* $\alpha, \beta, \gamma \in \sigma$.

Transitive structures are building blocks of more complex, cohesive structures, such as response-cliques, which facilitate the construction of knowledge by consensus [29]. The collaboration among cells follows a transitive mechanism to provide consistency. Transitivity among cells can be summarized by this example: if we consider three cells X, Y, Z and if X collaborates with Y, Y collaborates with Z, then indirectly X collaborates with Z.

Proposition 3 $\forall c_i \in \sigma$ *and* $\forall q_e \in Q$, $\exists \sigma_i, \sigma_j, \ldots, \sigma_n$ *s.t.* $\bigcup_{i,j}(\sigma_i \varphi \sigma_j) = \; > q_e$, *where* $i, j, e, n \in IN$.

COC's goal is to be introduced to serving Web customers with minimal cost, lower resource consumption and optimal results. For every customer request (t_e), there exists a cell collaboration ($\bigcup_{i,j}(\sigma_i \varphi \sigma_j)$) to return the appropriate answer. Cell collaboration is dynamic; results are produced without delay. Any future error in the proposed results generated by COC is corrected by an automatic repairing mechanism.

Definition 4 (*cell subsystems*)

An Executive Cell system is an ordered set $C = (DS, GSS, TMS, OBS, PVS, PAS, DFS)$ such that:

DS	set builds and manages cell decisions
GSS	set is responsible for cell process storage
TMS	set monitors the cell's characteristics
OBS	set maintains best output results of cells
PVS	set is responsible for cell process validation
PAS	set analyses the cell's business process
DFS	set is responsible for cell security

Proposition 4 *A relation between cell subsystems is managed according to a set of mathematical mappings M* $(\mu, \pi, \rho, \tau, \gamma, \delta, \varepsilon, \theta, \vartheta)$ *such that*:

$$\begin{aligned} \mu : DFS &\rightarrow DS \\ x &\rightarrow \mu(x) \end{aligned} \tag{F.1}$$

$$\begin{aligned} \pi : OBS &\rightarrow DS \\ x &\rightarrow \pi(x) \end{aligned} \tag{F.2}$$

$$\begin{aligned} \rho : TMS &\rightarrow OBS \\ x &\rightarrow \rho(x) \end{aligned} \tag{F.3}$$

$$\begin{aligned} \tau : PAS &\rightarrow OBS \\ x &\rightarrow \tau(x) \end{aligned} \tag{F.4}$$

$$\begin{aligned} \gamma : PVS &\rightarrow OBS \\ x &\rightarrow \gamma(x) \end{aligned} \tag{F.5}$$

$$\begin{aligned} \delta : GSS &\rightarrow TMS \\ x &\rightarrow \delta(x) \end{aligned} \tag{F.6}$$

$$\begin{aligned} \varepsilon : GSS &\rightarrow PAS \\ x &\rightarrow \varepsilon(x) \end{aligned} \tag{F.7}$$

$$\begin{aligned} \theta : GSS &\rightarrow PVS \\ x &\rightarrow \theta(x) \end{aligned} \tag{F.8}$$

Theorem *If q denotes a commander cell request and x denotes a cell gene,* then:

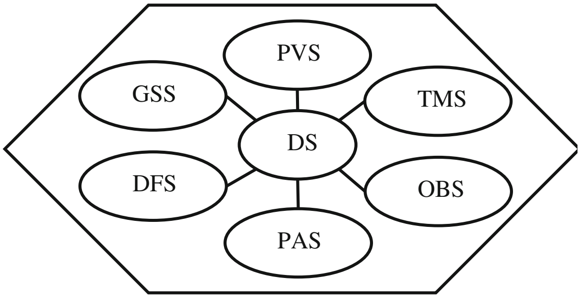
$$\mu(q) \equiv \pi(\rho(\delta(x)) \cap \tau(\varepsilon(x)) \cap \gamma(\theta(x)).$$

The management of a cell's internal system is divided among its subsystems according to a definite number of roles. In order to invoke a cell, a client request (q) must pass the cell's security system (F.1). After ensuring a secure cell invocation, DS begins the response process. It demands building output by the OBS (F.2). OBS output is based on a deep cell process analysis (F.4), a precise cell process validation (F.5) and assessing relevant cell characteristics (F.6). Tests (analysis and validation) are applied to cell process storage through GSS (F.6–F.8).

6 Components of Executive Cell

The proposed executive cell in cell theory is composed of (Fig. 5): decision system (DS), gene store system (GSS), trait maintenance system (TMS), output builder system (OBS), process validation system (PVS), process analyser system (PAS), defence system (DFS) and gene storage.

Fig. 5 Components of executive cell



6.1 Decision System (DS)

The decision system is the brain of the Executive cell in COC. It is controlled by the management and control centre and is responsible for taking decisions and directing other components of the cell. Cell inputs are received by the DS which study the client request and emit suitable outputs. Cell computing is characterized by two levels of collaboration that are managed through DSs. The first collaboration level is expressed by internal cooperation among cell subsystems, while the second level of collaboration is applied among cells to build a complete answer for cell customers. In the case of a customer request, the DS asks the defence system to verify the customer identity and request before starting the answer process. If the customer request is safe, DS sends the input to the OBS and waits for the answer. Sometimes, one cell is not sufficient to serve a customer. In this case, the DS asks for collaboration from other cells to produce an answer.

6.2 Defence System (DFS)

Cell computing aims to correct the problems of the service model. One of the main service-oriented computing problems is security. Security weakness is less of a danger in the case of Web service, but currently most cloud services are public and store sensitive data, so that any security fault may be fatal to some institutions. As a way of obtaining strict computing resource protection, COC introduces internal cell protection. As is well known, there are two main steps to protecting the Web. The first step is network protection via several encryption methods. However, the second step is characterized by server resources protection via user tokens and security tools. The proposed COC security technique ensures protection against any internal or external unauthorized access to a cell. In addition to network and system protection, the cell defence system aims to introduce a double verification method. This is a hidden type of cell protection that verifies, on one side, if a customer has the right to invoke a cell, while it also checks, on the other side, if a customer's machine is capable of receiving an output from such a cell. COC aims to make the distributed Web application as secure as possible.

6.3 Gene Store System (GSS)

There are several combinations of processes that return the same results in a distributed application. Some of these applications are Web services that are divided into a set of groups, such that in each group all the applications can do the same jobs. The problem for service theory is summed up by the question of how to select the best service from an ocean of similar job services? COC has indeed found a solution to the service selection problem. Simply put, why not transform all the business processes into a new structure to be used by a novel model like COC? In order to obtain a successful COC model, we need to build a suitable business process (gene) for each cell. The first step in building cell genes is to transform the service business processes and their combinations into a graph (or map) of abstract business processes. The obtained graph has no abstract information about any service business process. For example, if several services make a *division* job, then all of their *abstract business processes* are linked to a *division* node of the gene graph. Each cell uses a specific part of the obtained abstract graph and is known as a cell business process or gene. The gene store system's job is to store the genes and classify them, shaped by logical rules in a database to be easily used by cell subsystems.

6.4 Process Analyser System (PAS)

Changes allow companies to improve processes, to expand in new directions and to remain up-to-date with the times and technology. A business process is a sequence of steps performed for a given purpose. Business process analysis is the activity of reviewing existing business practices and changing these practices so that they fit a new and improved process. The role of PAS is to keep up-to-date analysis of the cells' business processes. A cell's business process design is based on a composition of process combinations transformed from service business processes. In order to return the best cell output, PAS must select the best plan from these combinations. This job requires a permanent process design analysis. Since business process combinations are transformed into a graph of abstract business processes, the process design analysis is achieved by a multi-graph analysis. The multigraph analysis is discussed in detail in our book chapter [30].

6.5 Process Validation System (PVS)

The cell business process, in COC, is built on a dynamic composition of a group of service business processes. If there are problems in one or more business applications that support a cell business process, then the consequences of disruption to the cell process can be serious. For example, some process compositions may result in infinite loops or deadlocks. The process validation system's job is to monitor and validate the changes in altered or new composition processes. The validation technique used by PVS is described in our chapter [28]. This validation technique is divided into two steps. First the business process is transformed into a graph. Then, a *depth first search* is applied on the obtained graph to detect deadlock errors.

6.6 Traits Maintenance System (TMS)

A cell business process is a dynamically coordinated set of collaborative and transactional activities that deliver value to customers. Cell process is complex, dynamic, automated and long running. One of the key characteristics of a good cell business process is continuous improvement. These improvements ensure a constant flow of ideal traits into the cell process. Cell computing is built upon achieving a group of architectural traits such as: performance, reliability, availability and security. These qualities require stable monitoring to maintain the supply of customers. Cells in COC apply internal and external efforts to maintain best traits. External efforts are achieved via cell collaboration, while internally the job is done by TMS. Indeed, TMS analyses the quality of gene (QoG) of a cell; these are combinations of the traditional quality of service (QoS) analysis. It uses a data warehouse of QoG to accomplish this type of analysis. The service analysis based on the QoS data warehouse is discussed in details in our book chapter [31].

6.7 Output Builder System (OBS)

Cells in COC are considered as intelligent modular applications that can be published, located and invoked across the Web. They are intended to give the client best results by composing their distributed business processes dynamically and automatically based on specific rules. Based on the service model, companies only implement their core business and outsource other application services over the Internet. However, no single Web service can satisfy the functionality required by the user; thus, companies try to combine services together in order to fulfil the request. Indeed, companies face a major problem: Web service composition is still a highly complex task and it is already beyond human capability to deal with the whole process manually. Therefore, building composite Web services with an automated or semi-automated tool is critical. As a solution to the service composition problem, cell theory proposes a cell that is capable of achieving an automated composition of its business process. In sum, after analysing, validating and ensuring the good characteristics of business process choices to be used by a cell by PAS, PVS and TMS, OBS selects and executes the best process plan based on the user's request. The role of OBS is to apply a dynamic and autonomic composition of the selected business processes of the collaborating cells.

7 Strategy of Cell Computing

Cell computing allows sharing of the process to reach a solution. This way of computing results, indirectly, in a shared resources environment similar to that of grid computing. Recursively, a client cell has access to all other executive cells as

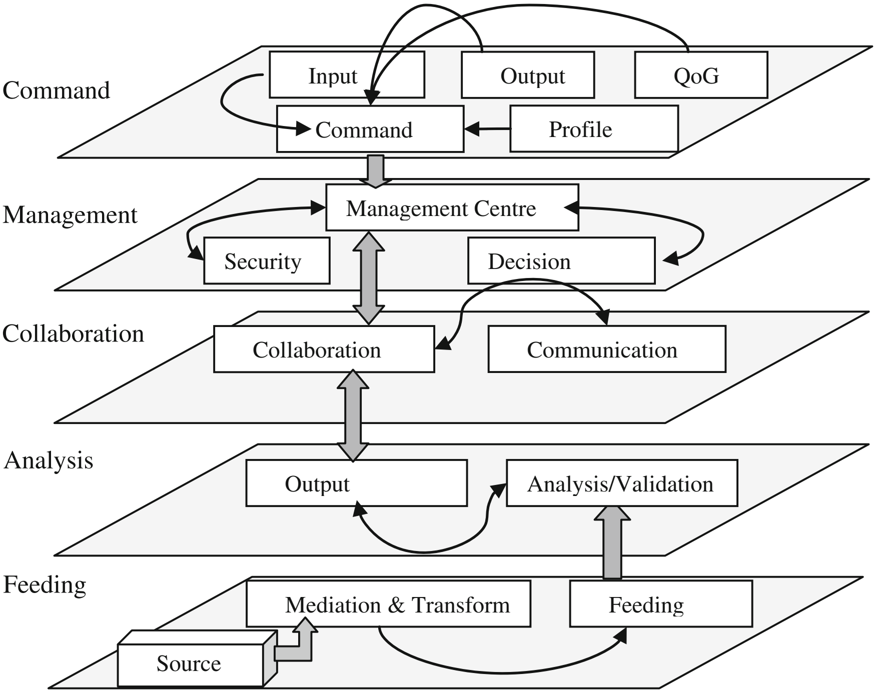


Fig. 6 Strategy of cell computing

they are running on one machine. The cell network is organized, secure, reliable, scalable and dynamic. Cell computing strategy, as shown in Fig. 6, is based on five main layers of computation: command layer, management layer, collaboration layer, analysis layer and feeding layer.

Command Layer: The command layer consists of solutions designed to make use of the smart selection of cells that can provide a specific service. It makes up the initial step of the exchange in cell architecture. An important role of the command layer is to allow for clear separation between available solutions and a logical methodology in producing a solution based on the client's command. The traditional Web service methodology gives clients the right to select one of the predesigned Web applications that will process their solution depending on several qualities and a complex selection process. However, cell methodology has improved the process by making clients give commands and creating the application according to these commands. This approach enables a slew of new applications to be created that make use of the COA's cooperative capabilities, without requiring in-depth knowledge of application processes, communication protocols, coding schemes or session management procedures; all these are handled by the upper layers of the cell strategy. This enables modular interfaces to incorporate new services via a set of commands composed of specifying inputs, output intervals, QoG requirements and the user profile.

Management Layer: this layer provides configurable controlling and reporting for client commands and server facilities at operational and services levels. It also provides visibility across physical, virtual-based layers, making it possible to govern the enforcement and migration of COA across the distributed enterprise. The management layer of the cell-based architecture not only reduces deployment, maintenance and operation costs but also allows for the provision of better performance, scalability and reliability. Its agent-based capabilities provide for comprehensive management of all cell collaboration procedures. The management layer controls the start-up and status of solutions, the logging of maintenance events, the generation and processing of Cells, the supervision of security and the management of application failures. The management layer provides centralized solution control and monitoring, displaying the real-time status of every configured solution object, as well as activating and deactivating solutions and single applications, including user-defined solutions. This layer additionally provides simple integration with a variety of enterprise-level business intelligence, reporting and event correlation tools for deeper analytics and insight. It automatically associates recovery with the solutions as active conditions in the system until they are removed by another maintenance event.

Collaboration Layer: in COA, cells work with each other to perform a task and to achieve a shared goal. They utilize recursive processing and a deep determination to reach the client's objective. Most collaboration requires leadership; in COA, each cell, by its decision system, can take the leading role. In particular, the collaborative property of the cells results in better processing power when facing competition for complex jobs. COA is based on specific rules of collaboration and manages the communications among cells. These rules characterize how a group moves through its activities. The desired cell collaboration aims to collect suitable sub-tasks that are composed to achieve a complete and efficient process in carrying out a specific job.

Analysis Layer: the analysis layer generates the statistical data used for interaction management and control centre reporting. It also enables solutions to communicate with various database management systems. Through the analysis layer, providers of processes can be seen as a store of dynamic, organized quality of process, generating new cell processes. In this layer, the collected data of ontologies that represent business processes are analysed, validated and tested before operational use by cells. Two types of analysis are used in cell methodology. The first type studies the qualities of source processes; while the other type studies the graph analysis measures of the selected sub-processes.

Feeding Layer: this layer aims to find sources of business processes and tries to handle the complexity and diversity transforming business processes through a special mediator. The feeding process starts by fetching sources about process designs and results in a semantic design, as ontology, compatible with cell requirements.

8 Discussion

The most dominant paradigms in distributed computing are Web service and software agent. Inserting intelligence into these paradigms is critical, since both paradigms depend on non-autonomous driven architecture (SOA) that prevents one-to-one concurrence with the needs of third party communication (i.e., the service registry). In addition to this, the Web service and agent paradigms suffer from negative complexity and migration effects, respectively. The complexity, in general, comes from the following sources. First, the number of services accessible over the Web has increased radically during recent years and a huge Web service repository to be searched is anticipated. Second, Web services can be formed and updated during normal processing; thus the composition system needs to detect this updating at runtime and make decisions based on the up-to-date information [32]. Third, Web services can be developed by different organizations, which use different conceptual models to describe the services; however, there is no unique business process to define and evaluate Web services. Therefore, building composite Web services with an automated or semi-automated tool is critical.

The migration of processes and the control of that migration, together with their effect on communication and security, was a problem for mobile agents. Indeed, the first problem of the Web is security; giving an application the ability to move among distributed systems and choose the place to make execution may cause critical problems. Agent methodology has several advantages; however, it can destroy human control if it is not covered by rules and limits.

How we can benefit from the wide use of service-oriented architecture in building intelligent architecture? How can we avoid the complex selection process of the Web service model? How can we achieve dynamic business process composition despite the variety of companies providing different types of service processing? How can we use the intelligence of multi-agent systems as a control mode from the client side? How can we reach the best non-functional properties of processes in an autonomic manner? How can we avoid the security weaknesses resulting from mobile agent communications? How can we prevent damage to service caused by internal and subservice fail? Why not separate software processes based on their purpose? How might we arrange procedures of distributed computing in a way that evades big data analysis problems resulting from random connections among distributed systems? How can globally consistent solutions be generated through the dynamic interaction of distributed intelligent entities that only have local information? How can heterogeneous entities share capabilities to carry out collaborative tasks? How can distributed intelligent systems learn, improving their performance over time, or recognize and manage faults or anomalous situations? Why not use dynamic online analysis centres that monitor the on-the-fly qualities of distributed software processes? How should we validate the processes of distributed software at the design phase? How should we accomplish the internal protection of distributed components based on the dual context-profile of both consumer and solution provider?

Cell methodology uses commands among smart components: neither an invocation of non-smart component nor a migration of processes. It is based on cells that can benefit from the variety of already built Web components to achieve intelligent distributed computing. They have brains, decision support systems that can do the same jobs as a mobile agent. This brain can communicate with the mobile agent on the client side by messages without process migration. Furthermore, it has its own strategy to analyse and organize connections based on communications with the management and control centre. Cell methodology requires no discovery or selection steps to use a cell because it uses a new model of the composition process to realize the user's request. It participates in solving the big data problem by making a real time analysis of communications. It is highly secure, since it uses a combination of context-aware and pervasive computing among cells.

Cells are smart components that combine a collection of characteristics from different environments. They apply autonomy and intelligence based on a mobile agent computational perspective. In addition, they map the human cell traits, such as inheritance and collaboration, into distributed computing. From the software engineering side, cells try to achieve best architecture properties such as security, availability and performance.

8.1 Autonomy

The cell approach proposes that the problem space should be decomposed into multiple autonomous components that can act and interact in a flexible way to achieve a processing goal. Cells are autonomous in the sense that they have total control over their encapsulated state and are capable of taking decisions about what to do based on this state, without a third party intervention.

8.2 Inheritance

The commander cell inherits the profile property from its environment (company, university, etc.). However, the executer cell can serve commanders according to their environmental profile (selection of suitable qualities of a process) or by special interference from the commander's side to specify more precisely the general design of a process and its qualities. This inheritance property in COA is similar to the inheritance among generations of human beings. For example, babies inherits traits of their parents such that cells combine traits from the father and the mother, but the parent can ask a doctor for specific trait in a baby different from their own traits (blue eyes, brown hair, etc.). In this case, they have given more specifications to the cell in order to select suitable genes.

8.3 Internal Security

When application logic is spread across multiple physical boundaries, implementing fundamental security measures such as authentication and authorization becomes more difficult. In the traditional client-server model, the server is most responsible for any protection requirements. Well-established techniques, such as secure socket layer (SSL), granted a so-called transport level of security. Service and agent models emphasize the emplacement of security logic at the messaging level. Cell methodology applies an internal level of security in cells. Thus, command and executive cells can communicate after the protection steps summarized by verifying the context profile of the cell that requests collaboration.

8.4 Availability

Availability of cells and data is an essential capability of cell systems; it is actually one of the core aspects giving rise to cell theory in the first instance. The novel methodology of cell theory decreases the redundancy of servers to ensure availability. Its strength lies in the ability to benefit from the redundancy of processes that serve similar goal, so failures can be masked transparently with less cost.

8.5 Collaboration

Collaborative components are need in today's primary resources to accomplish complex outcomes. Cell methodology depends on collaboration-by-command that enables coordination by one of the collaborative components. Collaboration allows cells to attain complex goals that are difficult for an individual cell to achieve. The cell collaborative process is recursive: the first collaborative agent makes a general command that is passed gradually through collaborative cells to more specific cells until reaching the desired results.

8.6 Performance

Distributed computational processes are disjointed; companies' coding is not ideal and it is difficult to monitor the complexity of every process. Thus, performance problems are widely spread among computational resources. Cell theory introduces a solution for performance problems in a distributed environment. The solution can be summarized as applying a permanent analysis of different processes aiming for the same goal, attached to a unified cell, then selecting the best process to do a job, based on basic properties such as response time and code complexity. Furthermore, an increase in communication acquaintance can be a guide to an improvement in performance as it enables cells to communicate in a more efficient manner.

8.7 Federation

Cells are independent in their jobs and goals. However, all distributed processes that do same type of job are connected to a specific executive cell. Thus each executive cell is federated with respect to the commander cell's request. Cells map can be considered as a set of federated components that are capable of collaborating to achieve a solution.

8.8 Self-error Cover

There are two types of errors that can be handled by cell computing: structural and resource errors. The cell process is based on a combination of codes that are fabricated by different computational sides. These combinations may fail because of coding or system errors and fall in deadlock. The process validation system's job is to monitor changes in process and recover errors if detected. Resource errors are described as failure in providing a service from the computational resource. The solution to these types of error is to connect spare procedures in each cell process to achieve the same quality of job from different sources.

8.9 Interoperability

Cell interoperability comes from the ability to communicate with different feeding sources and transform their business processes into cell business processes. For example, in spite of differences among business processes, such as BPEL and OWL-S, every provider of service is seen as a source of genes and as useful in cell computing. Based on cell interoperability, all procedures and applications used by service providers can be unified under a unique type of process computing, the cell gene, with respect to cell provider.

9 Conclusion

With the extensive deployment of distributed systems, the management and integration of these systems have become challenging problems, especially after smart procedures are implemented. Researchers build new technologies to deal with these problems. Trends of the future Web require inserting intelligence into the distributed computing model; thus, the goal of future research is intelligent distributed computing. At this time, the research introduces the cell computing theory to cover the distributed system problems through intelligent method of processing. Cell theory is the implementation of human cells' functions in a distributed computational environment. The cell is an intelligent, organized, secure and dynamic component that serves a specific type of job. Cell methodology divides the task between two types of components, the commander and the executer. The commander is a light cell that represents the client and can communicate smartly with its distributed environment to request solutions. The executive cell works as a smart supplier that depends on wide collaborations to fabricate a solution. Cell strategy is based on high-level communication among Cells, a permanent analysing process among collaborating components and context-based security among collaborating cells.

Acknowledgments This work has been supported by the University of Quebec at Chicoutimi and the Lebanese University (AZM Association).

References

1. Brain, M.: How cells work, howstuffworks? A Discovery Company (2013). http://science.howstuffworks.com/life/cellular-microscopic/cell.htm
2. Petrenko, A.I.: Service-oriented computing (SOC) in engineering design. In: Third International Conference on High Performance Computing HPC-UA (2013)
3. Feier, C., Polleres, A., Dumitru, R., Domingue, J., Stollberg, M., Fensel, D.: Towards intelligent web services: the web service modeling ontology (WSMO). In: 2005 International Conference on Intelligent Computing (ICIC'05), Hefei, 23–26 Aug 2005
4. Suwanapong, S., Anutariya, C., Wuwongse, V.: An intelligent web service system. Engineering Information Systems in the Internet Context, IFIP—The International Federation for Information Processing vol. 103 (2002), pp. 177–201 (2014)
5. Li, C., Zhu, Z., Li, Q., Yao, X.: Study on semantic web service automatic combination technology based on agent. In: Lecture Notes in Electrical Engineering, vol. 227, pp. 187–194. Springer, Berlin (2012)
6. Rajendran, T., Balasubramanie, P.: An optimal agent-based architecture for dynamic web service discovery with QoS. In: International Conference on Computing Communication and Networking Technologies (ICCCNT) (2010)
7. Sun, W., Zhang, X., Yuan, Y., Han, T.: Context-aware web service composition framework based on agent, information technology and applications (ITA). In: 2013 International Conference
8. Tong, H., Cao, J., Zhang, S., Li, M.: A distributed algorithm for web service composition based on service agent model. IEEE Trans. Parallel Distrib. Syst. **22**, 2008–2021 (2011)
9. Yang, S.Y.: A novel cloud information agent system with web service techniques: example of an energy-saving multi-agent system. Expert Syst. Appl. **40**, 1758–1785 (2013)
10. Maryam, M., Varnamkasti, M.M.: A secure communication in mobile agent system. Int. J. Eng. Trends Technol. (IJETT) **6**(4), 186–188 (2013)
11. Liu, C.H., Chen, J.J.: Role-based mobile agent for group task collaboration in pervasive environment. In Second International Conference, SUComS 2011, vol. 223, pp. 234–240 (2011)

12. Rogoza, W., Zabłocki, M.: Grid computing and cloud computing in scope of JADE and OWL based semantic agents—a survey, Westpomeranian Technological University in Szczecin (2014). doi:10.12915/pe.2014.02.25
13. Elammari, M., Issa, Z.: Using model driven architecture to develop multi-agent systems. Int. Arab J. Inf. Technol. **10**(4) (2013)
14. Brazier, F.M.T., Jonker, C.M., Treur, J.: Principles of component-based design of intelligent agents. Data Knowl. Eng. **41**, 1–27 (2002)
15. Shawish, A., Salama, M.: Cloud computing: paradigms and technologies. Stud. in Comput. Intell. **495**(2014), 39–67 (2014)
16. Jang, C., Choi, E.: Context model based on ontology in mobile cloud computing. Commun. Comput. Inf. Sci. **199**, 146–151 (2011)
17. Haase, P., Tobias, M., Schmidt, M.: Semantic technologies for enterprise cloud management. In Proceedings of the 9th International Semantic Web Conference (2010)
18. Block, J., Lenk, A., Carsten, D.: Ontology alignment in the cloud. In Proceedings of ontology matching workshop (2010)
19. Ghidini, C., Giunchiglia, F.: Local model semantics, or contextual reasoning = locality + compatibility. Artif. Intell. **127**(2), 221–259 (2001)
20. Serafini, L., Tamilin, A.: DRAGO: distributed reasoning architecture for the semantic web. In: Proceedings of the Second European Conference on the Semantic Web: Research and Applications (2005)
21. Borgida, A., Serafini, L.: Distributed description logics: assimilating information from peer sources. J. Data Semant. **2003**, 153–184 (2003)
22. Schlicht, A., Stuckenschmidt, H.: Distributed resolution for ALC. In: Proceedings of the 21th International Workshop on Description Logics (2008)
23. Schlicht, A., Stuckenschmidt, H.: Peer-peer reasoning for interlinked ontologies. Int. J. Semant. Comput. (2010)
24. Kahanwal, B., Singh, T.P.: The distributed computing paradigms: P2P, grid, cluster, cloud, and jungle. Int. J. Latest Res. Sci. **1**(2), 183–187 (2012). http://www.mnkjournals.com/ijlrst.htm
25. Shi, L., Shen, L., Ni, Y., Bazargan, M.: Implementation of an intelligent grid computing architecture for transient stability constrained TTC evaluation. Journal Electr Eng Technol **8** (1), 20–30 (2013)
26. Gjermundrod, H., Bakken, D.E., Hauser, C.H., Bose, A.: GridStat: a flexible QoS-managed data dissemination framework for the Power Grid. IEEE Trans. Power Deliv. **24**, 136–143 (2009)
27. Liang, Z., Rodrigues, J.J.P.C.: Service-oriented middleware for smart grid: principle, infrastructure, and application. IEEE Commun. Mag. **2013**(51), 84–89 (2013)
28. Karawash, A., Mcheick H., Dbouk, M.: Intelligent web based on mathematic theory, case study: service composition validation via distributed compiler and graph theory. Springer's Studies in Computation Intelligence (SCI) (2013)
29. Aviv, R.: Mechanisms of Internet-based collaborations: a network analysis approach. Learning in Technological Era, 15–25 (2006). Retrieved from http://telem-pub.openu.ac.il/users/chais/2006/04/pdf/d-chaisaviv.pdf
30. Karawash, A., Mcheick H., Dbouk, M.: Simultaneous analysis of multiple big data networks: mapping graphs into a data model. Springer's Studies in Computation Intelligence (SCI), (2014a)
31. Karawash, A., Mcheick H., Dbouk, M.: Quality-of-service data warehouse for the selection of cloud service: a recent trend. Springer's Studies in Computation Intelligence (SCI) (2014b)
32. Portchelvi, V., Venkatesan, V.P., Shanmugasundaram, G.: Achieving web services composition—a survey. Sci. Acad. Publ. **2**(5), 195–202 (2012)

Application of Genetic Algorithms for the Estimation of Ultrasonic Parameters

Mohamed Hesham Farouk El-Sayed

Abstract In this chapter, the use of genetic algorithm (GA) is investigated in the field of estimating ultrasonic (US) propagation parameters. Recent works are, then, surveyed showing an ever-spread of employing GA in different applications of US. A GA is, specifically, used to estimate the propagation parameters of US waves in polycrystalline and composite materials for different applications. The objective function of the estimation is the minimization of a rational difference error between the estimated and measured transfer functions of US-wave propagation. The US propagation parameters may be the phase velocity and attenuation. Based on tentative experiments, we will demonstrate how the evolution operators and parameters of GA can be chosen for modeling of US propagation. The GA-based estimation is applied, in a test experiment, on steel alloy and Aluminum specimens with different grain sizes. Comparative results of that experiment are presented on different evolution operators for less estimation errors and complexity. The results prove the effectiveness of GA in estimating parameters for US propagation.

Keywords Genetic algorithm (GA) · Inverse problem characterization · Ultrasonic (US) non-destructive testing (NDT) · Transfer function (TF) parameter estimation · Materials characterization

1 Introduction

Ultrasound waves are used, in practice, for nondestructive testing (NDT) and evaluation (NDE) of materials. In this area, the evaluation is achieved through estimating material parameters which are related to such wave propagation. The estimation of ultrasonic (US) propagation parameters; phase velocity and attenuation,

---

M.H.F. El-Sayed (✉)
Engineering Mathematics and Physics Department, Faculty of Engineering,
Cairo University, Giza 12613, Egypt
e-mail: mhesham@eng.cu.edu.eg

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_3

is an important task for many applications. US waves which have been transmitted through a material sample can be measured in the form of discrete time-series. Analysis of acquired time-series in both time and frequency domains allow acoustic and, hence, mechanical parameters of such sample to be estimated [8, 9] like wave velocity, attenuation or density [5]. The complexity of the transfer function of US wave requires an efficient estimation technique for identifying these parameters. The estimation of US propagation parameters was studied in different works as in [12, 15, 19].

The parametric modeling of the propagation transfer function (T.F.) is an appropriate approach when the material is either dispersive or exhibits frequency-dependent attenuation. The T.F. is obtained through a through-transmission experiment in which an US wave is transmitted through a test specimen. T.F. spectrum can be, then, expressed as a rational function in terms of the propagation velocity and attenuation. In general, the model parameters may be estimated by minimizing the error between the modeled spectrum and the measured one. However, most of the traditional optimization methods have many drawbacks when applied to multi-extremely nonlinear functions [30]. Well-developed techniques such as least-square, instrumental variable and maximum likelihood exist for parameters estimation of models. However, these techniques often fail in searching for the global optimum if the search space is not differentiable or continuous in the parameters [27]. Gradient-based methods may offer a sufficiently good approach. Nevertheless, in these cases if the measurements are noisy, such methods possibly will fail [28]. For these reasons, genetic algorithm (GA) may help in avoiding such a failure.

GAs are a subclass of evolutionary computational methods which emulate nature in evolution. A GA is a parallel optimization method which searches through the space of system parameters. A GA applies operators inspired by the mechanism of natural selection to a population of binary strings encoding the parameters space. GAs are considered as global optimizers which avoid the convergence to weak local solutions [30]. Over many generations, natural populations of the estimated parameters evolve according to the principle of natural selection and the survival of fittest including the concepts of crossover and mutation. The GAs give fast and accurate estimates irrespective of the complexity of the objective function. At each generation, it judges the objective function for different areas of the parameters space, and then directs the search towards the optimum region. By working with a population of solutions the algorithm can in effect search many local solutions and thereby increases the likelihood of finding the global one. It differs from other optimization techniques in that no differentiation is incurred through the algorithm and accordingly the searching space continuity is not a condition. These features enable GA to iterate several times guiding the search towards the global solution [15]. Recently, the evolution learning has been applied on identifying US data [5, 9, 22, 29].

2 Genetic-Algorithms

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. The continuing performance improvements of computational systems has made them attractive for some types of optimization. In particular, GA works very well on mixed (continuous and discrete), combinatorial problems. They are less susceptible to get local optima than other methods. Despite they did not require gradient calculations, they tend to be computationally expensive.

All living organisms consist of cells. In each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serves as a model for the whole organism. A chromosome consists of genes, blocks of DNA. Each gene encodes a particular protein. Each gene encodes a trait, for example color of eyes. Possible settings for a trait (e.g. blue, brown) are called alleles. Each gene has its own position in the chromosome. This position is called locus.

During reproduction, crossover, firstly, occurs. Genes from parents form in some way the whole new chromosome. The new created generation can then be mutated. Mutation means, that the elements of DNA are a bit changed. The fitness of an organism is measured by getting a good-state of the organism [2].

If we are solving a problem, we are usually searching for some solutions, which should be the best among others. The space of all feasible solutions is called search space (also state space). Each point in the search space represents one probable solution. Each solution can be, then, chosen or discarded according to its fitness for the problem. The final solution is one point (or more) [3] which have extreme values of an objective function in the search space. In many cases, the search can be very complicated. One does not know where to look for the solution and where to start. There are many methods, how to find some suitable solution (i.e. not necessarily the best solution), for example hill climbing, tabu search, simulated annealing and genetic algorithm. The solution found by these methods is often considered as a good solution, because it is not often possible to prove that it is optimum. In the following steps, we can summarize a typical GA.

Outline of the Basic Genetic Algorithm [11]

1 [**Start**] Generate random population of *n* chromosomes (suitable solutions for the problem)

2 [**Fitness**] Evaluate the fitness through an objective function $f(x)$ for each chromosome x in the population

3 [**New population**] Create a new population by repeating following steps until the new population is complete

[**Selection**] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

[**Crossover**] With a crossover probability, apply cross-over operator on the parents to form a new generation. If no crossover was performed, new children is an exact copy of his parent.

[**Mutation**] With a mutation probability, apply the mutation operator on the new generation at the selected position in chromosome.

4. [**Accepting**] Place new children in the new population
5. [**Replace**] Use new generated population for a further run of algorithm
6. [**Test**] If the end condition is satisfied, stop, and return the best solution in current population

[**Loop**] Go to step **2**

2.1 Evolution Operators

In summary, a GA operates on a symbolic representation of an estimated variable x which might be known as a chromosome p. A chromosome is a concatenation of genes which decodes to the x either in binary or decimal format. The chromosome is symbolically denoted as $p = \{g_j/j = 1,2,\ldots\ldots,N_{gl}\}$ where N_g is the genetic length or the number of alleles. Each individual chromosome has a corresponding fitness value of an objective function. The GA has also evolution operators which manipulate the search space represented by the chromosomes population. Typical operators are selection, crossover and mutation. The GA operators produce a succession of population whose members will generally improve the objective function values by scanning the parameters space. Given an initial population which is randomly distributed over the parameters space $p^k = \{p_i^k / i = 1,2,\ldots\ldots,T\}$, gives the corresponding values of objective function $F^k = \{f_i^k / i = 1,2,\ldots\ldots,T\}$. Each objective function value f_i^k is associated with a chromosome p_i^k. The application of reproduction operators on the initial population results in new improved generation p^k for any iteration k. Different alternatives of such operators have been extensively surveyed in [1, 4, 25].

The selection operator is intended to improve the average quality of the population by giving the individuals of higher fitness, a higher probability to be copied into the next generation and may exclude lower fitness individuals. There are several selection schemes; namely, tournament, truncation, linear and exponential ranking, and fitness proportional selections.

2.2 Chromosome

The chromosome is composed of the parameters which represent a solution for the studied problem. The most used way of encoding is a binary string. The chromosome then could look like this:

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

There are many other ways of encoding. This depends mainly on the solved problem. For example, one can encode directly integer or real numbers, sometimes it is useful to encode some permutations and so on.

2.3 Crossover Operator

During the reproductive phase of the GA, individuals are selected from the population and recombined, producing new generation which will comprise the next generation. Parents are selected randomly from the population which favors the more fit individuals. Good individuals will probably be selected several times in a generation, poor ones may be discarded. Having selected two parents, their chromosomes are recombined, typically using the mechanisms of crossover and mutation.

Crossover takes two individuals, and cuts their chromosome strings at some randomly chosen position, to produce two “head” segments and two “tails”. The segments are then swapped over to produce two new full length chromosomes. Crossover can then look like this (is the crossover point):

Chromosome 1	**11011·00100110110**
Chromosome 2	11001·11000011110
New generation 1	**11011**·11000011110
New generation 2	11001·**00100110110**

This is known as a single point crossover mechanism. Multipoint mechanisms are also used. Adding more crossover points may disrupt the building blocks of different chromosomes. However, an advantage of having more crossover points is that the problem space may be searched more thoroughly [3].

Crossover is not usually applied to all pairs of individuals selected for mating. Crossover probability determines the number of selected pairs to be recombined. The typical value of crossover probability is between 0.5 and 1 [2, 3, 25]. In uniform crossover, each gene in the new generation is created by copying the corresponding gene from one or another parent, chosen according to a randomly generated crossover mask. The process is repeated with the parents exchanged to produce the second generation. A new crossover mask is randomly generated for each pair of parents.

Crossover can be rather complicated and very dependable on encoding of the chromosome. Specific crossover made for a specific problem can improve performance of the genetic algorithm. A comparative study on crossover shows that the increased disruption of uniform crossover is beneficial if the population size is small and so gives a more robust performance [2, 3, 25].

2.4 Mutation Operator

Mutation is applied to each individual after crossover. It randomly alters each gene with a small probability (typically 0.001–0.02). This is to prevent falling all solutions in population into a local optimum of solved problem. The mutation is important for rapidly exploring the search space. Mutation provides a small amount of random search and helps to ensure that no point in the space has a zero probability of being examined [3].

For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can described through the following examples:

Original children 1	110**1**111000011110
Mutated children 1	110**0**111000011110
Original children 2	110110**0**100110110
Mutated children 2	110110**1**100110110

Some studies have clarified that much larger mutation rates decreasing over the course of evolution are often helpful with respect to the convergence reliability and velocity of GA [1]. A larger rate can speed up the search in early searching phases and then fine examination is followed using smaller rate. Some works excluded the mutation operator from GA which is used to segment a multi-component image [23].

3 Ultrasonic Waves

The term “ultrasonic” applied to sound with frequencies above audible sound, and nominally includes any frequency over 20 kHz. Frequencies used for medical diagnostic ultrasound scans extend to 10 MHz and beyond. The range of 20–100 kHz are commonly used for communication and navigation by bats, dolphins, and some other species.

US is based on the vibration in materials which is generally referred to as acoustics. All material substances are composed of atoms, which may be forced into vibrational motion about their equilibrium positions. Many different patterns of vibrational motion exist at the atomic level; however, most are irrelevant to acoustics and US testing. Acoustics is focused on particles that contain many atoms that move in harmony to produce a mechanical wave. When a material is not stressed in tension or compression beyond its elastic limit, its individual particles perform elastic oscillations. When the particles of a medium are displaced from their equilibrium positions, internal restoration forces arise. These elastic restoring forces between particles, combined with inertia of the particles, lead to the oscillatory motions of the medium. These mechanisms make solid materials as good conductor for sound waves.

The interaction effect of sound waves with the material is stronger the smaller the wave length which should be in the order of internal dimensions between atoms, this means the higher the frequency of the wave. Typical frequency range between about 0.5 and 25 MHz is used in NDT and NDE [14]. US NDT is one of the most frequently used methods of testing for internal flaws. This means that many volume tests are possible with the more economical and non-risk US test method. In cases where the highest safety requirements are demanded (e.g. nuclear power plants, aerospace industry) US methods are useful.

US waves are emitted from a transducer. US transducer contains a thin disk made of a crystalline material with piezoelectric properties, such as quartz. When alternating voltage is applied to piezoelectric materials, they begin to vibrate, using the electrical energy to create movement. When the mechanical sound energy comes back to the transducer, it is converted into electrical energy. Just as the piezoelectric crystal converted electrical energy into sound energy, it can also do the reverse.

In solids, sound waves can propagate in four principal modes that are based on the way the particles oscillate. Sound can propagate as longitudinal, shear, surface and in thin materials as plate waves. Longitudinal and shear waves are the two modes of propagation most widely used in US testing. Sound waves does, then, transmit at different speeds in different materials. This is because the mass of the atomic particles and the spring constants are different for different materials. The mass of the particles is related to the density of the material, and the spring constant is related to the elastic constants of a material. This relation may take a number of different forms depending on the type of wave (longitudinal or shear) and which of the elastic constants that are used. In isotropic materials, the elastic constants are the same for all directions within the material. However, most materials are anisotropic and the elastic constants differ with each direction. For example, in a piece of rolled aluminum plate, the grains are elongated in one direction and compressed in the others and the elastic constants for the longitudinal direction differs slightly from those for the transverse or short transverse directions.

When sound travels through a medium, its intensity diminishes with distance. In idealized materials, sound pressure (signal amplitude) is reduced due to the spreading of the wave. In natural materials, however, the sound amplitude is further weakened due to the scattering and absorption. Scattering is the reflection of the sound in directions other than its original direction of propagation. Absorption is the conversion of the sound energy to other forms of energy. The combined effect of scattering and absorption is called attenuation [16].

Basically, US waves are impinged from an US transducer into an object and the returning or transmitted waves are analyzed. If an impurity or a crack is present, the sound will bounce off of them and be seen in the returned signal. US measurements can be used to determine the thickness of materials and determine the location of a discontinuity within a part or structure. This is done by accurately measuring the time required for a US pulse to travel through the material and reflect from the back surface or the discontinuity.

4 Review on Previous Works

An earlier work in [9] explained the use of GA for estimating acoustic properties from the T.F. of US transmission. A comparative approach is followed for choosing the efficient evolution operators and parameters of GA. GA has been, recently, used to estimate the propagation parameters of US waves in polycrystalline materials. The objective function of the estimation is the minimization of a rational difference error between the estimated and measured transfer functions of US propagation model. The US propagation parameters are, customarily, the phase velocity and attenuation. A frequency-dependent form for the attenuation is considered during estimation. The proposed approach is applied on steel alloy and Aluminum specimens with different grain sizes. Comparative results are presented on different evolution operators for less estimation errors and search time. A recent study recommended the best selection operator, crossover technique and a mutation rate. The results are also compared with other previous works which adopt traditional methods for similar US problems [8, 18].

The mechanical properties of composite materials under loading conditions may be estimated by some conventional techniques like tensile and compressive tests. Such tests are destructive in nature and provide only a few elastic constants and are difficult to perform on thin structures [29]. US techniques possibly help in these aspects over the conventional techniques. The thickness of samples should be in the same order as the wavelength in the medium, otherwise or in case of multilayer samples, overlapping can occur and direct measurement of parameters is no longer possible [5]. A 1-D wave propagation model helps in obtaining an estimation for the spectrum of the signal transmitted through multilayer sample. In [5] US characterization of porous silicon applied GA to estimate acoustic properties of composites with different thicknesses. An one dimensional model describing wave propagation through a water immersed sample is used in order to compute transmission spectra. Optimum values of thickness, longitudinal wave velocity and density are obtained through searching parameters space using GA. A validation of the method is performed using aluminum (Al) plates with two different thicknesses as references. The experiment, is, then applied for obtaining the parameters for a bulk silicon wafer and a porous silicon layer etched on silicon wafer. A good agreement between retrieved values and theoretical ones is observed even in the case where US signals overlap.

In general, GA introduces a good alternative for solving inverse problems of estimating T.F. parameters. This is the case especially in finding the solution of complicated inverse problems, such as those resulting from the modeling and characterization of complex nonlinear systems, such as in particular materials with nonlinear elastic behavior. In [6] inverse problem solution of US T.F. is discussed highlighting the difficulties inherent in the application of traditional analytical–numerical techniques, and illustrating how GAs may in part alleviate such problems.

Another inverse estimation based on GA is presented in Sun et al. [26] to determine the elastic properties of functional graded materials (FGMs) plate from lamb wave phase velocity data. The properties of FGMs plate are estimated by minimizing the standard deviations between the actual and calculated phase velocities of lamb waves. This GA based work proves reliable determination of the FGM parameters with deviation which can be controlled below 5 %.

Another type of US data are collected from self emission of different materials under stress and called acoustic emission (AE). In Sibili et al. [24], a GA achieves superiority over the k-means algorithm since it allows the better clustering of AE data even on complex data sets.

Meanwhile, the goniometry based US through transmission data can be collected by using wide aperture low cost polyvinylidene fluoride (PVDF) based receiver for composite materials. The elastic moduli of polymer matrix based structural composite materials is necessary to characterize their strength after fabrication and while in service. The use of a hybrid inversion technique which combines GA based evolutionary approach, critical angle information and the use of stiffness invariants is implemented for determination of stiffness values from experimental transmission spectra measurements. Promising results for unidirectional and a multi-layered composite laminate are presented in [21].

In Luo et al. [17] US speckle technique is used for measuring the strain of a material. In this method, displacement measurements of an inner surface for underwater object are correlated. The GA searches, after adjusting genetic parameters, for the maximum value among the whole distribution of correlation coefficients efficiently. The results obtained with different algorithms, including the adaptive genetic, coarse-fine interpolation and hill-climbing searching algorithms, were compared with each other in Luo et al. [17]. It was clear that the adaptive GA (AGA) outperformed other methods in computational time and accuracy. Additionally, there was a good agreement of the measured strain with the corresponding simulation results from finite element analysis. Considering this performance and the penetration of ultrasound, this study recommends the US speckle measurement technique based on AGA for measuring strain of a material.

Another work reports a GA-based reconstruction procedure in Vishnuvardhan et al. [29] to determine the elastic constants of an orthotropic plate from US propagation velocity. Phase velocity measurements are carried out using US back-reflection technique on laminated unidirectional graphite–epoxy and quasi-isotropic graphite–epoxy fiber reinforced composite plates. The GA-based estimation using data obtained from multiple planes were evaluated and it was sufficient for the computation of seven elastic constants.

Reconstruction from US measurements using GA is, also, considered in Kodali et al. [13] for detecting material inclusion. Simulation results of the ultrasound computerized tomography (CT) are obtained with enhanced GA for detecting material inclusion. Multiple types of inclusions are detected in the test specimen to be reconstructed in this work. In addition to being able to identify inclusion of different materials, both the shape and location of the inclusion could be reconstructed. The preliminary results for a simple configuration are found to be better than previously reported ones. The results are, also, found to be consistent and satisfactory for a wide range of sizes and geometries of inclusion.

A different work explains in [7] the development of an effective methodology to determine the optimum welding conditions that maximize the strength of joints produced by US welding using response surface methodology (RSM) and GA. RSM is utilized to create an efficient analytical model for welding strength in terms of welding parameters namely pressure, weld time, and amplitude. Experiments were conducted as per central composite design of experiments for spot and seam welding for Aluminum specimens with different thicknesses. An effective second-order RSM is proposed utilizing experimental measurements. The model parameters are, then, estimated using GA optimizing the welding conditions for desired weld strength. Optimum welding conditions produced by GA are verified with experimental results and are found to be in good agreement.

As wideband piezoelectric transducers are extensively used for generating and detecting US waves in different applications such as industrial non-destructive evaluation, medical imaging and robotic vision. Estimating their operation and design parameters is essential. In Toledo et al. [28], a GA is applied for the estimation of some internal parameters of broadband piezoelectric transducers. The GA implementation was developed considering the inclusion of different terms in the objective function. Linear and quadratic terms have been included in such objective function. The performance of the GA estimation procedure was evaluated for different analyzed cases, and the implementation was applied to a practical wideband piezoceramic transducer. The estimation results were compared with experimental data based on the emitting transfer function of a practical US probe.

5 Objective Function Formulation

5.1 Through-Transmission Set-up and Transfer Function

The experimental setup used in US parameter estimation may be called *through-transmission technique* and is shown in Fig. 1.

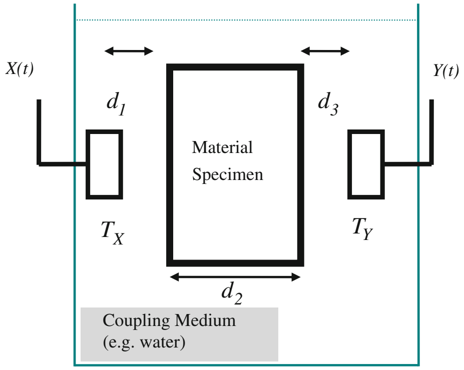


Fig. 1 Experimental setup for US testing, *Tx* and *Ty* are US transducers

T_X and T_Y are, respectively, the transmitter and receiving transducers. $X(t)$ is the input US pulse and $Y(t)$ is the variation of output with time t after passing through the specimen and making multiple reflection and transmission through it. According to the wave propagation model which describes the propagation of the US waves through the tested material, we can write [20]

$$H(\omega) = H_m(\omega).H_{cal}^{np}(\omega) \tag{1}$$

Where;

$$H_{cal}^{np}(\omega) = \frac{Y_{cal}(\omega)}{X_{cal}(\omega)}, \quad H_m^{np}(\omega) = \frac{Y_m(\omega)}{X_m(\omega)},$$

$H_m^{np}(\omega)$ is the non-parametric or measured T.F.

$X_m(\omega)$ and $Y_m(\omega)$ are, respectively, the input and output pulses in the frequency domain (where $\omega = 2\pi f$ and f is frequency of the pulse) when the specimen is immersed within the coupling medium which might be water in most cases. While, $X_{Cal}(\omega)$ and $Y_{Cal}(\omega)$ are respectively, the input and output pulses when the specimen is removed from the water path (during the calibration experiment).

The parametric T.F. of the propagation model $H_m(\omega)$ can be defined as follows,

$$H_m(\omega) = e^{-jK_w d_2} e^{-jK_M d_2} \frac{1 - R_{12}^2}{1 - R_{12}^2 e^{-2jK_M d_2}} \tag{2}$$

where; R_{12} is the reflection coefficient at the interface between the specimen and the coupling medium while the material wave number K_M is written as [16]

$$K_M = \frac{\omega}{V} - i\alpha(\omega),$$

Where V is the phase velocity of US propagation inside the tested material and α is the attenuation factor. K_w is the wave number of the water and d_2 is the thickness of the used specimen as shown in Fig. 1.

For metals, it is more practical for the attenuation to be formulated depending on frequency as follows [16]

$$\alpha(f) = a_1 f + a_2 f^4 \tag{3}$$

where, $\alpha(f)$ is the attenuation factor which depends on frequency f of US wave, a_1 is a constant that depends on hysteresis whilst the constant a_2 depends on scattering.

Thus, following this formulation, the attenuation factor will be determined as a function of frequency by the estimation of the values of the two constants a_1 and a_2. The phase velocity can be determined from the attenuation factor as a function of frequency by using the Kramers-Kronig relationship and it is given as [18]

$$V(f) = \frac{1}{V_0^{-1} + a_0 - \frac{a_1}{\pi^2}\ln(f) - \frac{a_2}{3\pi^2}f^3} \tag{4}$$

where; V_o is the dispersion-less phase velocity at the center frequency of the used transducer having f_o = 2 MHz.

and $$a_0 = \frac{a_1}{\pi^2}Ln(f_0) + \frac{a_2}{3\pi^2}f_0^3.$$

V and α were expressed as defined above and the set of parameters to be estimated are *a1, a2,* and V_0.

The application of this formulation to the through transmission experiment causes no restrictions on the behavior of the attenuation factor and the phase velocity with frequency. Every term in this formula has a physical interpretation for the impact of either scattering or hysteresis.

5.2 Objective Function

The time domain experimental data is processed and converted to the frequency domain by using Fourier Transform. The measured T.F. $H_m^{np}(\omega)$ is obtained using the through transmission experiment. This T.F. is compared with a modeled one using the form of $H(\omega)$ in (1). The simulated $H_m(\omega)$ is obtained by assuming a set of parameters describing the attenuation factor and the phase velocity as defined in (3) and (4). An error might be defined as the sum of the squared differences between the spectrum coefficients of the measured T.F $H_m^{np}(\omega)$ and the modeled one $H(\omega)$ as in Eq. (5). The squared difference error is, then, taken as a percentage of the total energy of the measured spectrum.

$$e = \frac{\sum_{i=1}^{p}\left[H_m^{np}(\omega_i) - H(\omega_i)\right]^2}{\sum_{i=1}^{p}\left[H_m^{np}(\omega_i)\right]^2} \times 100\% \tag{5}$$

where p is the spectrum order.

This error is, then, considered as an objective function which needs to be minimized.

5.3 Parameters Estimation Using GA

The minimization of the error in (5) is searched using GA over the space of *3* parameters which are V_o, *a1,* and *a2*. This process cannot be considered unconstrained search since the estimated velocity and attenuation have natural mean values. In other words, the natural mean values of the estimated parameters should

Table 1 An example of GA control parameters for US parameters estimation [9]

Control parameter	Size
Chromosome representation is binary	16 bits for each parameter i.e. the total no of bits is 48
population size	20
Max. no of generations	50
Min. parameter value	0
Max. parameter value	32,768
creep mutation probability	0.57
crossover probability	0.5
mutation probability	0.5
Tournament size (*t*)	2

not be exceeded. Accordingly, such constraints are included and the estimated parameter becomes just a perturbation (x_e) around the known mean value (x_o) as $x = x_o + x_e$ where x, is the parameter included into the T.F.

According to the above discussion, the following GA is used to deal with US parameter estimation problem

1. The individual chromosome is composed of cascaded propagation parameters in binary format as described in Sect. 3.
2. A starting population has been assumed using a normalized random distribution with a size as shown in Table 1.
3. Fitness function is evaluated for each individual and a fitness array can be obtained using Eq. (5).
4. Tournament selection can be applied to the population with a tournament size t [25].
5. Reproduction operators are then applied as discussed in Sect. 3.
6. New population is resulted as a new generation.
7. Iterations lead to step 4 until error limit is achieved or maximum iterations are exceeded.

The nominal value of the mutation probability is in the order of 0.02, however, it may be better to use higher values as indicated in some studies [9]. These studies have clarified that much larger mutation rates decreasing over the course of evolution are often helpful with respect to the convergence reliability and velocity of a GA [1]. This value helps in avoiding stagnation, to speed up the task and reduces the processing time to few generations toward the optimum parameters.

6 Illustrative Results

A through-transmission experimental set up was used as in Fig. 1. with transducers having *2 MHz* central frequency and water coupling medium. Five Steel specimens are used in [9] after a heat-treatment of the original steel sample. The Steel-A specimens contain 0.26 % Carbon. Samples of the used Steel (A1 to A4) were annealed for different periods of time at 900 °C in order to obtain different Ferrite grain sizes [18]. The annealing has been followed by grinding and polishing to remove any surface oxide. The grain sizes were measured by the linear intercept method [18]. Each recorded value is an average of 100 measurements.

The objective function defined in (5) is, then, minimized using typical GA with tournament selection and tentative parameters as listed in Table 1. The attenuation factor and the phase velocity are evaluated by using the set of estimated parameters a_1, a_2 and V_o.

The obtained results are listed in Tables 2, 3, 4, 5, 6 and 7 along with the grain size and the minimization error *e* which has been calculated using Eq. (5). The minimum

Table 2 The estimated parameters for a steel specimen of grain size = 14.4 μm

Crossover operator	V_0 (m/s)	$a_1 * 10^5$ ($m^{-1}Hz^{-1}$)	$a_2 * 10^{14}$ ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
Initial	4,000	0.1	0.013		
Single-point	5,714.45	1.08493	2.03256	1.617	22
Uniform	5,714.45	1.156	0.0843	1.634	15

Table 3 The estimated Parameters for Steel-A1 specimen of grain size = 17.8 μm

Crossover operator	V_0 (m/s)	$a_1 * 10^5$ ($m^{-1}Hz^{-1}$)	$a_2 * 10^{14}$ ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
Initial	4,000	0.1	0.013		
Single-point	5,682.15	1.8	0.59	0.783	7
Uniform	5,683.95	1.8486	0.277	0.781	19

Table 4 The estimated parameters for steel-A2 specimen of grain size = 21.2 μm

Crossover operator	V_0 (m/s)	$a_1 * 10^5$ ($m^{-1}Hz^{-1}$)	$a_2 * 10^{14}$ ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
Initial	4,000	0.1	0.013		
Single-point	5,560.65	1.94	1.17	0.799	11
Uniform	5,568.64	1.925	0.1917	0.768	19

Table 5 The estimated parameters for steel-A3 specimen of grain size = 39 μm

Crossover operator	V_0 (m/s)	$a_1 * 10^5$ ($m^{-1}Hz^{-1}$)	$a_2 * 10^{14}$ ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
Initial	4,000	0.01	0.0013		
Single-point	5,332.74	2.174466	1.389	1.509	20
Uniform	5,335.94	2.254000	2.440	1.501	17

Table 6 The estimated parameters for steel-A4 specimen of grain size = 47 μm

Crossover operator	V_0 (m/s)	a_1*10^5 ($m^{-1}Hz^{-1}$)	a_2*10^{14} ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
Initial	4,000	0.01	1.3 x 10^{-4}		
Single-point	5,238.84	2.66378	3.2000	1.159	21
Uniform	5,240.14	2.66740	7.1285	1.162	24

Table 7 The estimated parameters for steel-A3 specimen for different mutation rates and uniform crossover

Mutation rate	V_0 (m/s)	a_1*10^5 ($m^{-1}Hz^{-1}$)	a_2*10^{14} ($m^{-1}Hz^{-4}$)	*e* (%)	# Iterations
0.001	5,329.64	2.05500	3.242	1.590	18
0.020	5,331.00	2.27987	3.200	1.554	22
0.100	5,335.94	2.25437	2.440	1.501	17
0.120	5,3329.94	2.15917	0.707	1.527	20
0.150	5,330.34	2.19337	0.291	1.523	19
0.250	5,334.04	2.15600	0.185	1.517	6

error (*e*) is reached after certain number of genetic generations or search iterations, this number is listed in the last column of each table.

The estimation results in Tables 2, 3, 4, 5 and 6 show a comparison between single and multi-point (uniform) crossover for different steel samples. In this comparison, the crossover rate is preserved constant and equal 0.5. Hence, the comparison shows that the uniform crossover achieves a little success over single point crossover (except a little bit for steel and steel-A4 specimens) at the expense of more computations paid during making complex mating process of uniform crossover. The mating complexity results from multi-point crossover which is implied in uniform crossover. It can also be noted that more iterations are needed in general for achieving less error rate using uniform crossover.

Another comparison is shown in Table-7 where different mutation rates are used in treating the data of the sample steel-A3 as an example. The results show that less error can be attained if the mutation rate is higher than the conventional range (0.001–0.02). The minimum errors are also obtained approximately after the same number of iterations except for steel-A4. However, the rate of 0.1 can be considered the better for this problem. Higher mutation rate of 0.5 gave higher error rate [10].

In a similar experiment, three Aluminum specimens are used. Two of them (Al1 and AL2) are from alloy 3003 and one is from alloy 1100 Which are produced at Egyptian Aluminum company (EGYPTALUM). The optimum parameters for the different test specimens are shown in Table 8 along with the minimum estimation error and the grain size of each specimen. The estimated scattering coefficient (*a2*)

Table 8 The estimated parameters for the different aluminum specimens

Specimen	V_0 (m/s)	$a_1 * 10^5$ ($m^{-1}Hz^{-1}$)	$a_2 * 10^{14}$ ($m^{-1}Hz^{-4}$)	e (%)	Grain size µm
AL1	7,014.46	1.26600	08.0000	1.2922	250
AL2	6,819.45	1.98126	17.0345	0.4700	400
AL3	6,599.95	2.38957	12.9464	1.3560	500

is comparable with that in [16] when the assumed cubic grain size dependence is considered. These results show that as the grain size increases, the attenuation factor as a whole is increased and the increase in scattering coefficient turns the attenuation factor to deviate from linearity.

7 Conclusion

The use of GA in US parameters estimation is promising since it reduces the searching-problem complexity with better accuracy. A survey shows that the tournament selection in GA is preferred because it is either equivalent to other complex selection schemes (exponential selection) or it has better behavior than other simpler scheme (roulette-wheel).

Single-point and multi-point (uniform) crossover operators are compared numerically for the problem of US parameters estimation. The comparison shows that uniform crossover operator may help in achieving less error rate in general.

Mutation operator has been tested, in a tentative experiment, with different rates. The numerical comparison of different mutation rates reveal that better results can be obtained if the mutation rate is higher than the conventional range (0.001–0.02). However, mutation rate, in this problem, is increased up to a certain limit of (0.1) and the results deteriorate after that.

Finally, the evolution parameters of GA can form infinite combinations. Accordingly, more work can disclose new results for different applications. In conclusion, the problem of US propagation identification needs more experiments to explore more efficient methodology using GA.

Acknowledgment The author would like to deeply thank Dr. A. A. Hassanien who first taught the concept of NDT at Cairo University and for his contribution to different works lead to this work.

References

1. Back, T., Hammel, U., Schwefel, H.: Evolutionary computation: Comments on the history and current state. Evol. Comput. IEEE Trans. **1**(1), 3–17 (1997)
2. Beasley, D., Martin, R., Bull, D.: An overview of genetic algorithms: part 1. Fundamentals. Univ. Comput. **15**(2), 58–69 (1993)

3. Beasley, D., Martin, R., Bull, D.: An overview of genetic algorithms: part 2. Fundamentals. Univ. Comput. **15**(4), 170–181 (1993)
4. Blickle, T., Thiele, L.: A comparison of selection schemes used in genetic algorithms. TIK-Report, no. 11, Swiss Federal Institute of Technology, (ETCH), Switzerland, Dec 1995
5. Bustillo, J., Fortineau, J., Gautier, G., Lethiecq, M.: Ultrasonic characterization of porous silicon using a genetic algorithm to solve the inverse problem. NDT & E Int. **62**, 93–98 (2014)
6. Delsanto, P.P.: Universality of nonclassical nonlinearity. In: Delsanto, S., Griffa, S., Morra, L. (eds.) Inverse Problems and Genetic Algorithms, pp. 349–366. Springer, New York (2006)
7. Elangovan, S., Anand, K., Prakasan, K.: Parametric optimization of ultrasonic metal welding using response surface methodology and genetic algorithm. Int. J. Adv. Manufact. Technol. **63** (5–8), 561–572 (2012)
8. Hassanien, A., Hesham, M., Nour El-Din, A.M.: Grain-size effect on the attenuation and dispersion of ultrasonic waves. J. Eng. Appl. Sci. **46**(3), 401–411 (1999)
9. Hesham, M.: Efficient evolution operators for estimating ultrasonic propagation parameters using genetic algorithms. Ain Shams Eng. J. **36**(2), 517–532 (2001)
10. Hesham, M., Hassanien, A.: A genetic algorithm for polycrystalline material identification using ultrasonics, Egypt. J. Phys. **31**(2), pp. 149–161 (2000)
11. http://www.obitko.com/tutorials/genetic-algorithms/
12. Kinra, V., Dayal, V.: A new technique for ultrasonic-nondestructive evaluation of thin specimens. Exp. Mech. **28**(3), 288–297 (1988)
13. Kodali, S.P., Bandaru, S., Deb, K., Munshi, P., Kishore, N.N.: Applicability of genetic algorithms to reconstruction of projected data from ultrasonic tomography. In: C. Ryan, M. Keijzer (eds.), 'GECCO', ACM, pp. 1705–1706
14. Krautkrämer, J., Krautkrämer, H.: Ultrasonic Testing of Materials. Springer, Berlin (1969)
15. Kristinsson, K., Dumont, G.A.: System identification and control using genetic algorithms. Syst. Man Cybern. IEEE Trans. **22**(5), 1033–1046 (1992)
16. Kuttruff, H.: Ultrasonics Fundamentals and Applications. Elsevier Applied Science, London (1991)
17. Luo, Z., Zhu, H., Chu, J., Shen, L., Hu, L.: Strain measurement by ultrasonic speckle technique based on adaptive genetic algorithm. J. Strain Anal. Eng. Des. **48**(7), 446–456 (2013)
18. Nour-El-Din, A.M.: Attenuation and dispersion of ultrasonic waves in metals. M.Sc. thesis, Faculty of Engineering, Cairo University, May 1997
19. O'Donnell, M., Jaynes, E., Miller, J.: Kramers-Kronig relationship between ultrasonic attenuation and phase velocity. J. Acoust. Soc. Am. **69**(3), 696–701 (1981)
20. Peirlinckx, L., Pintelon, R., Van Biesen, L.: Identification of parametric models for ultrasonic wave propagation in the presence of absorption and dispersion. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **40**(4), 302–312 (1993)
21. Puthillath, P., Krishnamurthy, C., Balasubramaniam, K.: Hybrid inversion of elastic moduli of composite plates from ultrasonic transmission spectra using PVDF plane wave sensor. Compos. B Eng. **41**(1), 8–16 (2010)
22. Ramuhalli, P., Polikar, R., Udpa, L., Udpa, S.S.: Fuzzy ARTMAP network with evolutionary learning, Acoustics, Speech, and Signal Processing. In: Proceedings of IEEE International Conference on ICASSP'00, 6, pp. 3466–3469 (2000)
23. Rosenberger, C., Chehdi, K.: Genetic fusion: application to multi-components image segmentation, Acoustics, Speech, and Signal Processing. In: Proceedings of IEEE International Conference on ICASSP'00, 4, pp. 2223–2226 (2000)
24. Sibil, A., Godin, N., R'Mili, M., Maillet, E., Fantozzi, G.: Optimization of acoustic emission data clustering by a genetic algorithm method. J. Nondestr. Eval. **31**(2), 169–180 (2012)
25. Spears, W.M.: Adapting crossover in a genetic algorithm, Navy Center for Applied Research in Artificial Intelligence, (NCARAI) Naval Reaserch Lab., pp. 20375–5000, Washington, DC (1992)

26. Sun, K., Hong, K., Yuan, L., Shen, Z., Ni, X.: Inversion of functional graded materials elastic properties from ultrasonic lamb wave phase velocity data using genetic algorithm. J. Nondestr. Eval. **33**, 34–42 (2013)
27. Tavakolpour, A.R., Mat Darus, I.Z., Tokhi, O., Mailah, M.: Genetic algorithm-based identification of transfer function parameters for a rectangular flexible plate system. Eng. Appl. Artif. Intell. **23**(8), 1388–1397 (2010)
28. Toledo, A.R.; Fernández, A.R.; Anthony, D. K.: A comparison of GA objective functions for estimating internal properties of piezoelectric transducers used in medical echo-graphic imaging. Health Care Exchange (PAHCE), 2010 Pan American, vol. 185(190), pp. 15–19, Mar 2010
29. Vishnuvardhan, J., Krishnamurthy, C., Balasubramaniam, K.: Genetic algorithm reconstruction of orthotropic composite plate elastic constants from a single non-symmetric plane ultrasonic velocity data. Compos. B Eng. **38**(2), 216–227 (2007)
30. Weile, D.S., Michielssen, E.: Genetic algorithm optimization applied to electromagnetics: a review. Antennas Propag. IEEE Trans. **45**(3), 343–353 (1997)

A Hybridized Approach for Prioritizing Software Requirements Based on K-Means and Evolutionary Algorithms

Philip Achimugu and Ali Selamat

Abstract One of the major challenges facing requirements prioritization techniques is accuracy. The issue here is lack of robust algorithms capable of avoiding a mismatch between ranked requirements and stakeholder's linguistic ratings. This problem has led many software developers in building systems that eventually fall short of user's requirements. In this chapter, we propose a new approach for prioritizing software requirements that reflect high correlations between the prioritized requirements and stakeholders' linguistic valuations. Specifically, we develop a hybridized algorithm which uses preference weights of requirements obtained from the stakeholder's linguistic ratings. Our approach was validated with a dataset known as RALIC which comprises of requirements with relative weights of stakeholders.

Keywords Software · Requirements · Prioritization · Stakeholders · RALIC

1 Introduction

Currently, most software development organizations are confronted with the challenge of implementing ultra-large scale system especially in the advent of big data. This has led to the specification of large number of software requirements during the elicitation life cycle. These requirements contain useful information that will satisfy the need of the users or project stakeholders. Unfortunately, due to some crucial challenges involved in developing robust systems such as inadequate skilled programmers, limited delivery time and budget among others; requirements

---

P. Achimugu · A. Selamat (✉)
Department of Software Engineering, Faculty of Computing,
Universiti Teknologi Malaysia, Faculty of Computing,
Skudai 81310, Johor, Malaysia
e-mail: aselamat@utm.my

P. Achimugu
e-mail: check4philo@gmail.com

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_4

prioritization has become a viable technique for ranking requirements in order to plan for software release phases. During prioritization, requirements are classified according to their degrees of importance with the help of relative or preference weights from project stakeholders.

Data mining is one of the most effective and powerful techniques that can be used to classify information or data objects to enhance informed decision making. Clustering is a major data mining task that has to do with the process of finding groups or clusters or classes in a set of observations such that those belonging to the same group are similar, while those belonging to different groups are distinct, according to some criteria of distance or likeness. Cluster analysis is considered to be unsupervised learning because it can find and recognise patterns and trends in a large amount of data without any supervision or previously obtained information such as class labels. There are many algorithms that have been proposed to perform clustering.

Generally, clustering algorithms can be classified under two major categories [12]: hierarchical algorithms and partitional algorithms. Hierarchical clustering algorithms have the capacity of recursively identifying clusters either in an agglomerative (bottom-up) mode or in a divisive (top-down) mode. An agglomerative method is initiated with data objects in a separate cluster, which decisively merge the most similar pairs until termination criteria are satisfied. Divisive methods on the other hand deals with data objects in one cluster and iteratively divide each cluster into smaller clusters, until termination criteria are met. However, a partitional clustering algorithm concurrently identifies all the clusters without forming a hierarchical structure. A well-known type of partitional clustering algorithms is the centre-based clustering method, and the most popular and widely used algorithm from this class of algorithms is known as K-means algorithm. K-means is relatively easy to implement and effective most of the time [7, 13]. However, the performance of k-means depends on the initial state of centroids which is likely to converge to the local optima rather than global optima. The k-means algorithm tries to minimise the intra-cluster variance, but it does not ensure that the result has a global minimum variance [14].

In recent years, many heuristic techniques have been proposed to overcome this problem. Some of which include: a simulated annealing algorithm for the clustering problem [28]; a tabu search based clustering algorithm [2, 29]; a genetic algorithm based clustering approach [23]; a particle swarm optimization based clustering technique [19] and a genetic k-means based clustering algorithm [21] among others. A major reported limitation of K-means algorithm has to do with the fact that, the number of clusters must be known prior to the utilization of the algorithm because it is required as input before the algorithm can run. Nonetheless, this limitation has been addressed by some couple of authors [3, 8, 9, 26].

In this research, we propose the application of a hybridized algorithm based on evolutionary and k-means algorithms on cluster analysis during requirements prioritization. The performance of the proposed approach has been tested on standard and real datasets known as RALIC. The rest of the paper is organised as follows: Sect. 2 provides a brief background on requirement prioritization. Section 3 describes k-means and evolutionary algorithms. In Sect. 4, we present our proposed algorithm for solving requirement prioritization problem using evolutionary algorithm and k-means algorithm. Experimental results are discussed in Sect. 5. Finally, Sect. 6 presents the conclusions of this research with ideas for future work.

2 Software Requirements Prioritization

During requirement elicitation, there are more prospective requirements specified for implementation by relevant stakeholders with limited time and resources. Therefore, a meticulously selected set of requirements must be considered for implementation and planning for software releases with respect to available resources. This process is referred to as requirements prioritization. It is considered to be a complex multi-criteria decision making process [25].

There are so many advantages of prioritizing requirements before architecture design or coding. Prioritization aids the implementation of a software system with preferential requirements of stakeholders [1, 32]. Also, the challenges associated with software development such as limited resources, inadequate budget, insufficient skilled programmers among others makes requirements prioritization really important [15]. It can help in planning software releases since not all the elicited requirements can be implemented in a single release due to some of these challenges [5, 16]. It also enhances budget control and scheduling. Therefore, determining which, among a pool of requirements to be implemented first and the order of implementation is necessary to avoid breach of contract or agreement during software development. Furthermore, software products that are developed based on prioritized requirements can be expected to have a lower probability of being rejected. To prioritize requirements, stakeholders will have to compare them in order to determine their relative importance through a weight scale which is eventually used to compute the ranks [20]. These comparisons become complex with increase in the number of requirements [18].

Software system's acceptability level is mostly determined by how well the developed system has met or satisfied the specified requirements. Hence, eliciting and prioritizing the appropriate requirements and scheduling right releases with the correct functionalities are critical success factors for building formidable software systems. In other words, when vague or imprecise requirements are implemented, the resulting system will fall short of user's or stakeholder's expectations. Many software development projects have enormous prospective requirements that may be practically impossible to deliver within the expected time frame and budget [31]. It therefore becomes highly necessary to source for appropriate measures for planning and rating requirements in an efficient way.

Many requirements prioritization techniques exist in the literature. All of these techniques utilize a ranking process to prioritize candidate requirements. The ranking process is usually executed by assigning weights across requirement based on pre-defined criteria, such as value of the requirement perceived by relevant stakeholders or the cost of implementing each requirement. From the literature;

analytic hierarchy process (AHP) is the most prominently used technique. However, this technique suffers bad scalability. This is due to the fact that, AHP executes ranking by considering the criteria that are defined through an assessment of the relative priorities between pairs of requirements. This becomes impracticable as the number of requirements increases. It also does not support requirements evolution or rank reversals but provide efficient or reliable results [4, 17]. Also, most techniques suffer from rank reversals. This term refers to the inability of a technique to update rank status of ordered requirements whenever a requirement is added or deleted from the list. Prominent techniques that suffer from this limitation are case base ranking [25]; interactive genetic algorithm prioritization technique [31]; Binary search tree [17]; cost value approach [17] and EVOLVE [11]. Furthermore, existing techniques are prone to computational errors [27] probably due to lack of robust algorithms. [17] conducted some researches where certain prioritization techniques were empirically evaluated. From their research, they reported that, most of the prioritization techniques apart from AHP and bubble sorts produce unreliable or misleading results while AHP and bubble sorts were also time consuming. The authors submitted that; techniques like hierarchy AHP, spanning tree, binary search tree, priority groups produce unreliable results and are difficult to implement. [4] were also of the opinion that, techniques like requirement triage, value intelligent prioritization and fuzzy logic based techniques are also error prone due to their reliance on experts and are time consuming too. Planning game has a better variance of numerical computation but suffer from rank reversals problem. Wieger's method and requirement triage are relatively acceptable and adoptable by practitioners but these techniques do not support rank updates in the event of requirements evolution as well. The value of a requirement is expressed as its relative importance with respect to the other requirements in the set.

In summary, the limitations of existing prioritization techniques can be described as follows:

2.1.1 Scalability: Techniques like AHP, pairwise comparisons and bubblesort suffer from scalability problems because, requirements are compared based on possible pairs causing n (n−1)/2 comparisons [17]. For example, when the number of requirements is doubled in a list, other techniques will only require double the effort or time for prioritization while AHP, pairwise comparisons and bubblesort techniques will require four times the effort or time. This is bad scalability.

2.1.2 Computational complexity: Most of the existing prioritization techniques are actually time consuming in the real world [4, 17]. Furthermore, [1] executed a comprehensive experimental evaluation of five different prioritization techniques namely; AHP, binary search tree, planning game, $100 (cumulative voting) and a new method which combines planning game and AHP (PgcAHP), to determine their ease of use, accuracy and scalability. The author went as far as determining the average time taken to prioritize 13 requirements across 14 stakeholders with these techniques. At the end of the experiment; it was observed that, planning game was the fastest while AHP was the slowest. Planning game prioritized 13 requirements in about 2.5 min while AHP prioritized the same number of requirements in about 10.5 min. In other words, planning game technique took only 11.5 s to compute the priority scores of one requirement across 14 stakeholders while AHP consumed 48.5 s to accomplish the same task due to pair comparisons.

2.1.3 Rank updates: This is defined as 'anytime' prioritization [25]. It has to do with the ability of a technique to automatically update ranks anytime a requirement is included or excluded from the list. This situation has to do with requirements evolution. Therefore, existing prioritization techniques are incapable of updating or reflecting rank status whenever a requirement is introduced or deleted from the rank list. Therefore, it does not support iterative updates. This is very critical because, decision making and selection processes cannot survive without iterations. Therefore, a good and reliable prioritization technique will be one that supports rank updates. This limitation seems to cut across most existing techniques.

2.1.4 Communication among stakeholders: Most prioritization techniques do not support communication among stakeholders. One of the most recent works in requirement prioritization research reported communication among stakeholders as part of the limitations of their technique [25]. This can lead to the generation of vague results. Communication has to do with the ability of all relevant stakeholders to fully understand the meaning and essence of each requirement before prioritization commences.

2.1.5 Requirements dependencies: This is a crucial attribute that determines the reliability of prioritized requirements. These are requirements that depend on another to function. Requirements that are mutually dependent can eventually be merged as one; since without one, the other cannot be implemented. Prioritizing such requirements may lead to erroneous or redundant results. This attribute is rarely discussed among prioritization research authors. However, dependencies can be detected by mapping the pre and post conditions from the whole set of requirements, based on the contents of each requirement [24]. Therefore, a good prioritization technique should cater or take requirements dependences into cognizance before initiating the process.

2.1.6 Error proneness: Existing prioritization techniques are also prone to errors [27]. This could be due to the fact that, the rules governing the requirements prioritization processes in the existing techniques are not robust enough. This has also led to the generation of unreliable prioritization results because; such results do not reflect the true ranking of requirements from stakeholder's point of view or assessment after the ranking process. Therefore robust algorithms are required to generate reliable prioritization results.

3 K-Means and Evolutionary Algorithms

K-Means algorithm is one of the most popular types of unsupervised clustering algorithm [6, 10] which is usually used in data mining, pattern recognition and other related researches. It is aimed at minimizing cluster performance indexes, square-error and error criteria. The concept of this algorithm borders on the identification of *K* clusters that satisfies certain criteria. Usually, to demonstrate the applicability of *K*-means algorithm, some data objects are chosen to represent the initial cluster focal points and secondly, the rest of the data objects are assembled to their focal points based on the criteria of minimum distance which will eventually lead to the computation of the initial clusters. However, if these clusters are unreasonable, it is easily modified by re-computing each cluster's focal point. This process is repeatedly iterated until reasonable clusters are obtained. In the context of requirement prioritization, the numbers of clusters are likened to the number of requirements while the data objects are likened to the attributes describing the expected functionalities of a particular requirement. Therefore, *K*-Means algorithm is initiated with some random or heuristic-based centroids for the desired clusters and then assigns every data object to the closest centroid. After that, the k-means algorithm iteratively refines the current centroids to reach the (near) optimal ones by calculating the mean value of data objects within their respective clusters. The algorithm will terminate when any one of the specified termination criteria is met (i.e., a predetermined maximum number of iterations is reached, a (near) optimal solution is found or the maximum search time is reached).

Inversely, the evolutionary algorithm (EA) is best illustrated with principles of differential evolution algorithms, by considering requirements as individuals. After generating the initial solution, its requirements are stored into a pool, which forms the initial population in the tournament. Thereafter, the requirements are categorized into various classes containing its respective attributes. The weights of attributes between two requirements are randomly selected for computation until the entire weights are exhausted. Meaning, weights of attributes from each requirement are mated for crossover. For instance, assuming we have two requirements *X* and *Y* with respective attributes as $\langle \mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n} \rangle$ and $\langle \mathrm{y}_1, \ldots, \mathrm{y}_\mathrm{n} \rangle$. These two attributes are considered as prospective couple or parent. So, for each couple, we randomly extract crossover point, for instance 2 for the first and 5 for the second. The final step is to apply random mutation. These processes are performed from the first to the last requirement. Once a pair (combination) of requirement is selected, one out of the four local search operators is applied randomly based on stakeholder's weights. Finally, offspring (requirements) generated by these crossover operators are mutated according to the stakeholder's weights. Mutation is specifically achieved by selecting randomly one out of two operators according to the weights distribution. Selecting each possible pair of requirement is based on a random order. Mating and mutation operators are repeatedly applied for a certain number of generations, and finally a feasible solution is constructed using the requirements in the pool. To guarantee a feasible solution, recombination and mutation operators of

EA are not allowed to lose any customer i.e. the offspring must contain the same number of customers as the parent, otherwise parent are stored into the requirement pool instead of offspring.

4 Proposed Approach

Our approach is based on the relative weights provided by project stakeholders of a software development project. The evolutionary algorithm works as a hyper-heuristics which assigns different coefficient values to the relative scores obtained from the pool of functions included in the system. At the beginning of the process, all functions (or metrics) evenly contribute to the calculation of the relative cumulative values of a specific requirement. The system evolves so that the requirements which provide the most valued weights across the relevant stakeholders have the highest coefficients. The differential evolution (DE) algorithm [30] was chosen among other candidates because, after a preliminary study, we conclude that DE obtained very competitive results for the problem under consideration. The reason lies in how the algorithm makes the solutions evolve. Our system can be considered as a hyper-heuristics which uses differential evolution to assign each requirement with a specific coefficient. These values show the relative importance of each requirement. Differential evolution performs the search of local optima by making small additions and subtractions between the members of its population. This feature is capable of solving rank reversals problem during requirement prioritization since the algorithm works with the weights provided by the stakeholders. In fact, the stakeholder is defined as an array of floating point values, s, where s(fx) is the coefficient which modifies the result provided by the learning function fx. We consider a finite set of collection of requirements $X = \{R_{11}, R_{12}...R_{1k}\}$ that has to be ranked against $Y = \{R_{21}, R_{22}...R_{2k}\}$. Our approach consist of set of input R_{11}, R_{12}, ..., R_{1k}, associated with their respective weights w_1, w_2, ..., w_k that represents stakeholders' preferences and a fitness value function required to calculate the similarity weights across requirements. The requirement ($R_{11...}$, $R_{21...}$,..., R_{nk}) represent input data that are ranked using the fitness function on similarity scores and stored in the database. In this approach, we assume that, the stakeholder's preferences are expressed as weights, which are values between 5 and 1. These weights are provided by the stakeholders. In this research, one of our objectives is to automatically rank stakeholder's preferences using EA based learning. The designed algorithm will compute the ranks of requirements based on a training data set. To rank the preferential weights of requirements across relevant stakeholders, there is need to identify the following: a ranking technique for the best output and a measure of quality that aids the evaluation of the proposed algorithm.

In practical application of the learning process, $X = (r_1, r_2, \ldots, r_n)$; $Y = (r'_1, r'_2, \ldots, r'_n)$ probably represents two requirements with their respective attributes that are to be ranked. For each requirement, attributes are not necessarily mutually independent. In order to drive the synthetic utility values, we first exploited the factor analysis technique to extract the attributes that possess common functionalities. This caters for requirement dependencies challenges during the prioritization process. The attributes with the same functionalities are considered to be mutually dependent. Therefore, before relative weights are assigned to the requirements by relevant stakeholders, attention should be paid to requirement dependencies issues in order to avoid redundant results. However, when requirements evolve, it becomes necessary to add or delete from a set. The algorithm should also be able to detect this situation and update rank status of ordered requirements instantly. This is known as rank reversals. It is formally expressed as: (1) failure of the type $1 \rightarrow 5$ or $5 \rightarrow 1$; (2) failuresz of the type $1 \rightarrow \phi$ or $5 \rightarrow \phi$ (where ϕ = the null string) (called *deletions*); and (3) failures of the type $\phi \rightarrow 1$ or $\phi \rightarrow 5$ (called *insertions*). A weight metric w, on two requirement (X, Y) is defined as the smallest number of edit operations (deletions, insertions and updates) to enhance the prioritization process. Three types of rank updates operations on $X \rightarrow Y$ are defined as: a *change* operation ($X \neq \phi$ and $Y \neq \phi$), a *delete* operation ($Y = \phi$) and an *insert* operation ($X = \phi$). The weights of all the requirements can be computed by a *weight function* w. An arbitrary weight function w is obtained by computing all the assigned non-negative real number $w(X, Y)$ on each requirement. On the other hand, there is mutual independence between attributes, and the measurement is an additive case, so we can utilize the additive aggregate method to conduct the synthetic utility values for all the attributes in the entire requirements. As we can see in Algorithm 1, differential evolution starts with the generation of random population (line 1) through the assignment of a random coefficient to each attribute of the individual (requirement). The population consists of a certain number of solutions (this is, a parameter to be configured). Each individual (requirement) is represented by a vector of weighting factors provided by the stakeholders. After the generation of the population, the fitness of each individual is assigned to each solution using the Pearson correlation. This correlation, corr(X, Y), is calculated with the scores provided by stakeholders for every pair of requirement of the RALIC dataset [22].

The closer the value of correlation is to any of the extreme values, the stronger is the correlation between the requirements and the higher is the accuracy of the prioritized results. On the contrary, if the result tends toward 0, it means that the requirements are somewhat uncorrelated which gives an impression of poor quality prioritized solution.

Algorithm 1: Pseudo-code for the DE/K-means algorithm

```
1. generateRandom centroids from K clusters (population)
2. assign weights of each attribute to the cluster with the closest centroid
3. update the centroids by calculating the mean values of objects within clusters:
   Fitness (population)
4. while (stop condition not reached) do
5. for (each individual of the population)
6. selectIndividuals (xTarget, xBest, xInd1, xInd2)
7. xMut diffMutation (xBest, F, xInd1, xInd2)
```

```
 8. xTrial binCrossOver (xTarget, xMut, CrossProb)
 9. calculateFitness (xTrial)
10. updateIndividual (xTarget, xTrial)
11. endfor
12. endwhile
13. return bestIndividual (population)
```

From Algorithm 1, the main loop begins after evaluating the whole population (line 2). It is important to note that differential evolution and k-means algorithms are iterative in nature. This becomes very necessary when uninterrupted generations attempt to obtain an optimal solution and terminates when the maximum number of generations is reached (line 4). We initiated the process by selecting four parameters (line 6). xTarget and xBest are the parameters being processed. The former stands for the weights provided by project stakeholders and the latter stands for the prioritized weights for two randomly chosen requirements denoted as xInd1 and xInd2 respectively. Next, mutation is performed (line 7) according to the expression: xMut xBest + F (xInd1 _ xInd2) in order to determine sum of the cumulative relative weights of requirements across the total number of stakeholders involved in the software development project. This task execute in twofold: (diffMutation 1 and 2). The first phase has to do with the calculation of xDiff across project stakeholders with the help of expression: xDiff F (xInd1 _ xInd2). xDiff represents the mutation to be applied to compute the relative weights of requirements in order to calculate the best solution. Subsequently, the modification of each attribute of xBest in line with the mutation indicated in xDiff give rise to xMut. At the end of the mutation process, xTarget and xMut individuals are intersected (line 8) using binary crossover with the concept of crossover probability, crossProb. Then, the obtained individual, xTrial, is evaluated to determine its accuracy (line 8) which is compared against xTarget. The evaluation procedures of the prioritization process encompass the establishment of fitness function, disclosure of agreement and disagreement indexes, confirmation of credibility degree, and the ranking of attributes/requirements. These data are represented by weights reflecting the subjective judgment of stakeholders. The best individual is saved in the xTarget position (line 10). This process is iterated for each individual in the population (line 5) especially when the stoppage criteria are not met (line 4). However, in the context of this research, the stoppage criteria are certain number of generations which is also set during the configuration of the parameters. At the end of the process, the best individuals (most valued requirements) are returned as the final results of the proposed system (line 13). It is important to note here that results have been obtained after a complete experimental process using RALIC dataset.

Based on the above description, the proposed algorithm is built on three main steps. In the first step, EA-KM applies k-means algorithm on the selected dataset and tries to produce near optimal centroids for desired clusters. In the second step, the proposed approach produces an initial population of solutions, which will be applied by the EA algorithm in the third step. The generation of an initial population is carried out in several different ways. These could be through candidate solutions generated by the output of the k-means algorithm or randomly generated. The process generates a high-quality initial population, which will be used in the next step by the EA algorithm. Finally, in the third step, EA will be employed for determining an optimal solution for the clustering-based prioritization problem. To represent candidate solutions in the proposed algorithm, we used one-dimensional array to encode the centroids of the desired clusters. The length of the arrays is equal to d * k, where d is the dimensionality of the dataset under consideration or the number of features that each data object has and k is the number of clusters.

The optimal value of the fitness function $J(w, \mathbf{x}, \mathbf{c})$ is determined by the following prioritization equations:

$$\mathbf{x}(w) = f(w, \mathbf{x}, \mathbf{c}), \quad \mathbf{x}(0) = \mathbf{x}_0(\mathbf{c}(0)) \tag{1}$$

And the set of constraints

$$\begin{cases} g_i(\mathbf{x}, \mathbf{c}) = 0 & \text{for} \quad j = 1, \ldots, E \\ g_j(\mathbf{x}, \mathbf{c}) \geq 0 & \text{for} \quad j = E + 1, \ldots, S \end{cases} \tag{2}$$

where, $\mathbf{x} \in R^m$ is the requirement set described by the function $f \in R^m$; $\mathbf{c} \in R^n$ is the criteria used to determine the relative weights of requirements. The optimal value of $J(w, \mathbf{x}, \mathbf{c})$ is achieved by varying the criteria $c_i(w),\ i = \overline{1, n}$ within the boundaries specified by (1) and (2). All functions here are to be regarded as discrete, obtained through the relative weights $w_1, w_2, \ldots, w_n$ provided by the stakeholders. During prioritization, the state of the requirement at weight w is conventionally described by the number of stakeholders $N(s)$. Therefore, the prioritization process in this case is one-dimensional $(m = 1)$, and the Eq (2) becomes:

$$\frac{\Delta N(s)}{\Delta w} = f(N) - \kappa(c)c(w)N, \quad N(0) = N_0 \tag{3}$$

Where $f(N)$ is a real-valued function which models the increase in the number of requirements; $c(w)$ is the weight of requirements based on pre-defined criteria; $\kappa(c)$ is a quality representing the efficacy of the ranked weights. The rank criteria $c(w)$ in (3), is the only variable directly controllable by the stakeholders. Therefore, the problem of requirements prioritization can be regarded as the problem of planning for software releases based on the relative weights of requirements. The optimal weights of requirements are in the form of a discrete ordered program with N requirements given at weights $w_1, w_2, \ldots, w_n$. Each requirement is assessed by s stakeholders characterized by their defined criteria $C_{ij}, i = \overline{1, n}, j = \overline{1, d}$ in the set. These criteria can be varied within the boundaries specified by the constraints in Eq. (3). The conflicting nature of these constraints and the intention to develop a model-independent approach for prioritizing requirements makes the utilization of computational optimization techniques very viable.

All the experiments were executed under the same environment: an Intel Pentium 2.10 GHz processor and 500 GB RAM. Since we are dealing with a stochastic algorithm, we have carried out 50 independent runs for each experiment. Results provided in the following subsections are average results of these 50 executions. Arithmetic mean and standard deviation were used to statistically measure the performance of the proposed system.

5 Experimental Results and Discussion

The experiments described in this research considered the likelihood of calculating preference weights of requirements provided by stakeholders so as to compute their ranks. The RALIC dataset was used for validating the proposed approach. The PointP and RateP portions of the dataset were used, which consist of about 262 weighted attributes spread across 10 requirement sets from 76 stakeholders. RALIC stands for replacement access, library and ID card. It was a large-scale software project initiated to replace the existing access control system at University College London [22].

The dataset is available at: http://www.cs.ucl.ac.uk/staff/S.Lim/phd/dataset.html. Attributes were ranked based on 5-point scale; ranging from 5 (highest) to 1 (lowest). As a way of pre-processing the dataset, attributes with missing weights were given a rating of zero.

For the experiment, a Gaussian Generator was developed, which computes the mean and standard deviation of given requirement sets. It uses the Box-Muller transform to generate relative values of each cluster based on the inputted stakeholder's weights. The experiment was initiated by specifying a minimum and maximum number of clusters, and a minimum and maximum size for attributes. It then generates a random number of attributes with random mean and variance between the inputted parameters. Finally, it combines all the attributes into one and computes the overall score of attributes across the number of clusters k. The algorithm defined earlier attempts to use these combined weights of attributes in each cluster to rank each requirement. For the k-means algorithm to run, we filled in the variables/observations table which has to do with the two aspect of RALIC dataset that was utilized (PointP and RateP), followed by the specification of clustering criterion (Determinant W) as well as the number of classes. The initial partition was randomly executed and ran 50 times. The iteration completed 500 cycles and the convergence rate was at 0.00001. As an initialization step, the DE algorithm generated a random set of solutions to the problem (a population of genomes). Then it enters a cycle where fitness values for all solutions in a current population are calculated, individuals for mating pool are selected (using the operator of reproduction), and after performing crossover and mutation on genomes in the mating pool, offspring are inserted into a population. In this research, elitism was performed to save the chromosomes of the old solution so that crossover and mutation can re-occur for new solutions. Thus a new generation is obtained and the process begins again. The process stops after the stopping criteria are met, i.e. the "perfect" solution is recognized, or the number of generations has reached its maximum value. From each generation of individuals, one or few of them, that has the highest fitness values are picked out, and inserted into the result set.

The weights of requirements are computed based on their frequencies and a mean score is obtained to determine the final rank. Figure 1 depicts the fitness function for the mean weights of the dataset across 76 stakeholders. This is achieved by counting the numbers of requirements, where the DE simply add their sums and apportion precise values across requirements to determine their relative importance.

The results displayed in Table 1 shows the summary statistics of 50 experimental runs. For 10 requirements, the total number of attributes was 262 and the size of each cluster varied from 1 to 50 while, the mean and standard deviation of each cluster spanned from 1–30 and 15–30, respectively.

Also, Table 2 shows the results provided by each cluster that represents the 10 requirements during the course of running the algorithm on the data set. It displays the sum of weights, within-class variances, minimum distance to the centroid, average distance to the centroid and maximum distance to the centroids. Table 3 shows the distances between the class centroids for the 10 requirements across the total number of attributes while, Table 4 depict the analysis of each iteration. Analysis of multiple runs of this experiment showed exciting results as well. Using 500 trials, it was discovered that, the algorithm classified requirements correctly,

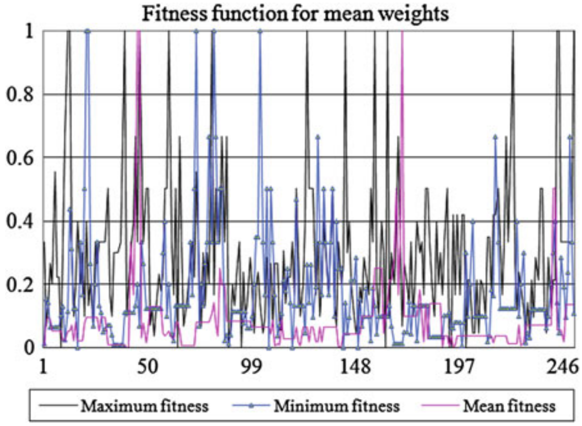


Fig. 1 Fitness function for mean weights

Table 1 Summary statistics

Variables	Obs.	Obs. with missing data	Obs. without missing data	Min	Max	Mean	Std. deviation
Point P	262	0	262	2.083	262	28.793	24.676
Rate P	262	0	262	0.000	262	5.123	15.864

Obs. = Objects

Table 2 Results by class

Class	1	2	3	4	5	6	7	8	9	10
Sum of weights	53	61	31	14	27	29	12	30	8	1
Within-class variance	7.302	8.283	37.897	172.896	2.393	12.699	3.607	1.992	1.190	0.000
Min. distance to centroid	0.532	0.365	0.352	3.769	0.695	0.232	0.663	0.253	0.412	0.000
Ave. distance to centroid	2.518	2.673	5.166	11.814	1.233	2.863	1.491	1.149	0.925	0.000
Max. distance to centroid	5.174	5.646	13.838	15.693	4.412	6.395	4.618	3.175	1.604	0.000

Min. = Minimum, Ave = Average, Max. = Maximum

where the determinants (W) for each variable were computed based on the stakeholder's weights. The sum of weights and variance for each requirement set was also calculated.

The learning process consists of finding the weight vector that allows the choice of requirements. The fitness value of each requirement can be measured on the basis of the weights vectors based on pre-defined criteria used to calculate the actual ranking. The disagreements between ranked requirements must be minimized as much as possible. We can also consider the disagreements on a larger number of top vectors, and the obtained similarity measure which can be used to enhance the agreement index. The fitness value will then be a weighted sum of these two similarity measures.

Definition 5.1 Let X be a measurable requirement that is endowed with attributes of σ-functionalities, where N is all subsets of X. A learning process g defined on the measurable space (X, N) is a set function $g : N \rightarrow [0, 1]$ which satisfies the following properties:

$$g(\phi) = 0, \; g(X) = 1 \tag{4}$$

But for requirements *X, Y;* the learning process equation will be:

$$X, N \subseteq Y \in N \rightarrow [0, 1] \tag{5}$$

From the above definition, *X*, *Y*, *N*, *g* are said to be the parameters used to measure or determine the relative weights of requirement. This process is monotonic. Consequently, the monotonicity condition is obtained as:

$$g(X \cup Y) \geq \max\{g(X), g(Y)\} \text{ and } g(X \cap Y) \leq \min\{g(X), g(Y)\} \tag{6}$$

Table 3 Distances between class centroids

Class	1	2	3	4	5	6	7	8	9	10
1	0	9.702	35.486	68.298	7.053	22.502	3.929	12.849	10.292	441.143
2	9.702	0	45.180	77.989	16.745	32.193	12.148	22.536	19.985	446.808
3	35.486	45.180	0	32.812	28.436	12.987	33.529	22.644	25.196	422.664
4	68.298	77.989	32.812	0	61.247	45.797	66.249	55.453	58.008	407.820
5	7.053	16.745	28.436	61.247	0	15.450	5.847	5.796	3.241	437.356
6	22.502	32.193	12.987	45.797	15.450	0	20.614	9.657	12.212	429.318
7	3.929	12.148	33.529	66.249	5.847	20.614	0	11.135	8.780	441.935
8	12.849	22.536	22.644	55.453	5.796	9.657	11.135	0	2.568	434.397
9	10.292	19.985	25.196	58.008	3.241	12.212	8.780	2.568	0	435.539
10	441.143	446.808	422.664	407.820	437.356	429.318	441.935	434.397	435.539	0

Table 4 Statistics for each iteration

Iteration	Within-class variance	Trace (W)	Determinant (W)	Wilks'Lambda
0	1,128.978	289,018.282	3.00114E + 12	0.899
1	74.074	18,962.929	106503842.6	0.000
2	29.394	7,524.991	33285074.33	0.000
3	20.151	5,158.594	22195537.74	0.000

In the case where $g(X \cup Y) \geq \max\{g(X), g(Y)\}$, the learning function g attempts to determine the total number of requirements being prioritized and if$g(X \cap Y) \leq \min\{g(X), g(Y)\}$, the learning function attempts to compute the relative weights of requirements provided by the relevant stakeholders.

Definition 5.2 Let $h = \sum_{i=1}^{n} X_i.1_{X_i}$ be a simple function, where 1_{X_i} is the attribute function of the requirements $X_i \in N, i = 1, \ldots, n$; X_i are pairwise disjoints, but if $M(X_i)$ is the measure of the weights between all the attributes contained in X_i, then the integral of h is given as:

$$\int h.dM = \sum_{i=1}^{n} M(X_i)._{X_i} \tag{7}$$

Definition 5.3 Let X, Y, N, g be the measure of weights between two requirements, the integral of weights measure $g : N \rightarrow [0, 1]$ with respect to a simple function h is defined by:

$$\int h(r)g(r) = \vee(h(r_i) \wedge g(X_i)) = \max\ \min\{r'_i, g(Y_i)\} \tag{8}$$

Where $h(r_i)$ is a linear combination of an attribute function $1r_i$ such that $X_1 \subset Y_1 \subset \ldots \subset X_n \subset Y_n$ and $X_n = \{r|h(r) \geq Y_n\}$.

Definition 5.4 Let X, N, g be a measure space. The integral of a measure of weights by the learning process $g : N \rightarrow [0, 1]$ with respect to a simple function h is defined by

$$\int h(r).dg \cong \sum [h(r_i) - h(r_{i-1})].g(X_i) \tag{9}$$

Similarly, if Y, N, g is a measure space; the integral of the measure of the weights with respect to a simple function h will be:

$$\int h(r').dg \cong \sum [h(r'_i) - h(r'_{i-1})].g(Y_i) \tag{10}$$

However, if *g* measures the relative weights of requirements, defined on a power set *P(x)* and satisfies the definition 5.1 as above; the following attribute is evident:

$$\forall X, Y \in P(x), X \cap Y = \phi \Rightarrow g_2(X \cup Y) = g_2(X) + g_2(Y) + \lambda g_2(X) g_2(Y) \quad (11)$$

For $0 \leq \lambda \leq \infty$

In practical application of the learning process, the number of cluster which represents the number of requirements must be determined first. The attributes that describes each requirement are known as the data elements that are to be ranked. Therefore, before relative weights are assigned to requirements by stakeholders, attention should be paid to requirement dependencies issues in order to avoid redundant results. Prioritizing software requirements is actually determined by relative perceptions which will inform the relative scores provided by the stakeholders to initiate the ranking process.

Prioritizing requirements is an important activity in software development [31, 32]. When customer expectations are high, delivery time is short and resources are limited, the proposed software must be able to provide the desired functionality as early as possible. Many projects are challenged with the fact that, not all the requirements can be implemented because of limited time and resource constraints. This means that, it has to be decided which of the requirements can be removed for the next release. Information about priorities is needed, not just to ignore the least important requirements but also to help the project manager resolve conflicts, plan for staged deliveries, and make the necessary trade-offs. Software system's acceptability level is frequently determined by how well the developed system has met or satisfied the specified user's or stakeholder's requirements. Hence, eliciting and prioritizing the appropriate requirements and scheduling right releases with the correct functionalities are essential ingredients for developing good quality software systems. The matlab function used for k-means clustering which is idx = k means (data, k), that partitions the points in the n-by-p data matrix data into k clusters was employed. This iterative partitioning minimizes the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Rows of data corresponds to attributes while columns corresponds to requirements. K-means returns an n-by-1vector idx containing the cluster indices of each attribute which is utilized in calculating the fitness function to determine the ranks.

Further analysis was performed using a two-way analysis of variance (ANOVA). On the overall dataset, we found significant correlations between the ranked requirements. The results of ANOVA produced significant effect on the Rate P and Rank P with minimized disagreement rates (p-value = 0.088 and 0.083 respectively).

The requirements are randomly generated as population, while the fitness value is calculated which gave rise to the best and mean fitnesses of the requirement generations that were subjected to a stoappge criteria during the iteration processes. The best fitness stopping criteria option was chosen during the simulation process.

Table 5 Prioritized results

Class	Point P	Rate P
1	17.347	4.604
2	7.652	4.230
3	52.831	4.258
4	85.639	3.714
5	24.396	4.370
6	39.844	4.172
7	19.435	1.276
8	30.188	4.167
9	27.635	4.410
10	26.000	2.620

The requirements generations significantly increased while the mean and best values were obtained for all the requirements which will aid the computation of final ranks for all the requirements. Also, the best, worst and mean scores of requirements were computed. In the context of requirement prioritization, the best scores can stand for the most valued requirements while the worst scores can stand for the requirements that were less ranked by the stakeholders. The mean scores are the scores used to determine the relative importance of requirements across all the stakeholders for the software development project. Therefore, mutation should be performed with respect to the weights of attributes describing each requirement set.

50 independent runs were conducted for each experiment. The intra cluster distances obtained by clustering algorithms on test dataset have been summarized in Table 5. The results contains the sum of weights and within-class variance of the requirements. The sum of weights are considered as the prioritized results. The iterations depcited in Table 6 represents the number of times that the clustering algorithm has calculated the fitness function to reach the (near) optimal solution. The fitness is the average correlation for all the lists of attribute weights. It is dependent on the number of iterations to reach the optimal solution. As seen from Table 6, the proposed algorithm has produced the highest quality solutions in terms of the determinant as well as the initial and final within class variances. Moreover, the standard deviation of solutions found by the proposed algorithm is small, which means that the proposed algorithm could find a near optimal solution in most of the runs. In other words, the results confirm that the proposed algorithm is viable and robust. In terms of the number of function evaluations, the k-means algorithm needs the least number of evaluations compared to other algorithms.

Table 6 Optimization summary

Repetition	Iteration	Initial Within-class variance	Final Within-class variance	Determinant (W)
1	6	1,106.869	23.889	26,745,715.142
2	6	1,136.975	16.750	23,724,606.023
3	6	1,121.779	16.807	23,820,814.376
4	4	1,117.525	24.996	27,447,123.233
5	4	1,125.507	17.803	25,404,643.926
6	4	1,125.913	19.185	27,601,524.028
7	4	1,128.978	18.691	21,068,536.793
8	4	1,117.436	18.454	25,247,590.810
9	3	1,108.735	521.172	195,940,759,324.757
10	3	1,118.845	20.662	26,631342.963
11	3	1,094.139	464.243	176,968,571,045.744
12	3	1,132.570	21.818	28,704,234.905
13	3	1,126.964	524.758	231,605,057,844.517
14	3	1,122.652	21.945	31,405,941.682
15	3	1,123.154	23.738	35,101,366.712
16	3	1,119.050	24.730	36,708,579.807
17	3	1,102.570	361.772	163,961,516,715.797
18	3	1,110.858	526.407	261,348,714,992.157
19	3	1,113.150	463.952	201,111,726,242.012
20	3	1,112.708	21.817	29,895,099.111
21	3	1,115.139	23.943	35,497,900.830
22	3	1,124.640	21.747	27,948,507.938
23	3	1,126.804	25.552	29,231,104.006
24	3	1,133.939	22.136	29,327,321.253
25	3	1,088.150	29.300	40,797,707.187
26	3	1,126.818	518.924	210,390,947,859.052
27	3	1,134.805	25.031	36,453,233.568
28	3	1,126.092	22.147	28,437,781.242
29	3	1,105.032	24.348	35,559,759.396
30	3	1,115.281	466.852	188,849,288,951.012
31	3	1,072.613	36.720	52,595,047.023
32	3	1,117.914	521.401	173,748,277,159.091
33	3	1,113.605	19.234	27,416,942.756
34	3	1,116.027	19.304	25,746,288.325
35	3	1,122.918	20.620	26,253,031.369
36	3	1,118.186	26.385	34,310,522.697
37	3	1,136.200	22.156	29,263,683.373
38	3	1,120.218	25.017	33,223,123.872
39	3	1,124.442	520.499	211,383,774,209.661

(continued)

Table 6 (continued)

Repetition	Iteration	Initial Within-class variance	Final Within-class variance	Determinant (W)
40	3	1,120.536	26.388	35,745,608.991
41	3	1,131.305	26.587	37,270,986.568
42	3	1,125.849	579.831	128,941,063,793.784
43	3	1,100.849	522.269	176,077,098,304.402
44	3	1,121.924	25.182	36,102,923.262
45	3	1,116.445	25.120	36,854,622.383
46	3	1,116.223	522.626	228,664,613,438.719
47	3	1,116.968	36.729	55,171,737.251
48	3	1,127.872	20.857	29,381,923.033
49	3	1,130.837	20.633	27,643,374.936
50	3	1,104.055	587.943	187,806,412,544.351

6 Conclusion

The requirements prioritization process can be considered as a multi-criteria decision making process. It is the act of pair-wisely selecting or ranking requirements based on pre-defined criteria. The aim of this research was to develop an improved prioritization technique based on the limitations of existing ones. The proposed algorithm resolved the issues of scalability, rank reversals and computational complexities. The method utilized in this research consisted of clustering/evolutionary based algorithms. The validation of the proposed approach was executed with relevant dataset while the performance was evaluated using statistical means. The results showed high correlation between the mean weights which finally yielded the prioritized results. The approach described in this research can help software engineers prioritize requirements capable of forecasting the expected behaviour of software under development. The results of the proposed system demonstrate two important properties of requirements prioritization problem; (i) Ability to cater for big data and (ii) ability to effectively update ranks and minimize disagreements between prioritized requirements. The proposed technique was also able to classify ranked requirements by computing the maximum, minimum and mean scores. This will help software engineers determined the most valued and least valued requirements which will aid in the planning for software releases in order to avoid breach of contracts, trusts or agreements. Based on the presented results, it will be appropriate to consider this research as an improvement in the field of computational intelligence. In summary, a hybrid method based on differential evolution and k-means algorithm was used in clustering requirements in order to determine their relative importance. It attempts to exploit the merits of two algorithms simultaneously, where the k-means was used in generating the initial solution and the differential evolution was utilized as an improvement algorithm.

Acknowledgment The authors appreciate the efforts of the Ministry of Science, Technology and Innovation Malaysia (MOSTI) under Vot 4S062 and Universiti Teknologi Malaysia (UTM) for supporting this research.

References

1. Ahl, V.: An experimental comparison of five prioritization methods-investigating ease of use, accuracy and scalability. Master's thesis, School of Engineering, Blekinge Institute of Technology, Sweden (2005)
2. Al-Sultan, K.S.: A Tabu search approach to the clustering problem. Pattern Recogn. **28**(9), 1443–1451 (1995)
3. Aritra, C., Bose, S., Das, S.: Automatic Clustering Based on Invasive Weed Optimization Algorithm: Swarm, Evolutionary, and Memetic Computing, pp. 105–112. Springer, Berlin (2011)
4. Babar, M., Ramzan, M., and Ghayyur, S.: Challenges and future trends in software requirements prioritization. In: Computer Networks and Information Technology (ICCNIT), pp. 319–324, IEEE (2011)
5. Berander, P., Svahnberg, M.: Evaluating two ways of calculating priorities in requirements hierarchies—An experiment on hierarchical cumulative voting. J. Syst. Softw. **82**(5), 836–850 (2009)
6. Chang, D., Xian, D., Chang, W.: A genetic algorithm with gene rearrangement for K-means clustering. Pattern Recogn. **42**, 1210–1222 (2009)
7. Ching-Yi, C., and Fun, Y.: Particle swarm optimization algorithm and its application to clustering analysis. In IEEE International Conference on Networking, Sensing and Control (2004)
8. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **38**(1), 218–237 (2008)
9. Das, S., Abraham, A., and Konar, A.: Automatic hard clustering using improved differential evolution algorithm. In: Studies in Computational Intelligence, pp. 137–174 (2009)
10. Demsar, J.: Statistical comparison of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
11. Greer, D., Ruhe, G.: Software release planning: an evolutionary and iterative approach. Inf. Softw. Technol. **46**(4), 243–253 (2004)
12. Hatamlou, A., Abdullah, S., and Hatamlou, M.: Data clustering using big bang-big crunch algorithm. In: Communications in Computer and Information Science, pp. 383–388 (2011)
13. Jain, A.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
14. Kao, Y., Zahara, E., Kao, I.: A hybridized approach to data clustering. Expert Syst. Appl. **34** (3), 1754–1762 (2008)
15. Karlsson, L., Thelin, T., Regnell, B., Berander, P., Wohlin, C.: Pair-wise comparisons versus planning game partitioning-experiments on requirements prioritization techniques. Empirical Softw. Eng. **12**(1), 3–33 (2006)
16. Karlsson, J., Olsson, S., Ryan, K.: Improved practical support for large scale requirements prioritizing. J. Requirements Eng. **2**, 51–67 (1997)
17. Karlsson, J., Wohlin, C., Regnell, B.: An evaluation of methods for prioritizing software requirements. Inf. Softw. Technol. **39**(14), 939–947 (1998)
18. Kassel, N.W., Malloy, B.A.: An approach to automate requirements elicitation and specification. In: Proceeding of the 7th IASTED International Conference on Software Engineering and Applications. Marina Del Rey, CA, USA (2003)
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (1995)

20. Kobayashi, M., Maekawa, M.: Need-based requirements change management. In: Proceeding of Eighth Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, pp. 171–178 (2001)
21. Krishna, K., Narasimha, M.: Genetic K-means algorithm. IEEE Trans. Syst. Man Cyber. Part B (Cyber.) **29**(3), 433–439 (1999)
22. Lim, S.L., Finkelstein, A.: takeRare: using social networks and collaborative filtering for large-scale requirements elicitation. Softw. Eng. IEEE Trans. **38**(3), 707–735 (2012)
23. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recogn. **33**(9), 1455–1465 (2000)
24. Moisiadis, F.: The fundamentals of prioritizing requirements. In: Proceedings of Systems Engineering Test and Evaluation Conference (SETE 2002) (2002)
25. Perini, A., Susi, A., Avesani, P.: A Machine Learning Approach to Software Requirements Prioritization. IEEE Trans. Software Eng. **39**(4), 445–460 (2013)
26. Qin, A.K., Suganthan, P.N.: Kernel neural gas algorithms with application to cluster analysis. In: Proceedings-International Conference on Pattern Recognition (2004)
27. Ramzan, M., Jaffar, A., Shahid, A.: Value based intelligent requirement prioritization (VIRP): expert driven fuzzy logic based prioritization technique. Int. J. Innovative Comput. **7**(3), 1017–1038 (2011)
28. Selim, S., Alsultan, K.: A simulated annealing algorithm for the clustering problem. Pattern Recogn. **24**(10), 1003–1008 (1991)
29. Sung, C.S., Jin, H.W.: A tabu-search-based heuristic for clustering. Pattern Recogn. **33**(5), 849–858 (2000)
30. Storn, R., Price, K.: Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, TR-95-012. Int. Comput. Sci. Inst., Berkeley (1995)
31. Tonella, P., Susi, A., Palma, F.: Interactive requirements prioritization using a genetic algorithm. Inf. Softw. Technol. **55**(2013), 173–187 (2012)
32. Thakurta, R.: A framework for prioritization of quality requirements for inclusion in a software project. Softw. Qual. J. **21**, 573–597 (2012)

One-Hour Ahead Electric Load Forecasting Using Neuro-fuzzy System in a Parallel Approach

Abderrezak Laouafi, Mourad Mordjaoui and Djalel Dib

Abstract Electric load forecasting is a real-life problem in industry. Electricity supplier's use forecasting models to predict the load demand of their customers to increase/decrease the power generated and to minimize the operating costs of producing electricity. This paper presents the development and the implementation of three new electricity demand-forecasting models using the adaptive neuro-fuzzy inference system (ANFIS) approach in parallel load series. The input-output data pairs used are the real-time quart-hourly metropolitan France electricity load obtained from the RTE website and forecasts are done for lead-time of a 1 h ahead. Results and forecasting performance obtained reveal the effectiveness of the third proposed approach and shows that 56 % of the forecasted loads have an APE (absolute percentage error) under 0.5, and an APE under one was achieved for about 80 % of cases. Which mean that it is possible to build a high accuracy model with less historical data using a combination of neural network and fuzzy logic.

1 Introduction

Forecasting electric load consumption is one of the most important areas in electrical engineering, due to its main role for the effectiveness and economical operation in power systems. It has become a major task for many researchers.

A. Laouafi (✉)
Department of Electrical Engineering, University of 20 August 1955—Skikda, Skikda, Algeria
e-mail: Laouafi_Abderrezak@yahoo.fr

M. Mordjaoui
Department of Electrical Engineering, LRPCSI Laboratory, University of 20 August 1955—Skikda, Skikda, Algeria
e-mail: Mordjaoui_Mourad@yahoo.fr

D. Dib
Department of Electrical Engineering, University of Tebessa, Tebessa, Algeria
e-mail: dibdjalel@gmail.com

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_5

The common approach is to analyse time series data of load consumption and temperature to modelling and to explain the series [30]. The intuition underlying time-series processes is that the future behavior of variables is related to its past values, both actual and predicted, with some adaptation/adjustment built-into take care of how past realizations deviated from those expected. The temporal forecasting can be broadly divided into 4 types:

- Very Short term (from few minutes to a 1 h).
- Short term (from 1 h to a week).
- Medium term (from a week to a year).
- Long term (from a year to several years).

Long term prediction is normally used for planning the growth of the generation capacity. This long term forecasting is used to decide whether to build new lines and sub-stations or to upgrade the existing systems. Medium-term load forecast is used to meet the load requirements at the height of the winter or the summer season and may require a load forecast to be made a few days to few weeks (or) few months in advance.

In STLF, the forecast calculates the estimated load for each hour of the day, the daily peak load and the daily/weekly energy generation. Many operations like real time generation control, security analysis, spinning reserve allocation, energy interchanges with other utilities, and energy transactions planning are done based on STLF.

Economic and reliable operation of an electric utility depends to a significant extent on the accuracy of the load forecast. The load dispatcher at main dispatch center must anticipate the load pattern well in advance so as to have sufficient generation to meet the customer requirements. Over estimation may cause the startup of too many generating units and lead to an unnecessary increase in the reserve and the operating costs. Underestimation of the load forecasts results in failure to provide the required spinning and standby reserve and stability to the system, which may lead into collapse of the power system network [1]. Load forecast errors can yield suboptimal unit commitment decisions. Hence, correct forecasting of the load is an essential element in power system.

In a deregulated, competitive power market, utilities tend to maintain their generation reserve close to the minimum required by an independent system operator. This creates a need for an accurate instantaneous-load forecast for the next several minutes. Accurate forecasts, referred to as very short-term load forecasts ease the problem of generation and load management to a great extent. These forecasts, integrated with the information about scheduled wheeling transactions, transmission availability, generation cost, spot market energy pricing, and spinning reserve requirements imposed by an independent system operator, are used to determine the best strategy for the utility resources. Very short-term load forecasting has become of much greater importance in today's deregulated power industry [5, 36].

A wide variety of techniques has been studied in the literature of short-term load forecasting [20]. For example, time series analysis (ARMA, ARIMA, ARMAX …etc.)

[4, 19], regression approach [34], exponential smoothing technique [44], artificial neural networks methods [35], hybrid approaches based on evolutionary algorithms [12] …etc.

The nature of electrical load forecasting problem is well suited to the technology of artificial neural networks (ANN) as they can model the complex non-linear relationships through a learning process involving historical data trends. Therefore, several studies in recent years have examined the application of ANN for short-term load forecasting [26].

Recently, hybrid neuro-fuzzy models have received a considerable attention from researchers in the field of short-term load forecasting [29, 30, 33]. Furthermore, the neuro-fuzzy approach attempts to exploit the merits of both neural-network and fuzzy-logic-based modeling techniques. For example, the fuzzy models are based on fuzzy IF-THEN rules and are, to a certain degree, transparent to interpretation and analysis, whereas the neural-networks based black-box model has a unique learning ability [32]. While building a FIS, the fuzzy sets, fuzzy operators, and the knowledge base are required to be specified. To implement an ANN for a specific application the architecture and learning algorithm are required. The drawbacks in these approaches appear complementary and consequently it is natural to consider implementing an integrated system combining the neuro-fuzzy concepts [41].

Nevertheless, very short-term load demand forecasting methods based on neuro-fuzzy approach are not so numerous [9, 10]. Therefore, this lack has motivated us to provide this paper to the development and the implementation of adaptive neuro-fuzzy inference system models devoted to VSTLF.

The paper is organized as follows. Section 2 is proposed to summarize very short-term load forecasting methods. Section 3 is devoted to the description of the ANFIS architecture. Section 4 describes the proposed estimation methods. Section 5 provides and explains forecasting results. Finally, Sect. 6 concludes the paper.

2 Overview of Very Short Term Load Forecasting Methods

Very short-term load forecasting (VSTLF) predicts the loads in electric power system 1 h into the future in steps of a few minutes in a moving window manner. Depending on the electric utilities, used data in VSTLF could be of, minute-by-minute basis [27, 43], 5-min intervals [11, 16, 17, 40], 15 min steps [6, 31], or a half-hourly intervals [24, 25].

Methods for very short-term load forecasting are limited. Existing methods include time series analysis, exponential smoothing, neural network (NN), fuzzy logic, adaptive Neuro-Fuzzy inference system, Kalman filtering, and Support Vector Regression. Usually, weather conditions in very short-term load forecasting are ignored because of the large time constant of load as a function of weather. The representative methods will be briefly reviewed in this Section.

2.1 Time Series Models

Time series models are based on the assumption that the data have an internal structure, such as autocorrelation, trend or seasonal variation. The forecasting methods detect and explore such a structure. A time series model includes:

- Autoregressive Model (AR)
- Moving Average Model (MA)
- Autoregressive Moving Average Model (ARMA)
- Autoregressive Integrated Moving Average model (ARIMA)
- Autoregressive Moving Average Model with exogenous inputs model (ARMAX)
- Autoregressive Integrated Moving Average with Explanatory Variable ARIMAX)

However, the most popular Time series models used in VSTLF are the autoregressive model [27], and the Autoregressive integrated moving average model [25].

2.2 Exponential Smoothing

The exponential smoothing approach is particularly convenient for short-time forecasting. Although it also employs weighting factors for past values, the weighting factors here decay exponentially with distance of the past values of the time series from the present time. This enables a compact formulation of the forecasting algorithm in which only a few most recent data are required and less calculation are needed. Principally, there are three exponential smoothing techniques, named simple, double and triple exponential smoothing technique. Simple exponential smoothing method is applied to short-term forecasting for time series without trend and seasonality. Double exponential smoothing is used in time series that contains a trend. For seasonal time series, the third technique, which known as Holt-winters method is useful because it can capture both trend and seasonality. For VSTLF, Holt winters technique is the mostly used [25, 31, 37, 43].

2.3 Neural Network

Neural networks (NN) assume a functional relationship between load and affecting factors, and estimate the functional coefficients by using historical data. There are many types of neural networks including the multilayer perceptron network (MLP), self-organizing network and Hopfield's recurrent network [27]. Based on learning strategies, neural network methods for load forecasting can be classified into two groups. The first one is a supervised neural network that adjusts its weights according to the error between pre-tested and desired output. The second are methods based on unsupervised learning algorithm. Generally, methods based on supervised learning algorithm like a feed forward multilayer perceptron are used.

Although MLP is a classical model, it is still the most favorite ANN architecture in forecasting applications. The structure of MLP consists of input layer, hidden layer, and output nodes connected in a feed-forward fashion via multiplicative weights. Inputs are multiplied by connection weights and passed on to the neurons in hidden layer nodes. The neurons in hidden and output layer nodes have a transfer function. The inputs to hidden layer are passed through a transfer function to produce output. ANN would learn from experience and is trained with back-propagation and supervised learning algorithm. The proper selection of training data improves the efficiency of ANN [8].

Most neural network methods for VSTLF use inputs e.g., time index, load of previous hour, load of the yesterday and previous week with same hour and weekday index to the target hour [5, 15, 39]. Chen and York [7] have presented a neural network based very short-term load prediction. Results indicated that under normal situations, forecasted minutely load values by NN-based VSTLP for the future 15 min are provided with good accuracy on the whole as well as for the worst cases.

2.4 Fuzzy Logic

Fuzzy logic is a generalization of Boolean logic; it can identify and approximate any unknown nonlinear dynamic systems on the compact set to arbitrary accuracy. However, model based on fuzzy logic are robust in forecasting because there are no need to mathematical formulation between system inputs and outputs. A defuzzification process is used to produce the desired output after processing logic inputs. A fuzzy logic system was implemented in the paper of Liu et al. [27] by drawing similarities in load trend (e.g., between weekdays and weekdays) from a huge of data. A pattern database generated via effective training was then used to predict the load change. The preliminary study shows that it is feasible to design a simple, satisfactory dynamic forecaster to predict the very short-term load trends on-line using fuzzy logic. The performances of FL-based forecaster are much superior to the one of AR-based forecaster.

2.5 Adaptive Neuro-fuzzy Inference System (ANFIS)

An adaptive Neuro-Fuzzy inference system is a combination of an artificial neural network and a fuzzy inference system. It is a fuzzy Takagi-Sugeno model put in the framework of adaptive systems to facilitate learning and adaptation [21].

An artificial neural network is designed to mimic the characteristics of the human brain and consists of a collection of artificial neurons. An adaptive network is a multi-layer feed-forward network in which each node (neuron) performs a particular function on incoming signals. The form of the node functions may vary from node to node. In an adaptive network, there are two types of nodes: adaptive and fixed. The function and the grouping of the neurons are dependent on the overall function of the network. However, the ANFIS network is composed of five layers. Each layer contains some nodes described by the node function. A few layers have the same number of nodes, and nodes in the same layer have similar functions.

de Andrade and da Silva [10] have presented the use of ANFIS for very short-term load demand forecasting, with the aim to regulate the demand and supply of electrical energy in order to minimize the fluctuations and to avoid undesirable disturbances in power systems operations. Used time series measured, in 5 min intervals, were collected from substations located in Cordeirópolis and Ubatuba, cities located in the countryside and seaside of São Paulo state, respectively. Authors denoted that a higher number of epochs didn't present better performance of ANFIS. The experimental results demonstrate that ANFIS is a good tool for forecasting one-step forward for very short-term load demand.

2.6 *Kalman Filtering*

The Kalman filtering (KF) algorithm is a robust tracking algorithm that has long been applied to many engineering fields such as radar tracking. In load forecasting, it is introduced to estimate the optimal load forecast parameters and overcome the unknown disturbance in the linear part of the systems during load prediction [48].

Very short-term load prediction in [45] was done using slow and fast Kalman estimators and an hourly forecaster. The Kalman model parameters are determined by matching the frequency response of the estimator to the load residuals. The methodology was applied to load data taken from the portion of the western North American power system operated by the BPA.

Guan et al. [18] have presented a method of wavelet neural networks trained by hybrid Kalman filters to produce very short-term forecasting with prediction interval estimates online. Testing results demonstrate the effectiveness of hybrid Kalman filters for capturing different features of load components, and the accuracy of the overall variance estimate derived based on a data set from ISO New England.

2.7 *Support Vector Regression*

Support vector machines (SVM) method, which was proposed by Vapnik [46], is used to solve the pattern recognition problems by determining a hyperplane that separates positive and negative examples, by optimization of the separation margin between them [32]. Later Vapnik promotes the SVM method to deal with the function fitting problems in 1998, which forms the support vector regression (SVR) method [47]. SVR produces a decision boundary that can be expressed in terms of a few support vectors and can be used with kernel functions to create complex nonlinear decision boundaries. Similarly to linear regression, SVR tries to find a function that best fits the training data.

Setiawan et al. [38] have presented a new approach for the very short-term electricity load demand forecasting using SVR. Support vector regression was applied to predict the load demand every 5 min based on historical data from the Australian electricity operator NEMMCO for 2006–2008. The results showed that SVR is a very promising prediction model, outperforming the back propagation neural networks (BPNN) prediction algorithms, which is widely used by both industry forecasters and researchers.

3 Adaptive Neuro-fuzzy Inference System

The hybrid neuro-fuzzy approach is a way to create a fuzzy model from data by some kind of learning method that is motivated by learning algorithms used in neural networks. This considerably reduces development time and cost while improving the accuracy of the resulting fuzzy model. Thus, neuro-fuzzy systems are basically adaptive fuzzy systems developed by exploiting the similarities between fuzzy systems and certain forms of neural networks, which fall in the class of generalized local methods. Therefore, the performance of a neuro-fuzzy system can also be represented by a set of humanly understandable rules or by a combination of localized basis functions associated with local models, making them an ideal framework to perform nonlinear predictive modeling. However, there are some ways to mix neural networks and fuzzy logic. Consequently, three main categories characterize these technologies: fuzzy neural networks, neural fuzzy systems and fuzzy-neural hybrid systems [2, 3]. In the last approach, both neural networks and fuzzy logic are used independently, becoming, in this sense, a hybrid system.

An adaptive Neuro-Fuzzy inference system is a cross between an artificial neural network and a fuzzy inference system. An artificial neural network is designed to mimic the characteristics of the human brain and consists of a collection of artificial neurons. Adaptive Neuro-Fuzzy Inference System (ANFIS) is one of the most successful schemes which combine the benefits of these two powerful paradigms into a single capsule [21]. An ANFIS works by applying neural learning rules to identify and tune the parameters and structure of a Fuzzy Inference System (FIS). There are several features of the ANFIS which enable it to achieve great success in a wide range of scientific applications. The attractive features of an ANFIS include: easy to implement, fast and accurate learning, strong generalization abilities, excellent explanation facilities through fuzzy rules, and easy to incorporate both linguistic and numeric knowledge for problem solving [22]. According to the neuro-fuzzy approach, a neural network is proposed to implement the fuzzy system, so that structure and parameter identification of the fuzzy rule base are accomplished by defining, adapting and optimizing the topology and the parameters of the corresponding neuro-fuzzy network, based only on the available data. The network can be regarded both as an adaptive fuzzy inference system with the capability of learning fuzzy rules from data, and as a connectionist architecture provided with linguistic meaning [2, 3].

3.1 Architecture of ANFIS

An adaptive Neuro-Fuzzy inference system implements a Takagi–Sugeno FIS, and uses a multilayer network that consists of five layers in which each node (neuron) performs a particular function on incoming signals. The form of the node functions may vary from node to node. In an adaptive network, there are two types of nodes: adaptive and fixed. The function and the grouping of the neurons are dependent on the overall function of the network.

A hybrid-learning algorithm proposed by Jang trains generally the ANFIS system [21]. This algorithm uses back-propagation learning to determine the parameters related to membership functions and least mean square estimation to determine the consequent parameters [41]. The role of training algorithm is tuning all the modifiable parameters to make the ANFIS output match the training data [30]. For representation, Fig. 1 shows an ANFIS with two inputs x_1 and x_2 and one output y, each variable has two fuzzy sets A_1, A_2, B_1 and B_2, circle indicates a fixed node, whereas a square indicates an adaptive node. Then a first order Takagi-Sugeno-type fuzzy *if-then* rule (Fig. 2) could be set up as

$$Rule\ 1 : if\ x_1\ is\ A_1\ and\ x_2\ and\ B_1, Then\ f_1 = f_1(x_1, x_2) = a_1x_1 + b_1x_2 + c_1 \tag{1}$$

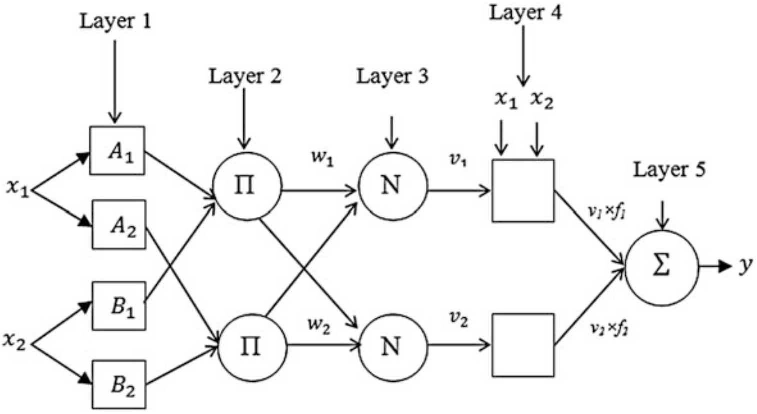


Fig. 1 ANFIS architecture

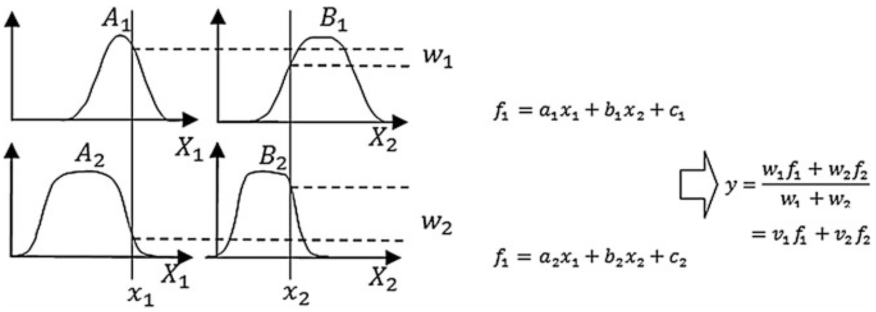

A_1
B_1
w_1
X_1
X_2
$f_1 = a_1x_1 + b_1x_2 + c_1$
A_2
B_2
w_2
$y = \frac{w_1f_1 + w_2f_2}{w_1 + w_2}$
$= v_1f_1 + v_2f_2$
X_1
X_2
x_1
x_2
$f_1 = a_2x_1 + b_2x_2 + c_2$

Fig. 2 A two input first order Sugeno fuzzy model with two rules

$$Rule\,2: if\; x_1\; is\, A_2\; and\, x_2\; is\, B_2,\; Then\, f_2 = f_2(x_1, x_2) = a_2x_1 + b_2x_2 + c_2 \tag{2}$$

f_i are the outputs within the fuzzy region specified by the Fuzzy rule, $\{a_i, b_i, c_i\}$ are the design parameters that are determined during the training process. Some layers of ANFIS have the same number of nodes, and nodes in the same layer have similar functions: ***Layer 1***: Every node i in this layer is an adaptive node. The outputs of layer 1 are the fuzzy membership grade of the inputs, which are given by:

$$O^1_{i,1} = \mu_{A_i}(x_1), i = 1, 2 \tag{3}$$

$$O^1_{i,2} = \mu_{B_i}(x_2), i = 1, 2 \tag{4}$$

It other words, $O^1_{i,1}$ is the membership function of A_i, and it specifies the degree to which the given input satisfies the quantifier A_i. $\mu_{A_i}(x_1)$ and $\mu_{B_i}(x_2)$ can adopt any fuzzy membership function. However, the most commonly used are Bell shaped and Gaussian membership functions. For example, if the bell shaped membership function is employed, $\mu_{A_i}(x_1)$ is given by:

$$\mu_{A_i}(x_1) = \frac{1}{1 + \left(\left(\frac{x_1 - c_i}{a_i}\right)^2\right)^{b_i}} \tag{5}$$

Where a_i, b_i and c_i are the parameters of the membership function, governing the bell shaped functions accordingly. ***Layer 2***: Every node in this layer is a circle node labeled Π, which multiplies the incoming signals and sends the product out. The Fuzzy operators are applied in this layer to compute the rule antecedent part [30]. The output of nodes in this layer can be presented as:

$$w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2) \quad i = 1, 2 \tag{6}$$

Layer 3: The fuzzy rule base is normalized in the third hidden layer. Every node in this layer is a circle node labeled N. The *i*th node calculates the ratio of the *i*th rule's firing strength to the sum of all rules' firing strengths:

$$v_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{7}$$

Layer 4: Every node i in this layer is a square node with a node function:

$$O_i^4 = v_i \times f_i = v_i(a_i x_1 + b_i x_2 + c_i) \quad i = 1, 2 \tag{8}$$

Layer 5: Finally, layer five, consisting of circle node labeled with $\sum$ is the summation of all incoming signals. Hence, the overall output of the model is given by:

$$O_i^5 = \sum_{i=1}^{2} v_i \times f_i = \frac{\sum_{i=1}^{2} w_i \times f_i}{\sum_{i=1}^{2} w_i} \tag{9}$$

3.2 Learning Algorithm of ANFIS

The hybrid-learning algorithm of ANFIS proposed by Jang et al. [23] is a combination of Steepest Descent and Least Squares Estimate Learning algorithm. Let the total set of parameters be S and let S1 denote the premise parameters and S2 denote the consequent parameters. The premise parameters are known as nonlinear parameters and the consequent parameters are known as linear parameters. The ANFIS uses a two pass learning algorithm: forward pass and backward pass. In forward pass the premise parameters are not modified and the consequent parameters are computed using the Least Squares Estimate Learning algorithm [28].

In backward pass, the consequent parameters are not modified and the premise parameters are computed using the gradient descent algorithm. Based on these two learning algorithms, ANFIS adapts the parameters in the adaptive network. The task of training algorithm for this architecture is tuning all the modifiable parameters to make the ANFIS output match the training data. Note here that *ai*, *bi* and *ci* describe the sigma, slope and the center of the bell MF's, respectively. If these parameters are fixed, the output of the network becomes:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{10}$$

Substituting Eq. (7) into Eq. (10) yields:

$$f = v_1 \times f_1 + v_2 \times f_2 \tag{11}$$

Substituting the fuzzy if-then rules into Eq. (11), it becomes:

$$f = v_1(a_1x_1 + b_1x_2 + c_1) + v_2(a_2x_1 + b_2x_2 + c_2) \tag{12}$$

After rearrangement, the output can be expressed as:

$$f = (v_1x_1) \cdot a_1 + (v_1x_2) \cdot b_1 + (v_1) \cdot c_1 + (v_2x_1) \cdot a_2 + (v_2x_2) \cdot b_2 + (v_2) \cdot c_2 \tag{13}$$

This is a linear combination of the modifiable parameters. For this observation, we can divide the parameter set S into two sets:
$S = S_1 \oplus S_2$
S = set of total parameters,
S_1 = set of premise (nonlinear) parameters,
S_2 = set of consequent (linear) parameters
$\oplus$: Direct sum

For the forward path (see Fig. 2), we can apply least square method to identify the consequent parameters. Now for a given set of values of S_1, we can plug training data and obtain a matrix equation:

$$A\Theta = y \tag{14}$$

where Θ contains the unknown parameters in S_2. This is a linear square problem, and the solution for Θ, which is minimizes $\|A\Theta = y\|$, is the least square estimator:

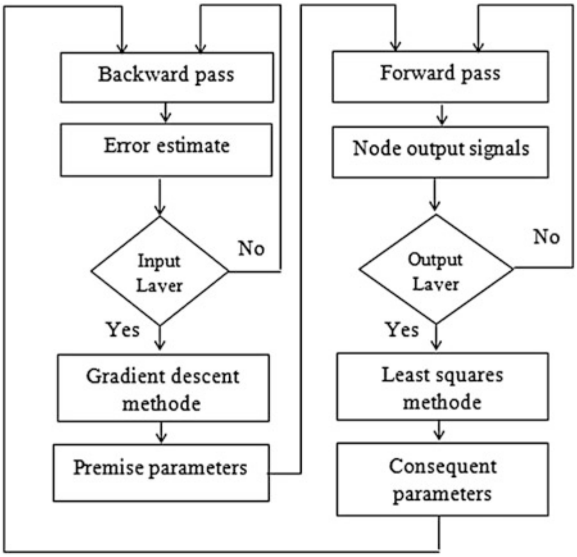
$$\Theta^* = \left(A^TA\right)^{-1}A^Ty \tag{15}$$

we can use also recursive least square estimator in case of on-line training. For the backward path (see Fig. 2), the error signals propagate backward. The premise parameters are updated by descent method, through minimising the overall quadratic cost function:

$$J(\Theta) = \frac{1}{2}\sum_{N=1}^{N}\left[y(k) - \hat{y}(k,\Theta)\right]^2 \tag{16}$$

In a recursive manner with respect $\Theta_{(S2)}$. The update of the parameters in the ith node in layer L^{th} layer can be written as:

Fig. 3 ANFIS training algorithm for adjusting production rules parameters



$$\hat{\Theta}_i(k) = \hat{\Theta}_i^L(k-1) + \eta \frac{\partial^+ E(k)}{\partial \hat{\Theta}_i^L(k)} \tag{17}$$

where η is the learning rate and the gradient vector.

$$\frac{\partial^+ E}{\partial \hat{\Theta}_i^L} = \varepsilon_{L,i} \frac{\partial \hat{z}_{L,i}}{\partial \hat{\Theta}_i^L} \tag{18}$$

$\partial \hat{z}_{L,i}$ being the node's output and $\varepsilon_{L,i}$ is the backpropagated error signal.

Figure 3 presents the ANFIS activities in each pass. As discussed earlier, the consequent parameters thus identified are optimal under the condition that the premise parameters are fixed.

The flow chart of training methodology of ANFIS system is shown in Fig. 4. Usually, the modeling process starts by obtaining a data set (input-output data pairs) and dividing it into training and checking data sets. Training data constitutes a pairs of input and output vectors. In order to make data suitable for the training stage, this data are normalized and used as the input and the outputs to train the ANFIS. Once both training and checking data were presented to ANFIS, the FIS was selected to have parameters associated with the minimum checking data model error. The stopping criterion of ANFIS is the testing error when it became less than the tolerance limit defining at the beginning of the training stage or by putting constraint on the number of learning iterations.

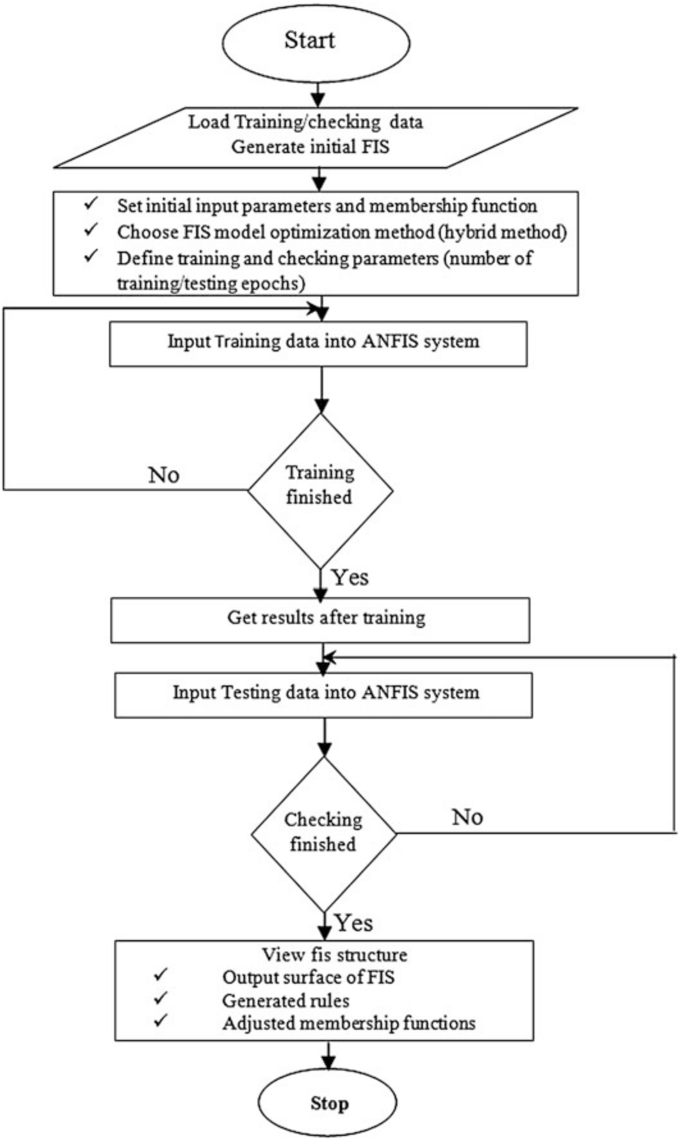

Start
Load Training/checking data
Generate initial FIS
✓ Set initial input parameters and membership function
✓ Choose FIS model optimization method (hybrid method)
✓ Define training and checking parameters (number of training/testing epochs)
Input Training data into ANFIS system
Training finished
No
Yes
Get results after training
Input Testing data into ANFIS system
Checking finished
No
Yes
View fis structure
✓ Output surface of FIS
✓ Generated rules
✓ Adjusted membership functions
Stop

Fig. 4 Flow chart of training methodology of ANFIS system [2, 3]

4 Proposed Methods

4.1 Load Data Treatment

Variations in electrical load are, among other things, time of the day dependent, introducing a dilemma for the forecaster: whether to partition the data and use a separate model for each specified time of the day (the parallel approach), or use a single model (the sequential approach) [14].

In this work, the electrical load time series are separated in autonomous points. A set of independent points means that the load at each quarter-hour of the day is independent from the load at any other quarter-hour. These are called parallel series. We propose three suggestions. The first is to split the French quart-hourly load data into 96 parallel series, each series is composed by loads consumed at a specified time of a distinctive day (Saturday, Sunday …, etc.). In the second, the parallel series contain loads from all previous days consumed at a specified quarter-hour. In the third, parallel load series are classified in three categories: Saturdays, Sundays, workdays.

Data classification need some knowledge such as the identification of the first day in the historical load data (Saturday, Sunday,…), the number of days in each month, the number of days available in the historical load data. By effecting simple If-Then statements, the parallel load series for each class can be extracted.

4.2 ANFIS Architecture

The proposed ANFIS model can be represented by seven steps:

Step 1: We select the day and the hour in which we would like to predict the load. Hence, the four series of this hour, noted y(i), y(i + 1), y(i + 2) and y(i + 3), represents the output of the ANFIS.

Step 2: As inputs, we creates seven parallel load series noted y(i − 1), y(i − 2), y(i − 3), y(i − 4), y(i − 5), y(i − 6) and y(i − 7), and an input index x(i). Hence, the inputs load series records loads consumed at previous nearest quarter-hours.

Step 3: Then, we remove the last load value from inputs and outputs series, where the last value of y(i), y(i + 1), y(i + 2) and y(i + 3) represents the load to be forecasted. These new series are noted y(i)’, y(i + 1)’, y(i + 2)’, y(i + 3)’, y(i − 1)’, y(i − 2)’, y(i − 3)’, y(i − 4)’, y(i − 5)’, y(i − 6)’, y(i − 7)’ and x(i)’.

Step 4: We performs now an exhaustive search within the available inputs to select only one input vector that most influence in y(i)’. The exhaustive search builds an ANFIS model, trains it for twenty epochs, and reports the performance achieved. Selected model should provide the minimum RMSE in the outputs predicting.

Step 5: Selected input from the previous step is used then to generate and trains a Sugeno FIS of two fuzzy rules, two sigmoid membership and twenty epochs.

Step 6: At last, original input related to the selected input is used to predict the load in y(i).

Step 7: We repeat then the two last previous steps in order to predict the desired load in y(i + 1), y(i + 2) and y(i + 3).

However, we propose in the paper, three ANFIS models:

- ***Method 1***: the electrical loads series in this method are obtained by implementing the first classification.
- ***Method 2***: the electrical loads series in this method are obtained by implementing the second classification.
- ***Method 3***: the electrical loads series in this method are obtained by implementing the third classification.

5 Results and Discussion

All three methods are applied in the French real time load data. These data consists of quart-hourly recording ranging from Sunday 07 April 2013 until Friday 28 February 2014, where the last month is used in a one-hour ahead forecasting. Used data are represented by Fig. 5.

The graphical user interface developed for all three methods is represented by Fig. 6. The essential function of this tool is to ensure, at any quarter-hour selected

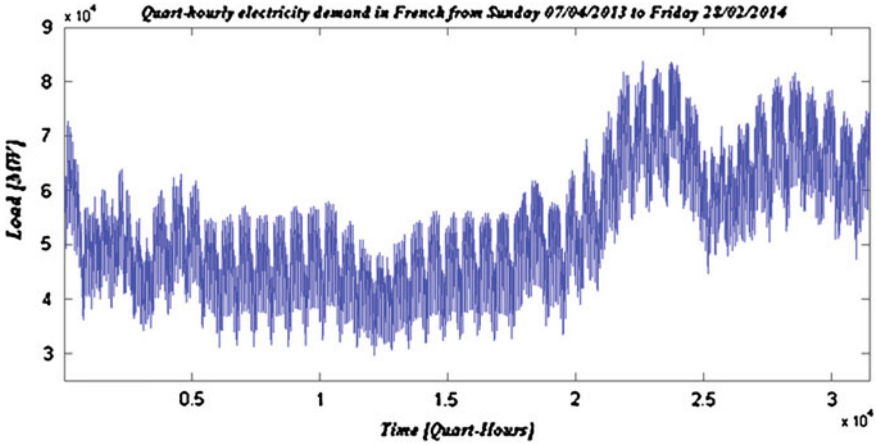
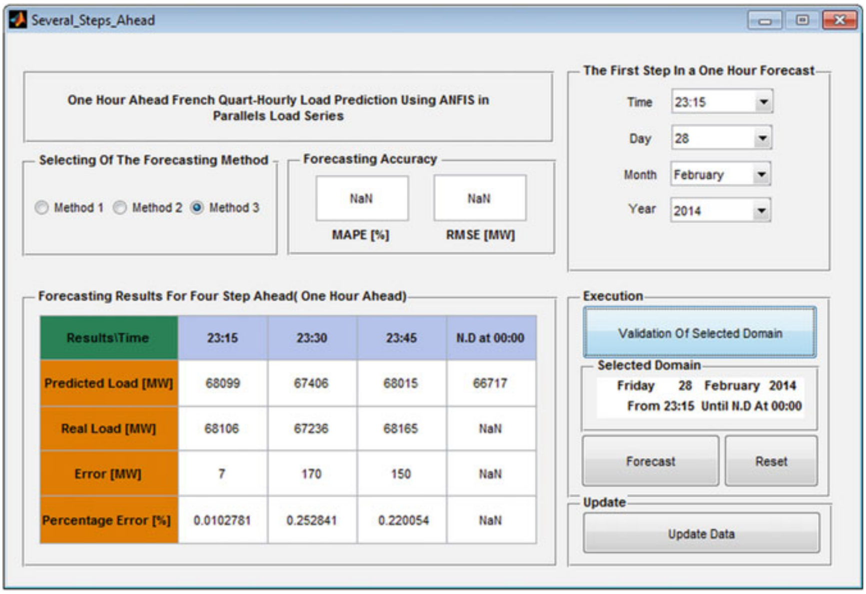


Fig. 5 Quart-hourly French electric load time series from Sunday 07 April 2013 to 28 February 2014


Several_Steps_Ahead

One Hour Ahead French Quart-Hourly Load Prediction Using ANFIS in Parallels Load Series

Selecting Of The Forecasting Method

○ Method 1 ○ Method 2 ● Method 3

Forecasting Accuracy

MAPE [%]: NaN

RMSE [MW]: NaN

The First Step In a One Hour Forecast

Time: 23:15

Day: 28

Month: February

Year: 2014

Forecasting Results For Four Step Ahead(One Hour Ahead)

Results\Time	23:15	23:30	23:45	N.D at 00:00
Predicted Load [MW]	68099	67406	68015	66717
Real Load [MW]	68106	67236	68165	NaN
Error [MW]	7	170	150	NaN
Percentage Error [%]	0.0102781	0.252841	0.220054	NaN

Selected Domain

Friday 28 February 2014

From 23:15 Until N.D At 00:00

Fig. 6 Developed forecasting tool for a 1-h ahead electric load forecasting using ANFIS

from the user, a 1-h ahead demand prediction. In addition, when actual values of the load are available, the performance of forecasted loads could be verified using different error measurement criteria. Moreover, the tool has the advantage, by division the original data into 96 separated load series, to reduce the number of data should be taken into consideration before predicting the load at the specified hour and, and by the way reducing computational time.

To evaluate and compare the performance of the new proposed methods, forecasts are done along the month of February 2014. For each day in the selected Month, first steps in the 1-h ahead prediction are 00:15, 01:15, 02:15… until 23:15. Forecasts by Method 3 in the field “11:15 p.m. to 00:00” are based on the first classification.

Results of three methods are represented in Figs. 7, 8 and 9. As shown in these figures, the proposed ANFIS models have successfully predict the load over the month of February 2014, and there is almost no different between predicted and real load.

To evaluate the performance of developed models, we have used APE (Absolute Percentage error), MAPE (Mean absolute percentage error) and RMSE (Root mean square error) criteria. Evaluation results are summarized in Table 1.

$$APE = \frac{|\hat{y}_t - y_t|}{y_t} \times 100 \tag{19}$$

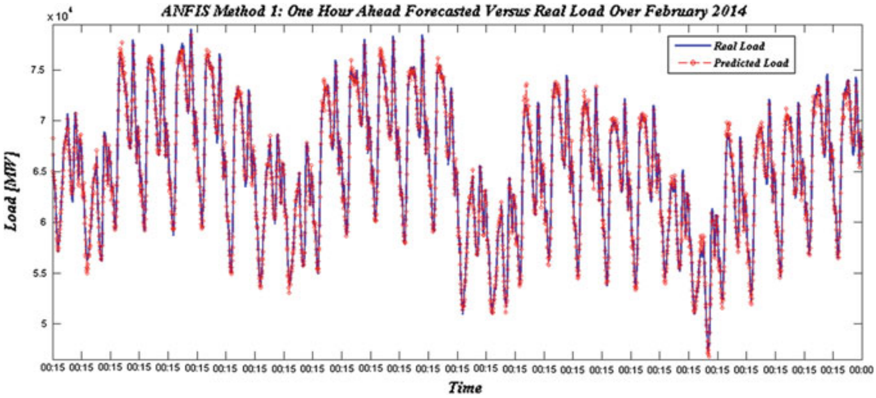


Fig. 7 One-hour ahead forecasted load versus real load for method 1

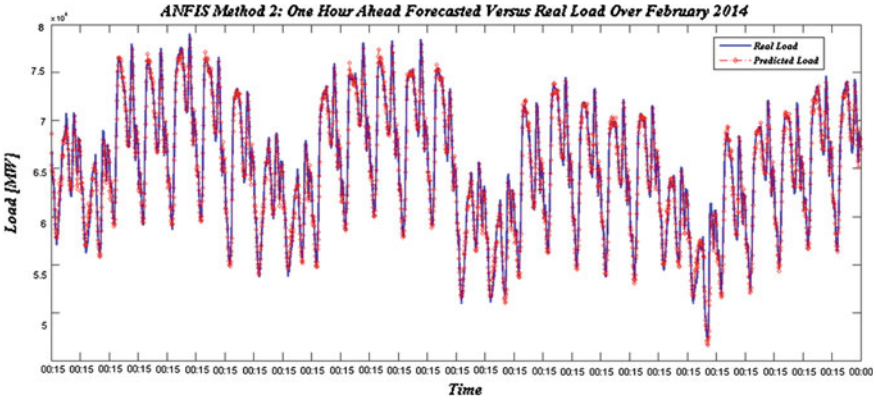


Fig. 8 One-hour ahead forecasted load versus real load for method 2

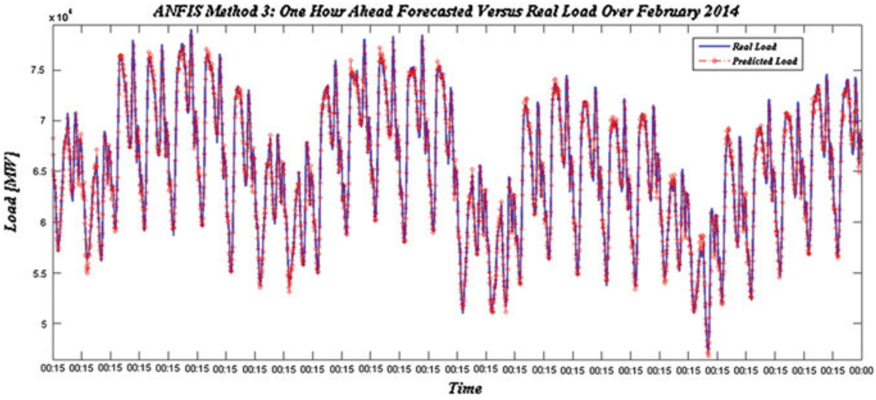


Fig. 9 One-hour ahead forecasted load versus real load for method 3

$$MAPE = \frac{1}{n}\sum_{m=1}^{n}\frac{|\hat{y}_t - y_t|}{y_t} \times 100 \tag{20}$$

$$RMSE = \frac{1}{n}\sqrt{\sum_{m=1}^{n}(\hat{y}_t - y_t)^2} \tag{21}$$

As shown in Table 1, all three methods achieve a high accuracy in the 1-h ahead load forecasting. We can perceive that's the accuracy of the second method decrease in free days compared to in working days, this can be justified by the fact that this model use loads from a specific quarter-hour in all previous days in the historical load data. Here, we should note that the load in Saturday and Sunday is very low compared to in working days. For example, if we would like to predict the load at a specified quarter-hour in a Saturday using the second method, than latest values of the parallel load series contains loads from previous nearest Monday until the previous day (a Friday). These values effects the prediction because they are height compared to the desired load in Saturday. This can be clearly observed in the first method, which is based on intraday classification and the parallel series contains load consumed at a specified quarter-hour in a typical day (Monday, Tuesday …), where the accuracy of prediction in Saturdays and Sundays is not different compared to in others days.

However, what is impressive; is that the number of data that should be taken into account in the second method is seven times higher than the used in the first method, while the obtained results clearly show that the first method is more accurate than the second method. This demonstrates that, in addition to the selection of an appropriate forecasting technique, classifying the historical data to extract useful knowledge and patterns from large database also, affect on the forecasting accuracy. Moreover, by classifying the data in the third method into three clusters: Saturdays, Sundays and working days, the accuracy is increased, and it is superior to that in the first and the second method.

Figure 10 represents the distribution of the maximum percentage error for three methods. We can perceive that the proposed ANFIS methods have failed to predict peaks consummation around 19:00 with a high accuracy, which make a real need in this field, to propose a separate model for predicting the peak consumption, or to train the ANFIS with more than one input. However, as shown if Fig. 11, for the third proposed method, 56 % of the forecasted loads have an APE under 0.5 and an APE under one was achieved for about 80 % of cases. Likewise, as demonstrate Figs. 12 and 13, the first and the second method provide also a good accuracy in most of time.

In addition to a robust model that assures a very high accuracy, time required in the forecasting procedure take an important role in real time electric load forecasting. Tables 2 show in detail, for all three methods, prediction results for four different hours. Results are obtained using Windows 7 64 bit and MATLAB R2013a in a

Table 1 One-hour ahead forecasting accuracy for a three proposed methods over the month of February 2014

	Method 1						Method 2					
	MAPE			RMSE			MAPE			RMSE		
Sat 01 Feb 2014	0.6117	0.7193	**0.7432**	614.3959	769.0680	**794.52**	1.0218	0.7877	**0.8404**	878.28	817.96	**862.22**
Sun 02 Feb 2014	0.9875			860.3865			1.2399			1102.7		
Mon 03 Feb 2014	0.7229			798.3312			0.6262			717.08		
Tue 04 Feb 2014	0.6374			761.9828			0.6081			681.86		
Wed 05 Feb 2014	0.7088			770.9580			0.6640			774.39		
Thu 06 Feb 2014	0.6188			704.2705			0.5958			699.53		
Fri 07 Feb 2014	0.7481			845.0963			0.7583			792.49		
Sat 08 Feb 2014	0.5808	0.6460		601.0464	733.7181		0.8091	0.7372		656.40	779.67	
Sun 09 Feb 2014	0.7268			783.3347			1.1295			1052.8		
Mon 10 Feb 2014	0.7171			779.3326			0.7075			769.91		
Tue 11 Feb 2014	0.7159			798.9643			0.6553			721.39		
Wed 12 Feb 2014	0.6148			687.2521			0.6556			720.37		
Thu 13 Feb 2014	0.6003			802.7999			0.6210			831.08		
Fri 14 Feb 2014	0.5661			656.5204			0.5823			627.08		
Sat 15 Feb 2014	0.6898	0.7493		715.4786	833.9910		0.9259	0.8790		783.74	891.84	
Sun 16 Feb 2014	0.7259			747.6218			1.4201			1291.7		
Mon 17 Feb 2014	0.9867			1076.7			0.9973			997.71		
Tue 18 Feb 2014	0.8307			977.6372			0.7036			795.45		
Wed 19 Feb 2014	0.6776			758.8888			0.6995			750.08		
Thu 20 Feb 2014	0.6761			748.8020			0.7456			765.08		
Fri 21 Feb 2014	0.6580			739.0911			0.6607			713.34		
Sat 22 Feb 2014	0.7332	0.8583		701.0428	836.5079		0.9180	0.9576		821.48	949.38	
Sun 23 Feb 2014	1.0665			949.4138			1.7065			1465.7		
Mon 24 Feb 2014	0.9972			991.8403			1.1547			1037.8		
Tue 25 Feb 2014	0.8315			787.2473			0.7563			710.35		
Wed 26 Feb 2014	0.7654			726.2326			0.7103			793.33		
Thu 27 Feb 2014	0.7744			790.2337			0.7671			819.81		
Fri 28 Feb 2014	0.8396			866.0148			0.6902			776.58		

(continued)

Table 1 (continued)

	Method 3					
	MAPE			RMSE		
Sat 01 Feb 2014	0.6117	0.6846	**0.6966**	614.39	733.23	**748.10**
Sun 02 Feb 2014	0.9875			860.38		
Mon 03 Feb 2014	0.5412			690.52		
Tue 04 Feb 2014	0.6077			695.24		
Wed 05 Feb 2014	0.7280			804.35		
Thu 06 Feb 2014	0.6035			694.97		
Fri 07 Feb 2014	0.7123			745.31		
Sat 08 Feb 2014	0.5808	0.6157		601.04	708.35	
Sun 09 Feb 2014	0.7268			783.33		
Mon 10 Feb 2014	0.6004			690.21		
Tue 11 Feb 2014	0.6115			721.01		
Wed 12 Feb 2014	0.6277			711.93		
Thu 13 Feb 2014	0.5954			811.52		
Fri 14 Feb 2014	0.5672			612.99		
Sat 15 Feb 2014	0.6898	0.6879		715.47	757.18	
Sun 16 Feb 2014	0.7259			747.62		
Mon 17 Feb 2014	0.8994			1010.3		
Tue 18 Feb 2014	0.6518			767.93		
Wed 19 Feb 2014	0.5989			685.11		
Thu 20 Feb 2014	0.6183			664.92		
Fri 21 Feb 2014	0.6313			648.38		
Sat 22 Feb 2014	0.7332	0.7982		701.04	791.16	
Sun 23 Feb 2014	1.0665			949.41		
Mon 24 Feb 2014	0.8746			872.60		
Tue 25 Feb 2014	0.6531			619.95		
Wed 26 Feb 2014	0.7518			758.67		
Thu 27 Feb 2014	0.7600			814.72		
Fri 28 Feb 2014	0.7484			776.89		

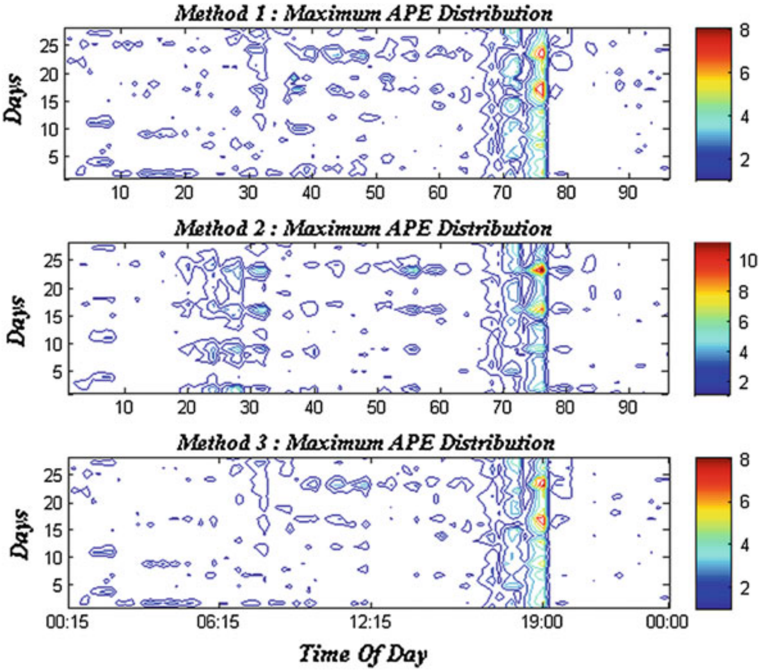
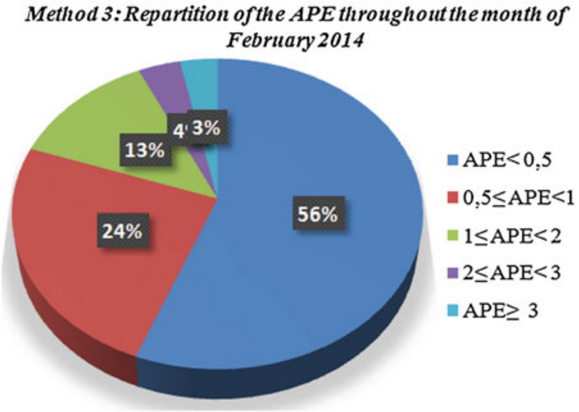


Fig. 10 Maximum APE distribution for all three methods

Fig. 11 Repartition of the APE throughout the month of February 2014 for the third method



laptop of 4 GB of RAM, Intel i3 380 M processor and 5,400-rpm hard drive. Results confirm the superior accuracy of the third proposed method. In addition, needed time in the forecasting procedure is less than two second. This time includes the exhaustive search affected to select the more appropriate input for training the ANFIS, and four ANFIS corresponding to each quarter-hour load series.

Fig. 12 Repartition of the APE throughout the month of February 2014 for the first Method

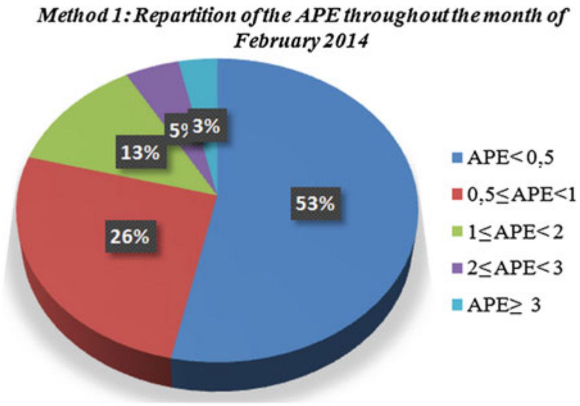
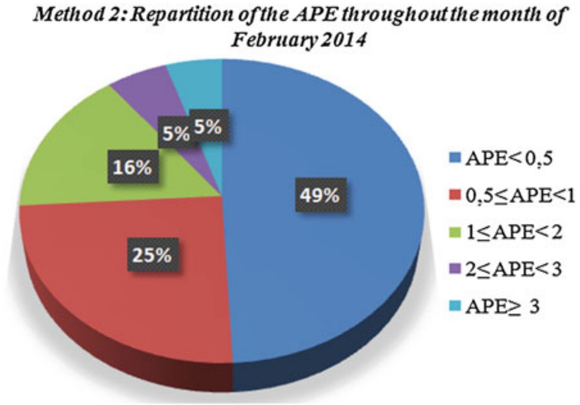


Fig. 13 Repartition of the APE throughout the month of February 2014 for the second method



Moreover, the accuracy decreases when the proposed methods are used to forecast the peak consummation at 19:00. For example, the MAPE pass from 0.23 % at the beginning hour of 1 February 2014 to 1.38 % at the field 18:15–19:00 of the last day of February 2014. These is more clearly perceived from Figs. 8, 9, 10 and 12 where maximums APE (between 3 and 8 % in first and third method, and between 3 and 10 % in the second method) are done around 18:15–19:00. This decrease can be justified by the non-consideration of weather condition in the proposed methods. As we know, changing weather conditions represent the major source of variation in peak load forecasting and the inclusion of temperature has a significant effect due to the fact that in winter heating systems are used specially in the evening around 19:00, whilst in summer air conditioning appliances are used particularly around 13:00. Other weather factors include relative humidity, wind speed and nebulosity. Therefore, numerous papers are devoted to electricity peak demand forecasting [13, 42]. However, since weather variables tend to change in a smooth fashion, Weather conditions are ignored in very short term load forecasting

Table 2 Forecasting results for four different hours in all three methods

		Method 1						Method 2				
Results Time		Real load (MW)	Predicted load (MW)	APE (%)	MAPE (%)	RMSE (MW)	Elapsed time (s)	Predicted load (MW)	APE (%)	MAPE (%)	RMSE (MW)	Elapsed Time (s)
01 Feb	00:15	68,270	68,264	0.0087	0.2322	205.12	1.2480	68,597	0.4789	0.3141	262.74	1.4670
	00:30	66,626	66,653	0.0405				66,593	0.0495			
	00:45	65,407	65,074	0.5091				65,343	0.0978			
	00:00	64,252	64,490	0.3704				64,657	0.6303			
10 Feb	06:15	62,739	62,532	0.3299	0.7238	508.82	1.0920	62,062	1.0790	1.9104	1292.83	1.4510
	06:30	64,621	63,912	1.097				63,127	2.3119			
	06:45	65,548	65,953	0.6178				64,219	2.0275			
	07:00	67,153	67,724	0.8502				65,660	2.2232			
19 Feb	12:15	70,632	70,782	0.2123	1.5312	1435.65	1.1380	71,275	0.9103	1.0796	767.08	1.3730
	12:30	70,245	70,124	0.1722				70,964	1.0235			
	12:45	70,960	69,106	2.6127				71,706	1.0513			
	12:00	69,827	72,011	3.1277				70,758	1.3333			
28 Feb	18:15	68,734	68,795	0.0887	1.5536	1521.45	1.1390	68,720	0.0203	1.5818	1497.83	1.3420
	18:30	70,121	69,783	0.4820				69,656	0.6631			
	18:30	72,242	70,709	2.122				70,501	2.4099			
	19:00	74,000	71,394	3.5216				71,607	3.2337			

(continued)

Table 2 (continued)

		Method 3					
Results		Real load (MW)	Predicted load (MW)	APE (%)	MAPE (%)	RMSE (MW)	Elapsed time (s)
Time							
01 Feb	00:15	68,270	68,264	0.0087	0.2322	205.12	1.2480
	00:30	66,626	66,653	0.0405			
	00:45	65,407	65,074	0.5091			
	00:00	64,252	64,490	0.3704			
10 Feb	06:15	62,739	62,404	0.5339	0.57799	457.483	1.4660
	06:30	64,621	63,816	1.2457			
	06:45	65,548	65,285	0.4012			
	07:00	67,153	67,065	0.1310			
19 Feb	12:15	70,632	70,987	0.5026	0.6617	487.69	1.2950
	12:30	70,245	70,640	0.5623			
	12:45	70,960	71,354	0.5552			
	12:00	69,827	70,544	1.0268			
28 Feb	18:15	68,734	68,793	0.0858	1.3793	1366.3	1.2640
	18:30	70,121	69,872	0.3551			
	18:45	72,242	70,866	1.9047			
	19:00	74,000	71,653	3.1716			

and they could be captured in the demand series itself. By the way, it would be more appropriate for us to propose a separate model for predicting the peak consumption.

6 Conclusion

In this paper, three new models based on the use of adaptive neuro-fuzzy inference system technique in parallel data were developed to forecast the French real time quart-hourly load, in a 1-h ahead basis. The best ANFIS technique found was the third, which classify the parallel load series in three categories. We have perceive that the proposed ANFIS methods have some failed to predict peaks consummation around 19:00; which make a real need in this field, to propose a separate model for predicting the peak consumption, or to train the ANFIS with more than one input. However, for the third method, 56 % of the forecasted loads have an APE under 0.5, and an APE under one was achieved for about 80 % of cases. Therefore, at exception for peak consummation, the third proposed method can be successfully applied to build a 1-h ahead electric load prediction in real time.

References

1. Amit, J., Srinivas, E., Rasmimayee, R.: Short term load forecasting using fuzzy adaptive inference and similarity. World Congress on Nature and Biologically Inspired Computing, pp. 1743–1748. NaBIC, Coimbatore India (2009)
2. Azar, A.T.: Adaptive neuro-fuzzy systems. In: Azar, A.T (ed.), Fuzzy Systems. InTech, Vienna, Austria, (2010a) ISBN 978-953-7619-92-3
3. Azar, A.T.: Fuzzy Systems. IN-TECH, Vienna, Austria (2010). ISBN 978-953-7619-92-3
4. Box, G.E.P., Jenkins, J.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1976)
5. Charytoniuk, W., Chen, M.S.: Very short-term load forecasting using artificial *neural networks*. IEEE Trans. Power Syst. **15**(1), 263–268 (2000)
6. Cheah, P.H., Gooi, H.B., Soo, F.L.: Quarter-hour-ahead load forecasting for microgrid energy management system. In: IEEE Trondheim PowerTech, Trondheim, 19–23 June 2011, pp. 1–6 (2011)
7. Chen, D., York, M.: Neural network based very short term load prediction, In: IEEE PES Transmission & Distribution Conference and Exposition, Chicago IL, 21–24 April 2008, pp. 1–9 (2008)
8. Daneshi, H., Daneshi, A.: Real time load forecast in power system. In: Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, DRPT2008, Nanjuing China, 6–9 April 2008 pp. 689–695 (2008)
9. de Andrade, L.C.M., da Silva, I.N.: Very short-term load forecasting based on ARIMA model and intelligent systems. In: 15th International Conference on Intelligent System Applications to Power Systems, ISAP ‘09, 8-12 Nov, Curitiba, pp. 1–6 (2009)
10. de Andrade, L.C.M., da Silva, I.N.: Very short-term load forecasting using a hybrid neuro-fuzzy approach. In: Eleventh Brazilian Symposium on Neural Networks (SBRN), 23–28 Oct, Sao Paulo, pp. 115–120 (2010a)

11. de Andrade, L.C.M., da Silva, I.N.: Using intelligent system approach for very short-term load forecasting purposes. In: IEEE International Energy Conference, 18–22 Dec. 2010, Manama, pp. 694–699 (2010b)
12. El-Telbany, M.: Short-term forecasting of Jordanian electricity demand using particle swarm optimization. Electr. Power Syst. Res. **78**(3), 425–433 (2008)
13. Fan, S., Mao, C., Chen, L.: Peak load forecasting using the self-organizing map. In: Advances in Neural Networks—ISNN 2005, Springer Berlin Heidelberg. New York, Part III, pp. 640–647 (2005)
14. Fay, D., Ringwood, J.V., Condon, M., Kellyc, M. 24-h electrical load data-a sequential or partitioned time series? *Neuro-computing*, vol 55(3–4), October 2003, pp. 469–498 (2003)
15. Guan, C., Luh, P.B., Coolbeth, M.A., Zhao, Y., Michel, L.D., Chen, Y., Manville, C. J., Friedland, P.B., Rourke, S.J.: Very short-term load forecasting: multilevel wavelet neural networks with data pre-filtering. In Proceeding of: Power and Energy Society General Meeting, 2009. Calgary, pp. 1–8 (2009)
16. Guan, C., Luh, P.B., Michel, L.D., Coolbeth, M.A., Friedland, P.B.: Hybrid Kalman algorithms for very short-term load forecasting and confidence interval estimation. In: IEEE Power and Energy Society General Meeting, 25–29 July 2010, Minneapolis, MN, pp 1–8 (2010)
17. Guan, C., Luh, P.B., Michel, L.D., Wang, Y., Friedland, P.B.: Very short-term load forecasting: wavelet neural networks with data pre-filtering. IEEE Trans. Power Syst. **28**(1), 30–41 (2013)
18. Guan, C., Luh, P.B., Michel, L.D., Chi, Z.: Hybrid Kalman filters for very short-term load forecasting and prediction interval estimation. IEEE Trans. Power Syst. **28**(4), 3806–3817 (2013)
19. Hagan, M.T., Behr, S.M.: The time series approach to short term load forecasting. IEEE Trans. Power Syst. **2**(3), 785–791 (1987)
20. Hesham, K.: Electric load forecasting: Literature survey and classification of methods. Int. J. Syst. Sci. **33**(1), 23–34 (2002)
21. Jang, J.S.R.: ANFIS: Adaptive network based fuzzy inference system. IEEE Trans. Syst., Man, Cybern. **23**(3), 665–685 (1993)
22. Jang, J.-S.R., Sun, C.-T.: Neuro-fuzzy modeling and control. Proc. IEEE **83**(3), 378–406 (1995)
23. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, pp. 353–360. Prentice-Hall, Englewood Cliffs, NJ (1997)
24. Khan, G.M., Zafari, F., Mahmud, S.A.: Very short term load forecasting using cartesian genetic programming evolved recurrent neural networks (CGPRNN). In: 2013 12th International Conference on Machine Learning and Applications (ICMLA), 4–7 Dec. 2013, Miami, FL, USA, vol 2, pp. 152–155 (2013)
25. Kotillová, A.: Very Short-Term Load Forecasting Using Exponential Smoothing and ARIMA Models. J. Inf., Control Manage. Syst. **9**(2), 85–92 (2011)
26. Kumar, M.: Short-term load forecasting using artificial neural network techniques. Thesis for Master of Science degree in Electrical Engineering. India, Rourkela, National Institute of Technology (2009)
27. Liu, K., Subbarayan, S., Shoults, R.R., Manry, M.T., Kwan, C., Lewis, F.L., Naccarino, J.: Comparison of very short-term load forecasting techniques. IEEE Trans. Power Syst. **11**(2), 877–882 (1996)
28. Mordjaoui, M., Chabane, M., Boudjema, B., Daira, R.: Qualitative ferromagnetic hysteresis Modeling. J. Comput. Sci. **3**(6), 399–405 (2007)
29. Mordjaoui, M., Boudjema, B., Bouabaz, M., Daira, R.: Short-term electric load forecasting using neuro-fuzzy modeling for nonlinear system identification. In: Proceeding of the 3rd Conference on Nonlinear Science and Complexity, Jul. 28–31, Ankara, Turkey, link: (2010) http://nsc10.cankaya.edu.tr/proceedings/, paper ID_64

30. Mordjaoui, M., Boudjema, B.: Forecasting and modeling electricity demand using anfis predictor. J. Math. Statis. **7**(4), 275–281 (2011)
31. Neusser, L., Canha, L.N., Abaide, A., Finger, M.: Very short-term load forecast for demand side management in absence of historical data. In: International Conference on Renewable Energies and Power Energies and Power Quality (ICREPQ'12), 28–30th March, Santiago de Compostela (Spain), (2012) link: http://www.icrepq.com/icrepq%2712/479-neusser.pdf
32. Palit A.K., Popovic D.: Computational intelligence in time series forecasting: theory and engineering applications. In: Series: Advance in Industrial Control Springer, New York, Inc. Secaucus, NJ, USA (2005)
33. Palit, A.K., Anheier, W., Popovic, D.: Electrical load forecasting using a neural-fuzzy approach. In: Natural Intelligence for Scheduling, Planning and Packing Problems, Studies in Computational Intelligence, Springer, Berlin Heidelberg, volume 250, pp. 145–173 (2009)
34. Papalexopoulos, A.D., Hesterberg, T.C.: A regression-based approach to short-term system load forecasting. IEEE Trans. Power Syst. **5**(4), 1535–1547 (1990)
35. 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 Trans. Power Syst. **6**(2), 442–449 (1991)
36. Qingle, P., Min, Z.: Very short-term load forecasting based on neural network and rough set. In: 2010 International Conference on Intelligent Computation Technology and Automation, 11-12 May 2010, Changsha, volume 3, pp. 1132–1135
37. Ramiro, S.: Very short-term load forecasting using exponential smoothing. In: Engineering Universe for Scientific Research and Management, volume 3(5) (2011) Link: http://www.eusrm.com/PublishedPaper/3Vol/Issue5/20112011eusrm03051037.pdf
38. Setiawan, A., Koprinska, I., Agelidis, V.G.: Very short-term electricity load demand forecasting using support vector regression. In: Proceedings of International Joint Conference on Neural Networks, 14–19 June 2009, Atlanta, Georgia, pp. 2888–289 (2009)
39. Shamsollahi, P., Cheung, K.W., Quan, C., Germain, E.H.: A neural network based very short term load forecaster for the interim ISO New England electricity market system. In: 22nd IEEE Power Engineering Society International Conference on PICA. Innovative Computing for Power—Electric Energy Meets the Market, 20–24 May 2001, Sydney, NSW, pp. 217–222 (2001)
40. Shankar, R., Chatterjee, K., Chatterjee, T.K.: A very short-term load forecasting using Kalman filter for load frequency control with economic load dispatch. J. Eng. Sci. Technol., Rev. **5**(1), 97–103 (2012)
41. Sumathi, S., Surekha, P.: Computational intelligence paradigms theory and applications using MATLAB. Taylor and Francis Group, LLC (2010)
42. Sigauke, C., Chikobvu, D.: Daily peak electricity load forecasting in South Africa using a multivariate non-parametric regression approach. ORiON: J. ORSSA, 26(2), pp. 97–111 (2010)
43. Taylor, J.W.: An evaluation of methods for very short-term load forecasting using minute-by-minute British data. Int. J. Forecast. **24**(4), 645–658 (2008)
44. Taylor, J.W.: Short-term load forecasting with exponentially weighted methods. IEEE Trans. Power Syst. **27**, 458–464 (2012)
45. Trudnowski, D.J., McReynolds, W.L., Johnson, J.M.: Real-time very short-term load prediction for power-system automatic generation control. IEE Trans. Control Syst. Technol. **9** (2), 254–260 (2001)
46. Vapnik, V.: The nature of statistic learning theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
47. Vapnik, V.: Statistical learning theory. Wiley, Inc., New York (1998)
48. Zhao, F., Su, H.: Short-term load forecasting using Kalman filter and Elman neural network. In: 2nd IEEE Conference on Industrial Electronics and Applications, 23–25 May 2007, Harbin, pp. 1043–1047 (2007)

A Computational Intelligence Optimization Algorithm Based on the Behavior of the Social-Spider

Erik Cuevas, Miguel Cienfuegos, Raul Rojas and Alfredo Padilla

Abstract Classical optimization methods often face great difficulties while dealing with several engineering applications. Under such conditions, the use of computational intelligence approaches has been recently extended to address challenging real-world optimization problems. On the other hand, the interesting and exotic collective behavior of social insects have fascinated and attracted researchers for many years. The collaborative swarming behavior observed in these groups provides survival advantages, where insect aggregations of relatively simple and “unintelligent” individuals can accomplish very complex tasks using only limited local information and simple rules of behavior. Swarm intelligence, as a computational intelligence paradigm, models the collective behavior in swarms of insects or animals. Several algorithms arising from such models have been proposed to solve a wide range of complex optimization problems. In this chapter, a novel swarm algorithm called the Social Spider Optimization (SSO) is proposed for solving optimization tasks. The SSO algorithm is based on the simulation of cooperative behavior of social-spiders. In the proposed algorithm, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. The algorithm considers two different search agents (spiders): males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are typically found in the colony. In order to illustrate the proficiency and robustness of the proposed approach, it is compared to other well-known evolutionary methods. The comparison examines several standard benchmark functions that are commonly considered within the literature of evolutionary

---

E. Cuevas (✉) · M. Cienfuegos
Departamento de Electrónica, CUCEI, Universidad de Guadalajara,
Guadalajara, Mexico
e-mail: erik.cuevas@cucei.udg.mx

R. Rojas
Institut Für Informatik, Freie Universität Berlin, Berlin, Germany

A. Padilla
Instituto Tecnológico de Celaya, Celaya, Mexico

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_6

algorithms. The outcome shows a high performance of the proposed method for searching a global optimum with several benchmark functions.

Keywords Swarm algorithms · Global optimization · Bio-inspired algorithms · Computational intelligence · Evolutionary algorithms · Metaheuristics

1 Introduction

Computational intelligence has emerged as powerful tools for information processing, decision making and knowledge management. The techniques of computational intelligence have been successfully developed in areas such as neural networks, fuzzy systems and evolutionary algorithms. It is predictable that in the near future computational intelligence will play a more important role in tackling several engineering problems.

The collective intelligent behavior of insect or animal groups in nature such as flocks of birds, colonies of ants, schools of fish, swarms of bees and termites have attracted the attention of researchers. The aggregative conduct of insects or animals is known as swarm behavior. Entomologists have studied this collective phenomenon to model biological swarms while engineers have applied these models as a framework for solving complex real-world problems. This branch of artificial intelligence which deals with the collective behavior of swarms through complex interaction of individuals with no supervision is frequently addressed as swarm intelligence. Bonabeau defined swarm intelligence as "any attempt to design algorithms or distributed problem solving devices inspired by the collective behavior of the social insect colonies and other animal societies" [5]. Swarm intelligence has some advantages such as scalability, fault tolerance, adaptation, speed, modularity, autonomy and parallelism [19].

The key components of swarm intelligence are self-organization and labor division. In a self-organizing system, each of the covered units responds to local stimuli individually and may act together to accomplish a global task, via a labor separation which avoids a centralized supervision. The entire system can thus efficiently adapt to internal and external changes.

Several swarm algorithms have been developed by a combination of deterministic rules and randomness, mimicking the behavior of insect or animal groups in nature. Such methods include the social behavior of bird flocking and fish schooling such as the Particle Swarm Optimization (PSO) algorithm [20], the cooperative behavior of bee colonies such as the Artificial Bee Colony (ABC) technique [17], the social foraging behavior of bacteria such as the Bacterial Foraging Optimization Algorithm (BFOA) [27], the simulation of the herding behavior of krill individuals such as the Krill Herd (KH) method [13], the mating behavior of firefly insects such as the Firefly (FF) method [41] and the emulation of the lifestyle of cuckoo birds such as the Cuckoo Optimization Algorithm (COA) [28].

In particular, insect colonies and animal groups provide a rich set of metaphors for designing swarm optimization algorithms. Such cooperative entities are complex systems that are composed by individuals with different cooperative-tasks where each member tends to reproduce specialized behaviors depending on its gender [4]. However, most of swarm algorithms model individuals as unisex entities that perform virtually the same behavior. Under such circumstances, algorithms waste the possibility of adding new and selective operators as a result of considering individuals with different characteristics such as sex, task-responsibility, etc. These operators could incorporate computational mechanisms to improve several important algorithm characteristics including population diversity and searching capacities.

Although PSO and ABC are the most popular swarm algorithms for solving complex optimization problems, they present serious flaws such as premature convergence and difficulty to overcome local minima [35, 36]. The cause for such problems is associated to the operators that modify individual positions. In such algorithms, during their evolution, the position of each agent for the next iteration is updated yielding an attraction towards the position of the best particle seen so-far (in case of PSO) or towards other randomly chosen individuals (in case of ABC). As the algorithm evolves, those behaviors cause that the entire population concentrates around the best particle or diverges without control. It does favors the premature convergence or damage the exploration-exploitation balance [3, 37].

The interesting and exotic collective behavior of social insects have fascinated and attracted researchers for many years. The collaborative swarming behavior observed in these groups provides survival advantages, where insect aggregations of relatively simple and “unintelligent” individuals can accomplish very complex tasks using only limited local information and simple rules of behavior [11]. Social-spiders are a representative example of social insects [22]. A social-spider is a spider species whose members maintain a set of complex cooperative behaviors [32]. Whereas most spiders are solitary and even aggressive toward other members of their own species, social-spiders show a tendency to live in groups, forming long-lasting aggregations often referred to as colonies [1]. In a social-spider colony, each member, depending on its gender, executes a variety of tasks such as predation, mating, web design, and social interaction [1, 6]. The web it is an important part of the colony because it is not only used as a common environment for all members, but also as a communication channel among them [23]. Therefore, important information (such as trapped prays or mating possibilities) is transmitted by small vibrations through the web. Such information, considered as a local knowledge, is employed by each member to conduct its own cooperative behavior, influencing simultaneously the social regulation of the colony.

In this paper, a novel swarm algorithm, called the Social Spider Optimization (SSO) is proposed for solving optimization tasks. The SSO algorithm is based on the simulation of the cooperative behavior of social-spiders. In the proposed algorithm, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. The algorithm considers two different search agents (spiders): males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are typical in a colony. Different to most of existent swarm algorithms, in the proposed approach, each individual is modeled considering two genders. Such fact allows not only to emulate in a better realistic way the cooperative behavior of the colony, but also to incorporate computational mechanisms to avoid critical flaws commonly present in the popular PSO and ABC algorithms, such as the premature convergence and the incorrect exploration-exploitation balance. In order to illustrate the proficiency and robustness of the proposed approach, it is compared to other well-known evolutionary methods. The comparison examines several standard benchmark functions which are commonly considered in the literature. The results show a high performance of the proposed method for searching a global optimum in several benchmark functions.

This paper is organized as follows. In Sect. 2, we introduce basic biological aspects of the algorithm. In Sect. 3, the novel SSO algorithm and its characteristics are both described. Section 4 presents the experimental results and the comparative study. Finally, in Sect. 5, conclusions are drawn.

2 Biological Fundamentals

Social insect societies are complex cooperative systems that self-organize within a set of constraints. Cooperative groups are better at manipulating and exploiting their environment, defending resources and brood, and allowing task specialization among group members [15, 25]. A social insect colony functions as an integrated unit that not only possesses the ability to operate at a distributed manner, but also to undertake enormous construction of global projects [14]. It is important to acknowledge that global order in social insects can arise as a result of internal interactions among members.

A few species of spiders have been documented exhibiting a degree of social behavior [22]. The behavior of spiders can be generalized into two basic forms: solitary spiders and social spiders [1]. This classification is made based on the level of cooperative behavior that they exhibit [6]. In one side, solitary spiders create and maintain their own web while live in scarce contact to other individuals of the same species. In contrast, social spiders form colonies that remain together over a communal web with close spatial relationship to other group members [23].

A social spider colony is composed of two fundamental components: its members and the communal web. Members are divided into two different categories: males and females. An interesting characteristic of social-spiders is the highly female-biased population. Some studies suggest that the number of male spiders barely reaches the 30 % of the total colony members [1, 2]. In the colony, each member, depending on its gender, cooperate in different activities such as building and maintaining the communal web, prey capturing, mating and social contact (Yip 2008). Interactions among members are either direct or indirect [29]. Direct interactions imply body contact or the exchange of fluids such as mating. For indirect interactions, the communal web is used as a “medium of communication” which conveys important information that is available to each colony member [23]. This information encoded as small vibrations is a critical aspect for the collective coordination among members (Yip 2008). Vibrations are employed by the colony members to decode several messages such as the size of the trapped preys, characteristics of the neighboring members, etc. The intensity of such vibrations depend on the weight and distance of the spiders that have produced them.

In spite of the complexity, all the cooperative global patterns in the colony level are generated as a result of internal interactions among colony members [12]. Such internal interactions involve a set of simple behavioral rules followed by each spider in the colony. Behavioral rules are divided into two different classes: social interaction (cooperative behavior) and mating [30].

As a social insect, spiders perform cooperative interaction with other colony members. The way in which this behavior takes place depends on the spider gender. Female spiders which show a major tendency to socialize present an attraction or dislike over others, irrespectively of gender [1]. For a particular female spider, such attraction or dislike is commonly developed over other spiders according to their vibrations which are emitted over the communal web and represent strong colony members (Yip 2008). Since the vibrations depend on the weight and distance of the members which provoke them, stronger vibrations are produced either by big spiders or neighboring members [23]. The bigger a spider is, the better it is considered as a colony member. The final decision of attraction or dislike over a determined member is taken according to an internal state which is influenced by several factors such as reproduction cycle, curiosity and other random phenomena (Yip 2008).

Different to female spiders, the behavior of male members is reproductive-oriented [26]. Male spiders recognize themselves as a subgroup of alpha males which dominate the colony resources. Therefore, the male population is divided into two classes: dominant and non-dominant male spiders [26]. Dominant male spiders have better fitness characteristics (normally size) in comparison to non-dominant. In a typical behavior, dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate upon the center of the male population as a strategy to take advantage of the resources wasted by dominant males [33].

Mating is an important operation that no only assures the colony survival, but also allows the information exchange among members. Mating in a social-spider colony is performed by dominant males and female members [16]. Under such circumstances, when a dominant male spider locates one or more female members within a specific range, it mates with all the females in order to produce offspring [8].

3 The Social Spider Optimization (SSO) Algorithm

In this paper, the operational principles from the social-spider colony have been used as guidelines for developing a new swarm optimization algorithm. The SSO assumes that entire search space is a communal web, where all the social-spiders interact to each other. In the proposed approach, each solution within the search space represents a spider position in the communal web. Every spider receives a weight according to the fitness value of the solution that is symbolized by the social-spider. The algorithm models two different search agents (spiders): males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are commonly assumed within the colony.

An interesting characteristic of social-spiders is the highly female-biased populations. In order to emulate this fact, the algorithm starts by defining the number of female and male spiders that will be characterized as individuals in the search space. The number of females N_f is randomly selected within the range of 65–90 % of the entire population N. Therefore, N_f is calculated by the following equation:

$$N_f = \text{floor}[(0.9 - \text{rand} \cdot 0.25) \cdot N], \tag{1}$$

where rand is a random number between [0,1] whereas floor$(\cdot)$ maps a real number to an integer number. The number of male spiders N_m is computed as the complement between N and N_f. It is calculated as follows:

$$N_m = N - N_f \tag{2}$$

Therefore, the complete population **S**, composed by N elements, is divided in two sub-groups **F** and **M**. The Group **F** assembles the set of female individuals ($\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}\}$) whereas **M** groups the male members ($\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_m}\}$), where $\mathbf{S} = \mathbf{F} \cup \mathbf{M} (\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\})$, such that $\mathbf{S} = \{\mathbf{s}_1 = \mathbf{f}_1, \mathbf{s}_2 = \mathbf{f}_2, \ldots, \mathbf{s}_{N_f} = \mathbf{f}_{N_f}, \mathbf{s}_{N_f+1} = \mathbf{m}_1, \mathbf{s}_{N_f+2} = \mathbf{m}_2, \ldots, \mathbf{s}_N = \mathbf{m}_{N_m}\}$.

3.1 Fitness Assignation

In the biological metaphor, the spider size is the characteristic that evaluates the individual capacity to perform better over its assigned tasks. In the proposed approach, every individual (spider) receives a weight w_i which represents the solution quality that corresponds to the spider i (irrespective of gender) of the population **S**. In order to calculate the weight of every spider the next equation is used:

$$w_i = \frac{J(\mathbf{s}_i) - worst_{\mathbf{S}}}{best_{\mathbf{S}} - worst_{\mathbf{S}}}, \tag{3}$$

where $J(\mathbf{s}_i)$ is the fitness value obtained by the evaluation of the spider position $\mathbf{s}_i$ with regard to the objective function $J(\cdot)$. The values $worst_{\mathbf{S}}$ and $best_{\mathbf{S}}$ are defined as follows (considering a maximization problem):

$$best_{\mathbf{S}} = \max_{k\in\{1,2,\ldots,N\}} (J(\mathbf{s}_k)) \text{ and } worst_{\mathbf{S}} = \min_{k\in\{1,2,\ldots,N\}} (J(\mathbf{s}_k)) \tag{4}$$

3.2 Modelling of the Vibrations Through the Communal Web

The communal web is used as a mechanism to transmit information among the colony members. This information is encoded as small vibrations that are critical for the collective coordination of all individuals in the population. The vibrations depend on the weight and distance of the spider which has generated them. Since the distance is relative to the individual that provokes the vibrations and the member who detects them, members located near to the individual that provokes the vibrations, perceive stronger vibrations in comparison with members located in distant positions. In order to reproduce this process, the vibrations perceived by the individual i as a result of the information transmitted by the member j are modeled according to the following equation:

$$Vib_{i,j} = w_j \cdot e^{-d_{i,j}^2}, \tag{5}$$

where the $d_{i,j}$ is the Euclidian distance between the spiders i and j, such that $d_{i,j} = \left\|\mathbf{s}_i - \mathbf{s}_j\right\|$.

Although it is virtually possible to compute perceived-vibrations by considering any pair of individuals, three special relationships are considered within the SSO approach:

1. Vibrations $Vibc_i$ are perceived by the individual i $(\mathbf{s}_i)$ as a result of the information transmitted by the member c $(\mathbf{s}_c)$ who is an individual that has two important characteristics: it is the nearest member to i and possesses a higher weight in comparison to i $(w_c > w_i)$.

$$Vibc_i = w_c \cdot e^{-d_{i,c}^2} \tag{6}$$

2. The vibrations $Vibb_i$ perceived by the individual i as a result of the information transmitted by the member b $(\mathbf{s}_b)$, with b being the individual holding the best weight (best fitness value) of the entire population $\mathbf{S}$, such that $w_b = \max_{k\in\{1,2,\ldots,N\}} (w_k)$.

$$Vibb_i = w_b \cdot e^{-d_{i,b}^2} \tag{7}$$

3. The vibrations $Vibf_i$ perceived by the individual i $(\mathbf{s}_i)$ as a result of the information transmitted by the member f $(\mathbf{s}_f)$, with f being the nearest female individual to i.

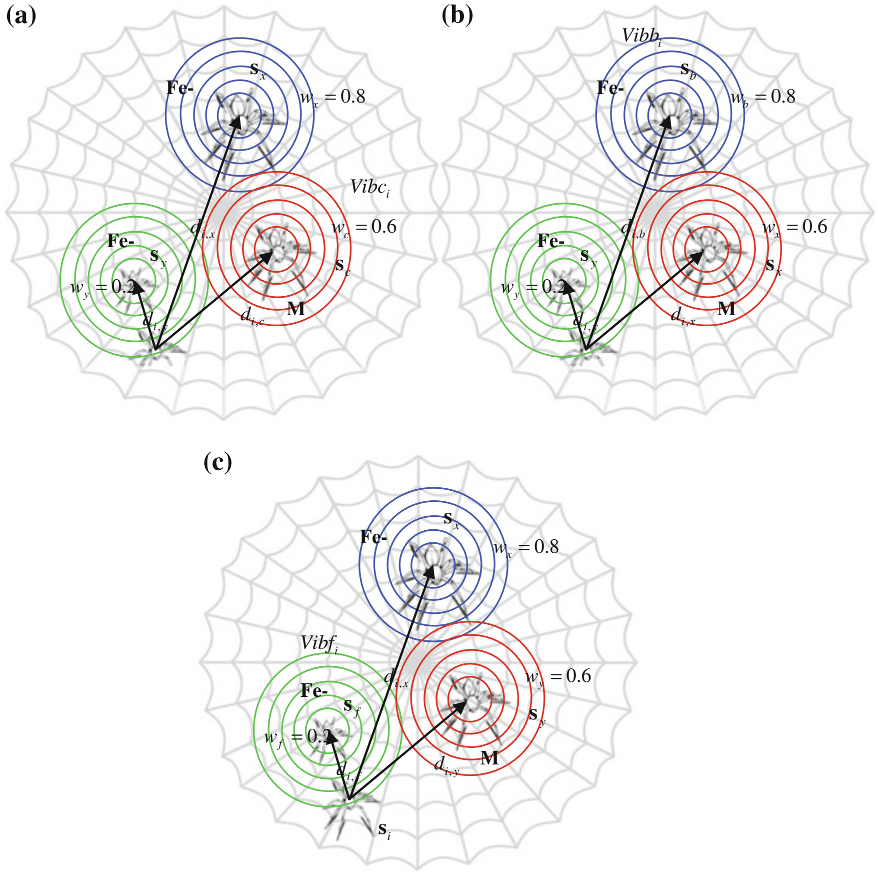


Fig. 1 Configuration of each special relation: **a** $Vibc_i$, **b** $Vibb_i$ and **c** $Vibf_i$

$$Vibf_i = w_f \cdot e^{-d_{i,f}^2} \tag{8}$$

Figure 1 shows the configuration of each special relationship: (a) $Vibc_i$, (b) $Vibb_i$ and (c) $Vibf_i$.

3.3 Initializing the Population

Like other evolutionary algorithms, the SSO is an iterative process whose first step is to randomly initialize the entire population (female and male). The algorithm begins by initializing the set $\mathbf{S}$ of N spider positions. Each spider position, $\mathbf{f}_i$ or $\mathbf{m}_i$, is a n-dimensional vector containing the parameter values to be optimized. Such values are randomly and uniformly distributed between the pre-specified lower initial parameter bound p_j^{low} and the upper initial parameter bound p_j^{high}, just as it described by the following expressions:

$$\begin{aligned} f_{i,j}^0 &= p_j^{low} + \text{rand}(0,1) \cdot (p_j^{high} - p_j^{low}) \\ i &= 1,2,\ldots,N_f;\ j = 1,2,\ldots,n \\ m_{k,j}^0 &= p_j^{low} + \text{rand}(0{,}1) \cdot (p_j^{high} - p_j^{low}) \\ k &= 1,2,\ldots,N_m;\ j = 1,2,\ldots,n, \end{aligned} \tag{9}$$

where j, i and k are the parameter and individual indexes respectively whereas zero signals the initial population. The function rand(0,1) generates a random number between 0 and 1. Hence, $f_{i,j}$ is the j-th parameter of the i-th female spider position.

3.4 Cooperative Operators

3.4.1 Female Cooperative Operator

Social-spiders perform cooperative interaction over other colony members. The way in which this behavior takes place depends on the spider gender. Female spiders present an attraction or dislike over others irrespective of gender. For a particular female spider, such attraction or dislike is commonly developed over other spiders according to their vibrations which are emitted over the communal web. Since vibrations depend on the weight and distance of the members which have originated them, strong vibrations are produced either by big spiders or other neighboring members lying nearby the individual which is perceiving them. The final decision of attraction or dislike over a determined member is taken considering an internal state which is influenced by several factors such as reproduction cycle, curiosity and other random phenomena.

In order to emulate the cooperative behavior of the female spider, a new operator is defined. The operator considers the position change of the female spider i at each iteration. Such position change, which can be of attraction or repulsion, is computed as a combination of three different elements. The first one involves the change in regard to the nearest member to i that holds a higher weight and produces the vibration $Vibc_i$. The second one considers the change regarding the best individual of the entire population $\mathbf{S}$ who produces the vibration $Vibb_i$. Finally, the third one incorporates a random movement.

Since the final movement of attraction or repulsion depends on several random phenomena, the selection is modeled as a stochastic decision. For this operation, a uniform random number r_m is generated within the range [0,1]. If r_m is smaller than a threshold PF, an attraction movement is generated; otherwise, a repulsion movement is produced. Therefore, such operator can be modeled as follows:

$$\mathbf{f}_i^{k+1} = \begin{cases} \mathbf{f}_i^k + \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) + \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{with probability } PF \\ \mathbf{f}_i^k - \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) - \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{with probability } 1 - PF \end{cases}, \tag{10}$$

where α, β, δ and rand are random numbers between [0,1] whereas k represents the iteration number. The individual $\mathbf{s}_c$ and $\mathbf{s}_b$ represent the nearest member to i that holds a higher weight and the best individual of the entire population $\mathbf{S}$, respectively.

Under this operation, each particle presents a movement which combines the past position that holds the attraction or repulsion vector over the local best element $\mathbf{s}_c$ and the global best individual $\mathbf{s}_b$ seen so-far. This particular type of interaction avoids the quick concentration of particles at only one point and encourages each particle to search around the local candidate region within its neighborhood $(\mathbf{s}_c)$, rather than interacting to a particle $(\mathbf{s}_b)$ in a distant region of the domain. The use of this scheme has two advantages. First, it prevents the particles from moving towards the global best position, making the algorithm less susceptible to premature convergence. Second, it encourages particles to explore their own neighborhood thoroughly before converging towards the global best position. Therefore, it provides the algorithm with global search ability and enhances the exploitative behavior of the proposed approach.

3.4.2 Male Cooperative Operator

According to the biological behavior of the social-spider, male population is divided into two classes: dominant and non-dominant male spiders. Dominant male spiders have better fitness characteristics (usually regarding the size) in comparison to non-dominant. Dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate in the center of the male population as a strategy to take advantage of resources that are wasted by dominant males.

For emulating such cooperative behavior, the male members are divided into two different groups (dominant members **D** and non-dominant members **ND**) according to their position with regard to the median member. Male members, with a weight value above the median value within the male population, are considered the dominant individuals **D**. On the other hand, those under the median value are labeled as non-dominant **ND** males. In order to implement such computation, the male population $\mathbf{M}$ $(\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_m}\})$ is arranged according to their weight value in decreasing order. Thus, the individual whose weight w_{N_f+m} is located in the middle is considered the median male member. Since indexes of the male population $\mathbf{M}$ in regard to the entire population $\mathbf{S}$ are increased by the number of female members N_f, the median weight is indexed by $N_f + m$. According to this, change of positions for the male spider can be modeled as follows:

$$\mathbf{m}_i^{k+1} = \begin{cases} \mathbf{m}_i^k + \alpha \cdot Vibf_i \cdot (\mathbf{s}_f - \mathbf{m}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{if } w_{N_f+i} > w_{N_f+m} \\ \mathbf{m}_i^k + \alpha \cdot \left(\frac{\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - \mathbf{m}_i^k \right) & \text{if } w_{N_f+i} \leq w_{N_f+m} \end{cases} \tag{11}$$

where the individual $\mathbf{s}_f$ represents the nearest female individual to the male member i whereas $\left(\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h} / \sum_{h=1}^{N_m} w_{N_f+h}\right)$ correspond to the weighted mean of the male population **M**.

By using this operator, two different behaviors are produced. First, the set **D** of particles is attracted to others in order to provoke mating. Such behavior allows incorporating diversity into the population. Second, the set **ND** of particles is attracted to the weighted mean of the male population **M.** This fact is used to partially control the search process according to the average performance of a subgroup of the population. Such mechanism acts as a filter which avoids that very good individuals or extremely bad individuals influence the search process.

3.5 Mating Operator

Mating in a social-spider colony is performed by dominant males and the female members. Under such circumstances, when a dominant male $\mathbf{m}_g$ spider ($g \in \mathbf{D}$) locates a set $\mathbf{E}^g$ of female members within a specific range r (range of mating), it mates, forming a new brood $\mathbf{s}_{new}$ which is generated considering all the elements of the set $\mathbf{T}^g$ that, in turn, has been generated by the union $\mathbf{E}^g \cup \mathbf{m}_g$. It is important to emphasize that if the set $\mathbf{E}^g$ is empty, the mating operation is canceled. The range r is defined as a radius which depends on the size of the search space. Such radius r is computed according to the following model:

$$r = \frac{\sum_{j=1}^{n} (p_j^{high} - p_j^{low})}{2 \cdot n} \tag{12}$$

In the mating process, the weight of each involved spider (elements of $\mathbf{T}^g$) defines the probability of influence for each individual into the new brood. The spiders holding a heavier weight are more likely to influence the new product, while elements with lighter weight have a lower probability. The influence probability Ps_i of each member is assigned by the Roulette method, which is defined as follows:

$$Ps_i = \frac{w_i}{\sum_{j \in \mathbf{T}^k} w_j}, \tag{13}$$

where $i \in \mathbf{T}^g$.

Once the new spider is formed, it is compared to the new spider candidate $\mathbf{s}_{new}$ holding the worst spider $\mathbf{s}_{wo}$ of the colony, according to their weight values (where $w_{wo} = \min_{l \in \{1,2,\ldots,N\}}(w_l)$). If the new spider is better than the worst spider, the worst spider is replaced by the new one. Otherwise, the new spider is discarded and the

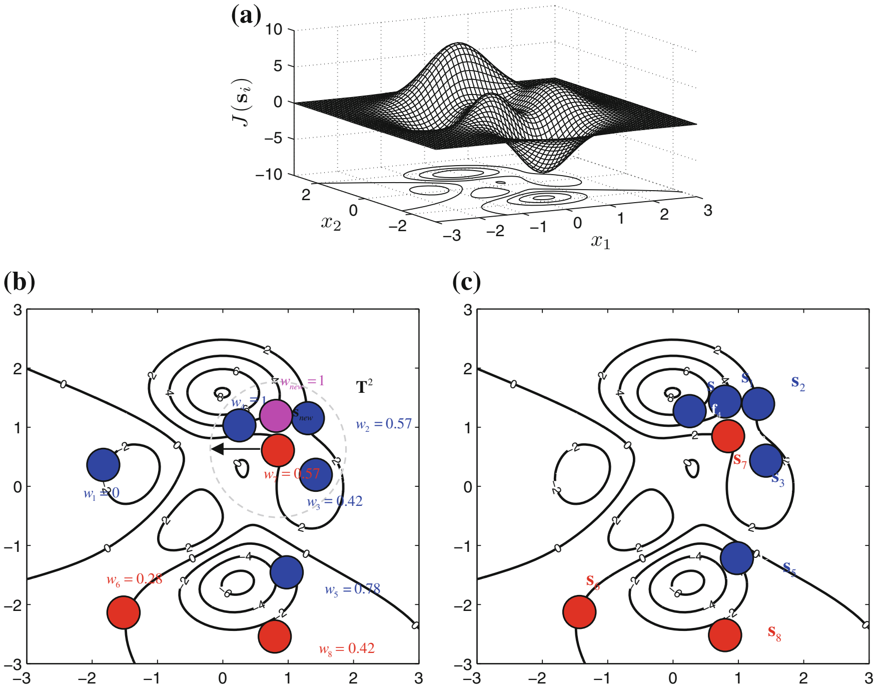
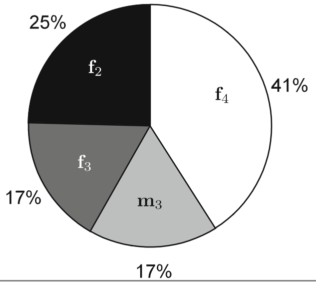


Fig. 2 Example of the mating operation: **a** optimization problem, **b** initial configuration *before* mating and **c** configuration *after* the mating operation

population does not suffer changes. In case of replacement, the new spider assumes the gender and index from the replaced spider. Such fact assures that the entire population **S** maintains the original rate between female and male members.

In order to demonstrate the mating operation, Fig. 2a illustrates a simple optimization problem. As an example, it is assumed a population **S** of eight different 2-dimensional members ($N = 8$), five females ($N_f = 5$) and three males ($N_m = 3$). Figure 2b shows the initial configuration of the proposed example with three different female members $\mathbf{f}_2(\mathbf{s}_2), \mathbf{f}_3(\mathbf{s}_3)$ and $\mathbf{f}_4(\mathbf{s}_4)$ constituting the set $\mathbf{E}^2$ which is located inside of the influence range r of a dominant male $\mathbf{m}_2(\mathbf{s}_7)$. Then, the new candidate spider $\mathbf{s}_{new}$ is generated from the elements $\mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4$ and $\mathbf{m}_2$ which constitute the set $\mathbf{T}^2$. Therefore, the value of the first decision variable $s_{new,1}$ for the new spider is chosen by means of the roulette mechanism considering the values already existing from the set $\{f_{2,1}, f_{3,1}, f_{4,1}, m_{2,1}\}$. The value of the second decision variable $s_{new,2}$ is also chosen in the same manner. Table 1 shows the data for constructing the new spider through the Roulette method. Once the new spider $\mathbf{s}_{new}$ is formed, its weight w_{new} is calculated. As $\mathbf{s}_{new}$ is better than the worst member $\mathbf{f}_1$ that is present in the population **S**, $\mathbf{f}_1$ is replaced by $\mathbf{s}_{new}$. Therefore, $\mathbf{s}_{new}$ assumes the same gender and index from $\mathbf{f}_1$. Figure 2c shows the configuration of **S** after the mating process.

Table 1 Data for constructing the new spider $\mathbf{s}_{new}$ through the Roulette method

Spider		Position	w_i	Ps_i	Roulette
$\mathbf{s}_1$	$\mathbf{f}_1$	(−1.9,0.3)	0.00	–	
$\mathbf{s}_2$	$\mathbf{f}_2$	(1.4,1.1)	0.57	0.22	
$\mathbf{s}_3$	$\mathbf{f}_3$	(1.5,0.2)	0.42	0.16	
$\mathbf{s}_4$	$\mathbf{f}_4$	(0.4,1.0)	1.00	0.39	
$\mathbf{s}_5$	$\mathbf{f}_5$	(1.0,−1.5)	0.78	–	
$\mathbf{s}_6$	$\mathbf{m}_1$	(−1.3,−1.9)	0.28	–	
$\mathbf{s}_7$	$\mathbf{m}_2$	(0.9,0.7)	0.57	0.22	
$\mathbf{s}_8$	$\mathbf{m}_3$	(0.8,−2.6)	0.42	–	
$\mathbf{s}_{new}$		(0.9,1.1)	1.00	–	

Under this operation, new generated particles locally exploit the search space inside the mating range in order to find better individuals.

3.6 Computational Procedure

The computational procedure for the proposed algorithm can be summarized as follows:

Step 1:	Considering N as the total number of n-dimensional colony members, define the number of male N_m and females N_f spiders in the entire population $\mathbf{S}$.
	$N_f = \text{floor}[(0.9 - \text{rand} \cdot 0.25) \cdot N]$ and $N_m = N - N_f$, where rand is a random number between [0,1] whereas floor$(\cdot)$ maps a real number to an integer number.
Step 2:	Initialize randomly the female ($\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}\}$) and male ($\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_m}\}$) members (where $\mathbf{S} = \{\mathbf{s}_1 = \mathbf{f}_1, \mathbf{s}_2 = \mathbf{f}_2, \ldots, \mathbf{s}_{N_f} = \mathbf{f}_{N_f}, \mathbf{s}_{N_f+1} = \mathbf{m}_1, \mathbf{s}_{N_f+2} = \mathbf{m}_2, \ldots, \mathbf{s}_N = \mathbf{m}_{N_m}\}$ and calculate the radius of mating.
	$r = \frac{\sum_{j=1}^{n}(p_j^{high} - p_j^{low})}{2 \cdot n}$
	for (i = 1;i < N_f + 1;i++)
	for(j = 1;j < n+1;j++)
	$f_{i,j}^0 = p_j^{low} + \text{rand}(0,1) \cdot (p_j^{high} - p_j^{low})$
	end for
	end for
	for (k = 1;k < N_m + 1;k++)
	for(j = 1;j < n + 1;j++)
	$m_{k,j}^0 = p_j^{low} + \text{rand} \cdot (p_j^{high} - p_j^{low})$
	end for
	end for

(continued)

(continued)

```
Step 3: Calculate the weight of every spider of $\mathbf{S}$ (Sect. 3.1).
        for ($i = 1; i < N+1; i$ ++)
            $w_i = \frac{J(\mathbf{s}_i)-worst_{\mathbf{S}}}{best_{\mathbf{S}}-worst_{\mathbf{S}}}$ where $best_{\mathbf{S}} = \max_{k\in\{1,2,\ldots,N\}}(J(\mathbf{s}_k))$ and
            $worst_{\mathbf{S}} = \min_{k\in\{1,2,\ldots,N\}}(J(\mathbf{s}_k))$
        end for
Step 4: Move female spiders according to the female cooperative operator (Sect. 3.4).
        for ($i = 1; i < N_f + 1; i$++)
            Calculate $Vibc_i$ and $Vibb_i$ (Sect. 3.2)
            If ($r_m < PF$); where $r_m \in \text{rand}(0,1)$
                $\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) + \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$
            else if
                $\mathbf{f}_i^{k+1} = \mathbf{f}_i^k - \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) - \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$
            end if
        end for
Step 5: Move the male spiders according to the male cooperative operator (Sect. 3.4).
        Find the median male individual ($w_{N_f+m}$) from $\mathbf{M}$.
        for ($i = 1; i < N_m + 1; i$ ++)
            Calculate $Vibf_i$ (Sect. 3.2)
            If ($w_{N_f+i} > w_{N_f+m}$)
                $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha \cdot Vibf_i \cdot (\mathbf{s}_f - \mathbf{m}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$
            Else if
                $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha \cdot \left( \frac{\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - \mathbf{m}_i^k \right)$
            end if
        end for
Step 6: Perform the mating operation (Sect. 3.5).
        for ($i = 1; i < N_m + 1; i$ ++)
            If ($\mathbf{m}_i \in \mathbf{D}$)
                Find $\mathbf{E}^i$
                If ($\mathbf{E}^i$ is not empty)
                    Form $\mathbf{s}_{new}$ using the Roulette method
                    If ($w_{new} > w_{wo}$)
                        $\mathbf{s}_{wo} = \mathbf{s}_{new}$
                    end if
                end if
            end if
        end for
Step 7: If the stop criteria is met, the process is finished; otherwise, go back to Step 3
```

3.7 Discussion About the SSO Algorithm

Evolutionary algorithms (EA) have been widely employed for solving complex optimization problems. These methods are found to be more powerful than conventional methods based on formal logics or mathematical programming [40]. In an EA algorithm, search agents have to decide whether to explore unknown search positions or to exploit already tested positions in order to improve their solution quality. Pure exploration degrades the precision of the evolutionary process but increases its capacity to find new potential solutions. On the other hand, pure exploitation allows refining existent solutions but adversely drives the process to local optimal solutions. Therefore, the ability of an EA to find a global optimal solutions depends on its capacity to find a good balance between the exploitation of found-so-far elements and the exploration of the search space [7]. So far, the exploration–exploitation dilemma has been an unsolved issue within the framework of evolutionary algorithms.

EA defines individuals with the same property, performing virtually the same behavior. Under these circumstances, algorithms waste the possibility to add new and selective operators as a result of considering individuals with different characteristics. These operators could incorporate computational mechanisms to improve several important algorithm characteristics such as population diversity or searching capacities.

On the other hand, PSO and ABC are the most popular swarm algorithms for solving complex optimization problems. However, they present serious flaws such as premature convergence and difficulty to overcome local minima [35, 36]. Such problems arise from operators that modify individual positions. In such algorithms, the position of each agent in the next iteration is updated yielding an attraction towards the position of the best particle seen so-far (in case of PSO) or any other randomly chosen individual (in case of ABC). Such behaviors produce that the entire population concentrates around the best particle or diverges without control as the algorithm evolves, either favoring the premature convergence or damaging the exploration-exploitation balance [3, 37].

Different to other EA, at SSO each individual is modeled considering the gender. Such fact allows incorporating computational mechanisms to avoid critical flaws such as premature convergence and incorrect exploration-exploitation balance commonly present in both, the PSO and the ABC algorithm. From an optimization point of view, the use of the social-spider behavior as a metaphor introduces interesting concepts in EA: the fact of dividing the entire population into different search-agent categories and the employment of specialized operators that are applied selectively to each of them. By using this framework, it is possible to improve the balance between exploitation and exploration, yet preserving the same population, i.e. individuals who have achieved efficient exploration (female spiders) and individuals that verify extensive exploitation (male spiders). Furthermore, the social-spider behavior mechanism introduces an interesting computational scheme with three important particularities: first, individuals are separately processed according to their characteristics. Second, operators share the same communication

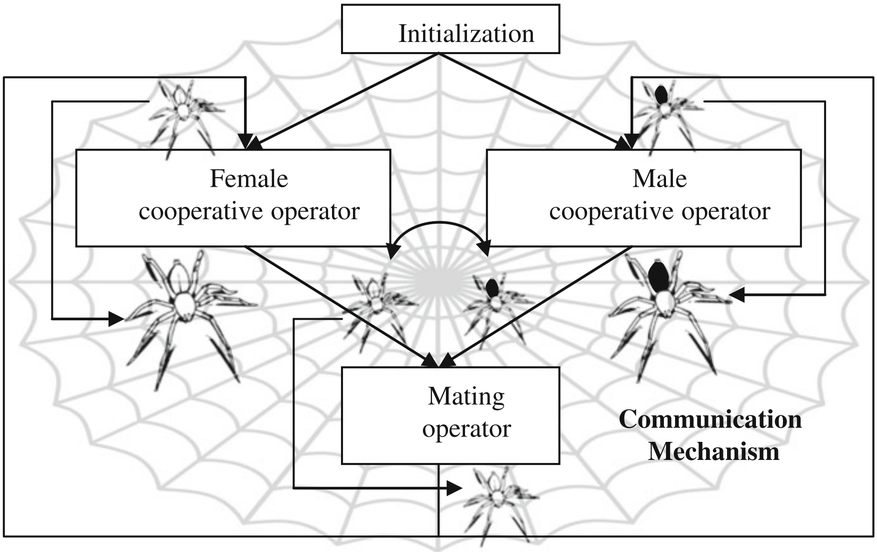


Fig. 3 Schematic representation of the SSO algorithm-data-flow

mechanism allowing the employment of important information of the evolutionary process to modify the influence of each operator. Third, although operators modify the position of only an individual type, they use global information (positions of all individual types) in order to perform such modification. Figure 3 presents a schematic representation of the algorithm-data-flow. According to Fig. 3, the female cooperative and male cooperative operators process only female or male individuals, respectively. However, the mating operator modifies both individual types.

4 Experimental Results

A comprehensive set of 19 functions, which have been collected from Refs. [18, 9, 21, 24, 31, 34, 39], has been used to test the performance of the proposed approach. Table 4 in the Appendix A presents the benchmark functions used in our experimental study. In the table, n indicates the function dimension, $f(\mathbf{x}^*)$ the optimum value of the function, $\mathbf{x}^*$ the optimum position and S the search space (subset of R^n). A detailed description of each function is given in the Appendix A.

4.1 Performance Comparison to Other Swarm Algorithms

We have applied the SSO algorithm to 19 functions whose results have been compared to those produced by the Particle Swarm Optimization (PSO) method

Table 2 Minimization results of benchmark functions of Table 4 with $n = 30$. Maximum number of iterations = 1,000

		SSO	ABC	PSO
$f_1(x)$	**AB**	**1.96E−03**	2.90E−03	1.00E+03
	MB	2.81E−03	1.50E−03	**2.08E−09**
	SD	**9.96E−04**	1.44E−03	3.05E+03
$f_2(x)$	**AB**	**1.37E−02**	1.35E−01	5.17E+01
	MB	**1.34E−02**	1.05E−01	5.00E+01
	SD	**3.11E−03**	8.01E−02	2.02E+01
$f_3(x)$	**AB**	**4.27E–02**	1.13E+00	8.63E+04
	MB	**3.49E−02**	6.11E−01	8.00E+04
	SD	**3.11E−02**	1.57E+00	5.56E+04
$f_4(x)$	**AB**	**5.40E−02**	5.82E+01	1.47E+01
	MB	**5.43E−02**	5.92E+01	1.51E+01
	SD	**1.01E−02**	7.02E+00	3.13E+00
$f_5(x)$	**AB**	**1.14E+02**	1.38E+02	3.34E+04
	MB	**5.86E+01**	1.32E+02	4.03E+02
	SD	**3.90E+01**	1.55E+02	4.38E+04
$f_6(x)$	**AB**	**2.68E−03**	4.06E−03	1.00E+03
	MB	2.68E−03	3.74E−03	**1.66E–09**
	SD	**6.05E–04**	2.98E−03	3.06E+03
$f_7(x)$	**AB**	**1.20E+01**	1.21E+01	1.50E+01
	MB	**1.20E+01**	1.23E+01	1.37E+01
	SD	**5.76E−01**	9.00E−01	4.75E+00
$f_8(x)$	**AB**	**2.14E+00**	3.60E+00	3.12E+04
	MB	**3.64E+00**	8.04E−01	2.08E+02
	SD	**1.26E+00**	3.54E+00	5.74E+04
$f_9(x)$	**AB**	**6.92E−05**	1.44E−04	2.47E+00
	MB	**6.80E−05**	8.09E−05	9.09E−01
	SD	**4.02E−05**	1.69E−04	3.27E+00
$f_{10}(x)$	**AB**	**4.44E−04**	1.10E−01	6.93E+02
	MB	**4.05E−04**	4.97E−02	5.50E+02
	SD	**2.90E−04**	1.98E−01	6.48E+02
$f_{11}(x)$	**AB**	**6.81E+01**	3.12E+02	4.11E+02
	MB	**6.12E+01**	3.13E+02	4.31E+02
	SD	**3.00E+01**	4.31E+01	1.56E+02
$f_{12}(x)$	**AB**	**5.39E−05**	1.18E−04	4.27E+07
	MB	**5.40E−05**	1.05E−04	1.04E−01
	SD	**1.84E−05**	8.88E−05	9.70E+07
$f_{13}(x)$	**AB**	**1.76E−03**	1.87E−03	5.74E−01
	MB	**1.12E−03**	1.69E−03	1.08E−05
	SD	**6.75E−04**	1.47E−03	2.36E+00

(continued)

Table 2 (continued)

		SSO	ABC	PSO
$f_{14}(x)$	**AB**	**−9.36E+02**	−9.69E+02	−9.63E+02
	MB	**−9.36E+02**	−9.60E+02	−9.92E+02
	SD	**1.61E+01**	6.55E+01	6.66E+01
$f_{15}(x)$	**AB**	**8.59E+00**	2.64E+01	1.35E+02
	MB	**8.78E+00**	2.24E+01	1.36E+02
	SD	**1.11E+00**	1.06E+01	3.73E+01
$f_{16}(x)$	**AB**	**1.36E−02**	6.53E−01	1.14E+01
	MB	**1.39E−02**	6.39E−01	1.43E+01
	SD	**2.36E−03**	3.09E−01	8.86E+00
$f_{17}(x)$	**AB**	**3.29E–03**	5.22E−02	1.20E+01
	MB	**3.21E−03**	4.60E−02	1.35E−02
	SD	**5.49E−04**	3.42E–02	3.12E+01
$f_{18}(x)$	**AB**	**1.87E+00**	2.13E+00	1.26E+03
	MB	**1.61E+00**	2.14E+00	5.67E+02
	SD	**1.20E+00**	1.22E+00	1.12E+03
$f_{19}(x)$	**AB**	**2.74E−01**	4.14E+00	1.53E+00
	MB	**3.00E−01**	4.10E+00	5.50E−01
	SD	**5.17E−02**	4.69E−01	2.94E+00

[20] and the Artificial Bee Colony (ABC) algorithm [17]. These are considered as the most popular swarm algorithms for many optimization applications. In all comparisons, the population has been set to 50 individuals. The maximum iteration number for all functions has been set to 1,000. Such stop criterion has been selected to maintain compatibility to similar works reported in the literature [42].

The parameter setting for each algorithm in the comparison is described as follows:

1. PSO: The parameters are set to $c_1 = 2$ and $c_2 = 2$; besides, the weight factor decreases linearly from 0.9 to 0.2 [20].
2. ABC: The algorithm has been implemented using the guidelines provided by its own reference [17], using the parameter *limit* = 100.
3. SSO: Once it has been determined experimentally, the parameter *PF* has been set to 0.7. It is kept for all experiments in this section.

The experiment compares the SSO to other algorithms such as PSO and ABC. The results for 30 runs are reported in Table 2 considering the following performance indexes: the Average Best-so-far (AB) solution, the Median Best-so-far (MB) and the Standard Deviation (SD) of best-so-far solution. The best outcome for each function is boldfaced. According to this table, SSO delivers better results than PSO and ABC for all functions. In particular, the test remarks the largest difference in performance which is directly related to a better trade-off between exploration and exploitation.

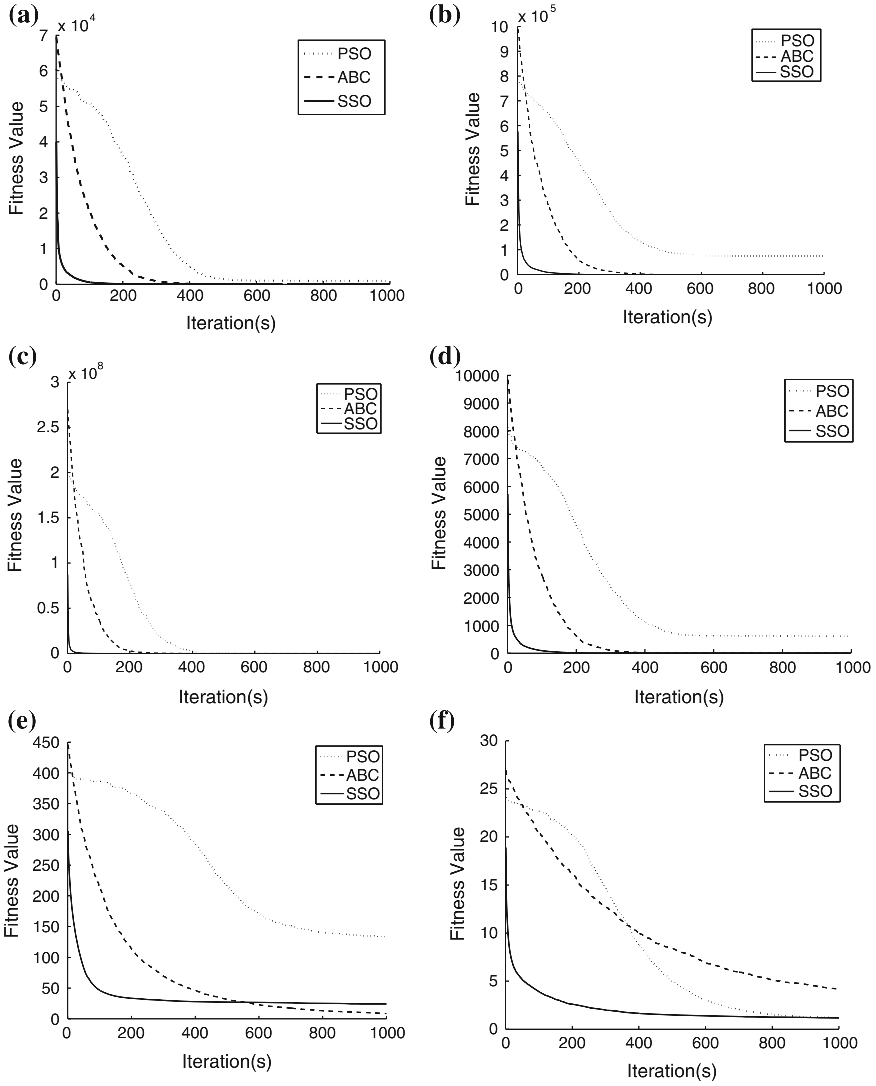


Fig. 4 Evolution curves for PSO, ABC and the proposed algorithm considering as examples the functions **a** f_1, **b** f_3, **c** f_5, **d** f_{10}, **e** f_{15} and **f** f_{19} from the experimental set

Figure 4 presents the evolution curves for PSO, ABC and the proposed algorithm considering as examples the functions f_1,f_3,f_5,f_{10},f_{15} and f_{19} from the experimental set. Among them, the rate of convergence of SSO is the fastest, which finds the best solution in less of 400 iterations on average while the other three algorithms need much more iterations. A non-parametric statistical significance proof known as the Wilcoxon's rank sum test for independent samples [10, 38] has been conducted over the "average best-so-far" (AB) data of Table 2, with an 5 %

Table 3 *p*-values produced by Wilcoxon's test comparing SSO versus ABC and SSO versus PSO, over the "average best-so-far" (AB) values from Table 2

Function	SSO versus ABC	SSO versus PSO
$f_1(x)$	0.041	1.8E−05
$f_2(x)$	0.048	0.059
$f_3(x)$	5.4E−04	6.2E−07
$f_4(x)$	1.4E−07	4.7E−05
$f_5(x)$	0.045	7.1E−07
$f_6(x)$	2.3E−04	5.5E−08
$f_7(x)$	0.048	0.011
$f_8(x)$	0.017	0.043
$f_9(x)$	8.1E−04	2.5E−08
$f_{10}(x)$	4.6E−06	1.7E−09
$f_{11}(x)$	9.2E−05	7.8E−06
$f_{12}(x)$	0.022	1.1E−10
$f_{13}(x)$	0.048	2.6E−05
$f_{14}(x)$	0.044	0.049
$f_{15}(x)$	4.5E−05	7.9E−08
$f_{16}(x)$	2.8E−05	4.1E−06
$f_{17}(x)$	7.1E−04	6.2E−10
$f_{18}(x)$	0.013	8.3E−10
$f_{19}(x)$	4.9E−05	5.1E−08

significance level. Table 3 reports the *p*-values produced by Wilcoxon's test for the pair-wise comparison of the "average best so-far" of two groups. Such groups are constituted by SSO versus PSO and SSO versus ABC. As a null hypothesis, it is assumed that there is no significant difference between mean values of the two algorithms. The alternative hypothesis considers a significant difference between the "average best-so-far" values of both approaches. All *p*-values reported in Table 3 are less than 0.05 (5 % significance level) which is a strong evidence against the null hypothesis. Therefore, such evidence indicates that SSO results are statistically significant and it has not occurred by coincidence (i.e. due to common noise contained in the process).

5 Conclusions

In this paper, a novel swarm algorithm called the Social Spider Optimization (SSO) has been proposed for solving optimization tasks. The SSO algorithm is based on the simulation of the cooperative behavior of social-spiders whose individuals emulate a group of spiders which interact to each other based on the biological laws of a cooperative colony. The algorithm considers two different search agents (spiders): male and female. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors within the colony.

In contrast to most of existent swarm algorithms, the proposed approach models each individual considering two genders. Such fact allows not only to emulate the cooperative behavior of the colony in a realistic way, but also to incorporate computational mechanisms to avoid critical flaws commonly delivered by the popular PSO and ABC algorithms, such as the premature convergence and the incorrect exploration-exploitation balance.

SSO has been experimentally tested considering a suite of 19 benchmark functions. The performance of SSO has been also compared to the following swarm algorithms: the Particle Swarm Optimization method (PSO) [20], and the Artificial Bee Colony (ABC) algorithm [17]. Results have confirmed a acceptable performance of the proposed method in terms of the solution quality of the solution for all tested benchmark functions.

The SSO's remarkable performance is associated with two different reasons: (i) their operators allow a better particle distribution in the search space, increasing the algorithm's ability to find the global optima; and (ii) the division of the population into different individual types, provides the use of different rates between exploration and exploitation during the evolution process.

Appendix A. List of Benchmark Functions

See Table 4

Table 4 Test functions used in the experimental study

Name	Function	S	Dim	Minimum
Sphere	$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Schwefel 2.22	$f_2(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Schwefel 1.2	$f_3(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
F4	$f_4(\mathbf{x}) = 418.9829n + \sum_{i=1}^{n} \left(-x_i \sin\left(\sqrt{\lvert x_i \rvert} \right) \right)$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Rosenbrock	$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^n$	$n = 30$	$\mathbf{x}^* = (1, \ldots, 1)$; $f(\mathbf{x}^*) = 0$
Step	$f_6(\mathbf{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Quartic	$f_7(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4 + random(0, 1)$	$[-1.28, 1.28]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Dixon and price	$f_8(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=1}^{n} i\left(2x_i^2 - x_{i-1}\right)^2$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \ldots, 0)$; $f(\mathbf{x}^*) = 0$
Levy	$f_9(\mathbf{x}) = 0.1 \left\{ \begin{array}{l} \sin^2(3\pi x_1) \\ + \sum_{i=1}^{n} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] \\ + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \end{array} \right\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$; $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (1, \ldots, 1)$; $f(\mathbf{x}^*) = 0$

(continued)

Table 4 (continued)

Name	Function	S	Dim	Minimum
Sum of squares	$f_{10}(\mathbf{x}) = \sum_{i=1}^{n} ix_i^2$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Zakharov	$f_{11}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	$[-5, 10]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Penalized	$f_{12}(\mathbf{x}) = \frac{\pi}{n}\left\{\begin{array}{l} 10\sin(\pi y_1) + \\ \sum_{i=1}^{n-1} (y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2 \end{array}\right\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{(x_i + 1)}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < a \end{cases}$	$[-50, 50]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Penalized 2	$f_{13}(\mathbf{x}) = 0.1\left\{\begin{array}{l} \sin^2(3\pi x_1) \\ +\sum_{i=1}^{n} \begin{array}{l} (x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] \\ +(x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right] \end{array} \end{array}\right\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ where $u(x_i, a, k, m)$ is the same as Penalized function.	$[-50, 50]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Schwefel	$f_{14}(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]^n$	$n = 30$	$\mathbf{x}^* = (420,\ldots,420)$; $f(\mathbf{x}^*) = -418.9829 \times n$
Rastrigin	$f_{15}(\mathbf{x}) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12, 5.12]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Ackley	$f_{16}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $- \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + \exp$	$[-32, 32]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Griewank	$f_{17}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Powelll	$f_{18}(\mathbf{x}) = \sum_{i=1}^{n/k} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2$ $+ (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	$[-4, 5]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$
Salomon	$f_{19}(\mathbf{x}) = -\cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2 + 1}$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0,\ldots,0)$; $f(\mathbf{x}^*) = 0$

References

1. Aviles, L.: Sex-ratio bias and possible group selection in the social spider Anelosimus eximius. Am. Nat. **128**(1), 1–12 (1986)
2. Avilés, L.: Causes and consequences of cooperation and permanent-sociality in spiders. In: Choe, B.C. (ed.) The Evolution of Social Behavior in Insects and Arachnids, pp. 476–498. Cambridge University Press, Cambridge (1997)
3. Banharnsakun, A., Achalakul, T., Sirinaovakul, B.: The best-so-far selection in artificial bee colony algorithm. Appl. Soft Comput. **11**, 2888–2901 (2011)
4. Bonabeau, E.: Social insect colonies as complex adaptive systems. Ecosystems **1**, 437–443 (1998)
5. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
6. Burgess, J.W.: Social spacing strategies in spiders. In: Rovner, P.N. (ed.) Spider communication: mechanisms and ecological significance, pp. 317–351. Princeton University Press, Princeton (1982)

7. Chen, D.B., Zhao, C.X.: Particle swarm optimization with adaptive population size and its application. Appl. Soft Comput. **9**(1), 39–48 (2009)
8. Damian, O., Andrade, M., Kasumovic, M.: Dynamic population structure and the evolution of spider mating systems. Adv. Insect Physiol. **41**, 65–114 (2011)
9. Duan, X., Wang, G.G., Kang, X., Niu, Q., Naterer, G., Peng, Q.: Performance study of mode-pursuing sampling method. Eng. Optim. **41**(1) (2009)
10. Garcia, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J. Heurist. (2008). doi:10.1007/s10732-008-9080-4
11. Gordon, D.: The organization of work in social insect colonies. Complexity **8**(1), 43–46 (2003)
12. Gove, R., Hayworth, M., Chhetri, M., Rueppell, O.: Division of labour and social insect colony performance in relation to task and mating number under two alternative response threshold models. Insectes Soc. **56**(3), 19–331 (2009)
13. Hossein, A., Hossein-Alavi, A.: Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul. **17**, 4831–4845 (2012)
14. Hölldobler, B., Wilson, E.O.: The Ants. Harvard University Press (1990). ISBN 0-674-04075-9
15. Hölldobler, B., Wilson, E.O.: Journey to the Ants: A Story of Scientific Exploration (1994). ISBN 0-674-48525-4
16. Jones, T., Riechert, S.: Patterns of reproductive success associated with social structure and microclimate in a spider system. Anim. Behav. **76**(6), 2011–2019 (2008)
17. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06. Engineering Faculty, Computer Engineering Department, Erciyes University (2005)
18. Karaboga, D, Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. **214**(1), 108–132 (2009). ISSN 0096-3003
19. Kassabalidis, I., El-Sharkawi, M.A., Marks, R.J., Arabshahi, P., Gray, A.A.: Swarm intelligence for routing in communication networks. Global Telecommunications Conference, GLOBECOM'01, 6, IEEE, pp. 3613–3617 (2001)
20. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, December 1995
21. Krishnanand, K.R., Nayak, S.K., Panigrahi, B.K., Rout, P.K.: Comparative study of five bio-inspired evolutionary optimization techniques. In: Nature & Biologically Inspired Computing, NaBIC, World Congress on, pp.1231–1236 (2009)
22. Lubin, T.B.: The evolution of sociality in spiders. In: Brockmann, H.J. (ed.) Advances in the Study of Behavior, vol. 37, pp. 83–145. Academic Press, Burlington (2007)
23. Maxence, S.: Social organization of the colonial spider Leucauge sp. in the Neotropics: vertical stratification within colonies. J. Arachnol. **38**, 446–451 (2010)
24. 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 (GECCO '06). ACM, New York, NY, USA, pp. 485–492 (2006)
25. Oster, G., Wilson, E.: Caste and ecology in the social insects. Princeton University Press, Princeton (1978)
26. Pasquet, A.: Cooperation and prey capture efficiency in a social spider, Anelosimus eximius (Araneae, Theridiidae). Ethology **90**, 121–133 (1991)
27. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. **22**(3), 52–67 (2002)
28. Rajabioun, R.: Cuckoo optimization algorithm. Appl. Soft Comput. **11**, 5508–5518 (2011)
29. Rayor, E.C.: Do social spiders cooperate in predator defense and foraging without a web? Behav. Ecol. Sociobiol. **65**(10), 1935–1945 (2011)
30. Rypstra, A.: Prey size, prey perishability and group foraging in a social spider. Oecologia **86** (1), 25–30 (1991)

31. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristicfor global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1995)
32. Uetz, G.W.: Colonial web-building spiders: balancing the costs and benefits of group-living. In: Choe, E.J., Crespi, B. (eds.) The Evolution of Social Behavior in Insects and Arachnids, pp. 458–475. Cambridge University Press, Cambridge (1997)
33. Ulbrich, K., Henschel, J.: Intraspecific competition in a social spider. Ecol. Model. **115**(2–3), 243–251 (1999)
34. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Evolutionary Computation, 2004. CEC2004. Congress on 19–23 June, vol. 2, pp. 1980–1987 (2004)
35. Wan-Li, X., Mei-Qing, A.: An efficient and robust artificial bee colony algorithm for numerical optimization. Comput. Oper. Res. **40**, 1256–1265 (2013)
36. Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., Tian, Q.: Self-adaptive learning based particle swarm optimization. Inf. Sci. **181**(20), 4515–4538 (2011)
37. Wang, H., Sun, H., Li, C., Rahnamayan, S., Jeng-shyang, P.: Diversity enhanced particle swarm optimization with neighborhood. Inf. Sci. **223**, 119–135 (2013)
38. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1**, 80–83 (1945)
39. Yang, E., Barton, N.H., Arslan, T., Erdogan, A.T.: A novel shifting balance theory-based approach to optimization of an energy-constrained modulation scheme for wireless sensor networks. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1–6, 2008, Hong Kong, China, pp. 2749–2756. IEEE (2008)
40. Yang, X.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, Beckington (2008)
41. Yang, X.S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, Hoboken (2010)
42. Ying, J., Ke-Cun, Z., Shao-Jian, Q.: A deterministic global optimization algorithm. Appl. Math. Comput. **185**(1), 382–387 (2007)

Black Hole Algorithm and Its Applications

Santosh Kumar, Deepanwita Datta and Sanjay Kumar Singh

Abstract Bio-inspired computation is a field of study that connects together numerous subfields of connectionism (neural network), social behavior, emergence field of artificial intelligence and machine learning algorithms for complex problem optimization. Bio-inspired computation is motivated by nature and over the last few years, it has encouraged numerous advance algorithms and set of computational tools for dealing with complex combinatorial optimization problems. Black Hole is a new bio-inspired metaheuristic approach based on observable fact of black hole phenomena. It is a population based algorithmic approach like genetic algorithm (GAs), ant colony optimization (ACO) algorithm, particle swarm optimization (PSO), firefly and other bio-inspired computation algorithms. The objective of this book chapter is to provide a comprehensive study of black hole approach and its applications in different research fields like data clustering problem, image processing, data mining, computer vision, science and engineering. This chapter provides with the stepping stone for future researches to unveil how metaheuristic and bio-inspired commutating algorithms can improve the solutions of hard or complex problem of optimization.

Keywords Metaheuristic · Black hole · Swarm intelligence · K-means · Clustering

---

S. Kumar (✉) · D. Datta · S.K. Singh
Department of Computer Science and Engineering, Indian Institute of Technology
(Banaras Hindu University), Varanasi, India
e-mail: santosh.rs.cse12@iitbhu.ac.in

D. Datta
e-mail: welcomedeepanwita@gmail.com

S.K. Singh
e-mail: sks.cse@iitbhu.ac.in

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_7

1 Introduction

Bio-inspired computation and swarm intelligence based algorithms have attracted significant attention in recent years for solving the complex and combinatorial optimization problems of data clustering, feature selection and maximization of matching scores for authentication of human in biometrics [1] computer vision, data mining and machine learning based algorithms. Motivated from the natural and social behavioral phenomena, bio-inspired computation algorithms have significant research area during the recent years from both multidisciplinary research and the scientific research purpose. In the last 30 years, a great interest has been devoted to bio-inspired metaheuristics and it has encouraged and provides successful algorithms and computational simulated tools for dealing with complex and optimization problems (ISA Trans [2]). Several of these approaches are motivated from natural processes and generally start with an initial set of variables and then evolve to obtain the global minimum or maximum of the objective function and it has been an escalating interest in algorithms motivated by the behaviors of natural phenomena which are incorporated by many scientists and researchers to solve hard optimization problems. Hard problems cannot be solved to optimality, or to any guaranteed bound by any exact (deterministic) method within a '*reasonable*' time limit [3–10]. It is computational problems such as optimization of objective functions [11, 12] pattern recognition [1, 13] control objectives [2, 4, 14], image processing [15, 16] and filter modeling [17, 18] etc. There are different heuristic approaches that have been implemented by researches so far, for example Genetic algorithm [10] is the most well-known and mostly used evolutionary computation technique and it was developed in the early 1970s at the University of Michigan by John Holland and his students, whose re-search interests were devoted to the study of adaptive systems [19]. The basic genetic algorithm is very general and it can be implemented differently according to the problem: representation of solution (chromosomes), selection strategy, type of crossover (the recombination operator) and mutation operators. The fixed-length binary string is the most common representation of the chromosomes applied in GAs and a simple bit manipulation operation allow the implementation of crossover and mutation operations. Emphasis is mainly concentrated on crossover as the main variation operator that combines multiple (generally two) individuals that have been selected together by exchanging some of their parts. An exogenous parameter pc$\in$ [0.6, 1.0] (crossover rate) indicates the probability per individual to undergo crossover. After evaluating the fitness value of each individual in the selection pool, Individuals for producing offspring are selected using a selection strategy. A few of the popular selection schemes are mainly roulette-wheel selection, tournament selection and ranking selection, etc. After crossover operation, individuals are subjected to mutation process. Mutation initiates some randomness into the search space to prevent the optimization process from getting trapped into local optima. Naturally, the mutation rate is applied with less than 1 % probability but the appropriate value of the mutation rate for a given optimization problem is an open issue in research.

Simulated Annealing [9] is inspired by the annealing technique used by the different metallurgists to get a “well ordered” solid state of minimal energy (while avoiding the “meta stable” structures, characteristic of the local minima of energy), Ant Colony optimization (ACO) algorithm is a metaheuristic technique to solve problems that has been motivated by the ants’ social behaviors in finding shortest paths. Real ants walk randomly until they find food and return to their nest while depositing pheromone on the ground in order to mark their preferred path to attract other ants to follow [6, 20, 21], Particle Swarm Optimization (PSO) was introduced by James Kennedy and Russell Eberhart as a global optimization technique in 1995. It uses the metaphor of the flocking behavior of birds to solve optimization problems [22], firefly algorithm is a population based metaheuristic algorithm. It has become an increasingly important popular tool of Swarm Intelligence that has been applied in almost all research area so of optimization, as well as science and engineering practice. Fireflies have their flashing light. There are two fundamental functions of flashing light of firefly: (1) to attract mating partners and (2) to warn potential predators. But, the flashing lights comply with more physical rules. On the one hand, the light intensity of source (I) decrease as the distance (r) increases according to the term $I \propto 1/r^2$. This phenomenon inspired [23] to develop the firefly algorithm [23–25], Bat-inspired algorithm is a metaheuristic optimization algorithm. It was invented by Yang et al. [26–28] and it is based on the echolocation behavior of microbats with varying pulse rates of emission and loudness. And honey bee algorithm [29] etc. Such algorithms are progressively analyzed, deployed and powered by different researchers in many different research fields [3, 5, 27, 28, 30–32]. These algorithms are used to solve different optimization problems. But, there is no specific algorithm or direct algorithms to achieve the best solution for all optimization problems. Numerous algorithms give a better solution for some particular problems than others. Hence, searching for new heuristic optimization algorithms is an open problem [29] and it requires a lot of exploration of new metaheuristic algorithms for solving of hard problems.

Recently, one of the metaheuristic approaches has been developed for solving the hard or complex optimization and data clustering problem which is NP-hard problem known as black hole heuristic approach. Black Hole heuristic algorithm is inspired by the black hole phenomenon and black hole algorithm (BH) starts with an initial population of candidate solutions to an optimization problem and an objective function that is calculated for them similar to other population-based algorithms.

2 Heuristics Algorithm and Metaheuristic Algorithm

2.1 Heuristics Algorithm

The “*heuristic*” is Greek word and means *“to know*”, “*to find*”, “*to discover*” or “to guide an investigation” by trial and error methodology [33]. Specifically, heuristics are techniques which search for near-optimal solutions of problem at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is? [34]. The main algorithmic characteristic of heuristic is based on mimic physical or biological processes which are motivated by nature phenomena. Quality solution to complex optimization problems can be founded in reasonable amount of time however, there is no guarantee that is optimal solutions are reached. A heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods or deterministic approaches fail to provides any exact solution of a hard or complex problem. This is achieved by trading optimality, completeness, accuracy or precision for speed. Heuristic search exploits additional knowledge about the problem that helps direct search to more promising paths [23].

2.2 Metaheuristic Algorithm

Metaheuristic algorithms are the master strategy key that modify and update the other heuristic produced solution that is normally generated in the quest of local optimal. These nature-inspired algorithms are becoming popular and powerful in solving optimization problems. The suffix “meta” Greek means upper level methodology, beyond or higher level and it generally perform better than simple heuristic approach. Metaheuristic algorithms are conceptual set of all heuristic approach which is used to find the optimal solution of a combinatorial optimization problem. The term “metaheuristic” was introduced by Sir F. Glover in research paper. In addition, metaheuristic algorithms use certain trade-off of randomization and local search for finding the optimal and near optimal solution. Local search is a general methodology for finding the high quality solutions to complex or hard combinatorial optimization problems in reasonable amount of time. It is basically an iterative based search approach to diversification of neighbor of solutions trying to enhance the current solution by local changes [23, 35].

2.2.1 Characteristics of Metaheuristics

Each meta-heuristic search process depends on balancing between two major components which is involved through-out search of optimal solution. Two main components are known as diversification (exploration) and intensification (exploitation) [35].

2.2.2 Diversification or Exploration

Diversification phase ensures that the algorithm explores the search space more efficiently and thoroughly and helps to generate diverse solutions. On other hand, when diversification is too much, it will increase the probability of finding the true optimality globally solutions. But, it will often to slow the exploration process with much low rate of convergence of problem.

2.2.3 Intensification or Exploitation

It uses the local information in search process to generate better solutions of problems. If there is too much intensification, it will may lead to converge rapidly often to a local optimum or a wrong solution with respect to problem and reduce the probability of finding the global optimum solutions of a complex problem. Therefore, there is requirement of fine tuning or a fine balance and trade-off between intensification and diversification characteristics of metaheuristic approach. These metaheuristic techniques are in combination with the solutions of the best solutions of the complex combinatorial optimization problems. The main objective behind the best solution of metaheuristic ensures that the solution will converge to the optimization, while the diversification via randomization avoids the solution beings trapped or struck at local minima at the same time and increase the diversity of the solutions of hard problems and solving optimization problems with multiple often conflicting objectives is generally a very difficult target. The good combinations of these two major components (diversification and intensification) will usually ensure that the global optimality of given hard or complex is achievable and it provides a way solve large size population based problem instances by delivering the efficient solution in reasonable amount of time.

In short, metaheuristics are high level strategies for diversifying search spaces by using different algorithmic approach. Of great importance hereby is that a dynamic balance is given between diversification and intensification. The term diversification generally refers to the exploration of the search space, whereas the term intensification refers to the exploitation of the accumulated search experience [36].

3 Classification of Bio-inspired Metaheuristic Algorithm

Bio-inspired metaheuristic algorithms can be classified as: population based algorithm, trajectory based, swarm intelligence based, artificial intelligence based and bio-insect behavior based approaches. Some of the most famous algorithms are Genetic Algorithm (GAs), Simulated Annealing (SA) [9], Artificial Immune System (AIS) [7], Ant Colony Optimization (ACO) [6], Particle Swarm Optimization (PSO) [22] and Bacterial Foraging Algorithm (BFA) [37]. Genetic Algorithm (GAs) are enthused from Darwinian evolutionary theory [10], simulated annealing method mimics the thermodynamics process of cooling of molten metal for getting the minimum free energy state. It works with a point and at each iteration builds a point according to the Boltzmann probability distribution [9, 38]. Artificial immune systems (AIS) simulate biological immune systems [7], ant colony optimization (ACO) field study a model derived from the observation of real ant's behavior and uses these models as source of inspiration for the design of innovative and novel approaches for solution of optimization and distributed control problems. The main objective of ant colony algorithm is that the self organizing principle which allows the highly coordinated behavior of ants can be exploited to coordinate transport, Bacterial foraging algorithm (BFA) comes from search and optimal foraging of bacteria and particle swarm optimization (PSO) simulates the behavior of flock of birds and fish schooling which search for best solution in both local and global search space [5, 22, 38]. Based on bio-inspired characteristics, various algorithms are illustrated as below (Table 1).

Unlike exact algorithm methodologies (it is guaranteed to find the optimal solution and to prove its optimality for every finite size instance of a combinatorial optimization problem within an instance dependent run time.). The metaheuristic algorithms ensure to find the optimal solution of a given hard problem and reasonable amount of time. The application of metaheuristic falls into a large number of area some of them are as follows:

- Engineering design, topological optimization, structural optimizations in electronics and VLSI design, aerodynamics based structural design.
- Fluid dynamics, telecommunication field, automotives and robotics design and robotic roadmap planning optimization.
- In data mining and machine learning: Data mining in bioinformatics, computational biology.
- System modeling simulations and identification in chemistry, physics and biology.
- Images processing and control signal processing: Feature extraction from data and selection of feature with help of metaheuristic approach.
- Planning in routing based problems, robotic planning, scheduling and production based problems, logistics and transportation, supply chain management and environmental.

4 Black Hole Phenomena

In the eighteens-century, Dr. John Michel and Pierre Pierre Simon de Laplace were established to blemish the idea of black holes. Based on Newton's law, they invented the concept of a star turning into invisible to the human eye but during that period it was not able to recognize as a black hole in 1967, John Wheeler the American physicist first named the phenomenon of mass collapsing as a black hole [42]. A black hole in space is a form when a star of massive size collapses named the development of mass collapsing as apart. The gravitational power of the black hole is too strong that even the any light cannot escape from it. The gravity of such body is so

Table 1 Description of bio-inspired algorithms

S.No.	Metaheuristic algorithms	Description of metaheuristic algorithms
1.	Genetic algorithms (GAs) [10]	Genetic algorithm is a search and optimization based techniques that evolve a population of candidate solutions to a given problem, using natural genetic variation and natural selection operators
2.	Simulated annealing(SA) algorithm [9]	Simulated Annealing is developed by modeling the steel annealing process and gradually decreases the temperature (T)
3.	Ant colony optimization (ACO) [6]	Ant Colony Optimization is inspired from the behavior of a real ant colony, which is able to find the shortest path between its nest and a food source (destination)
4.	Particle swarm optimization (PSO) algorithm [22]	Particle Swarm Optimization is developed based on the swarm behavior such as fish and bird schooling in nature
5.	The gravitational search algorithm (GSA) [39]	It is constructed based on the law of gravity and the notion of mass interactions. In the GSA algorithm, the searcher agents are a collection of masses that interact with each other based on the Newtonian gravity and the laws of motion
6.	Intelligent water drops algorithm [40]	It is inspired from observing natural water drops that flow in rivers and how natural rivers find almost optimal paths to their destination. In the IWD algorithm, several artificial water drops cooperate to change their environment in such a way that the optimal path is revealed as the one with the lowest soil on its links
7.	Firefly algorithm (FA) [23, 41]	The firefly algorithm (FA) was inspired by the flashing behavior of fireflies in nature. FA is nature inspired optimization algorithm that imitates or stimulates the flash pattern and characteristics of fireflies. It is used data analysis and to identify homogeneous groups of objects based on the values of their attributes
8.	Honey bee mating optimization (HBMO) algorithm [29]	It is inspired by the process of marriage in real honey bees
9.	Bat algorithm (BA)	It is inspired by the echolocation behavior of bats. The capability of the echolocation of bats is fascinating as they can find their prey and recognize different types of insects even in complete darkness
10.	Harmony search optimization algorithm	It is inspired by the improvising process of composing a piece of music. The action of finding the harmony in music is similar to finding the optimal solution in an optimization process
11.	Big Bang–Big Crunch (Bb–By) optimization	It is based on one of the theories of the evolution of the universe. It is composed of the big bang and big crunch phases. In the big bang phase the candidate solutions are spread at random in the search space and in the big crunch phase a contraction procedure calculates a center of mass for the population

(continued)

Table 1 (continued)

S.No.	Metaheuristic algorithms	Description of metaheuristic algorithms
12.	Black hole (BH) algorithm	It is inspired by the black hole phenomenon. The basic idea of a black hole is simply a region of space that has so much mass concentrated in it that there is no way for a nearby object to escape its gravitational pull. Anything falling into a black hole, including light, is forever gone from our universe

strong because matter has been squeezed into a tiny space and anything that crosses the boundary of the black hole will be consumed or by it and vanishes and nothing can get away from its enormous power. The sphere-shaped boundary of a black hole in space is known as the event horizon. The radius of the event horizon is termed as the Schwarzschild radius. At this radius, the escape speed is equal to the speed of light, and once light passes through, even it cannot escape. Nothing can escape from within the event horizon because nothing can go faster than light. The Schwarzschild radius (R) is calculated by $R = \frac{2GM}{c^2}$, where G is the gravitational constant (6.67×10^{-11} N × $(\text{m/kg})^2$), M is the mass of the black hole, and c is the speed of light. If star moves close to the event horizon or crosses the Schwarzschild radius it will be absorbed into the black hole and permanently disappear. The existence of black holes can be discerned by its effect over the objects surrounding it [43, 44].

4.1 Black Hole

A black hole is a region of space-time (x, y, t) whose gravitational field is so strong and powerful that nothing can escape from it. The theory and principle of general relativity predicts that a sufficiently compact mass will deform space-time to form a black hole. Around a black hole, there is a mathematically defined surface called an event horizon that marks the point of no return. If anything moves close to the event horizon or crosses the Schwarzschild radius, it will be absorbed into the black hole and permanently disappear. The existence of black holes can be discerned by its effect over the objects surrounding it [45]. The hole is called *black* because it absorbs all the light that hits the horizon, reflecting nothing, just like a perfect black body in thermodynamics [46, 47]. A black hole has only three independent physical properties: Black hole's mass (M), charge (Q) and angular momentum (J). A charged black hole repels other like charges just like any other charged object in given space. The simplest black holes have mass but neither electric charge nor angular momentum [48, 49].

4.2 Black Hole Algorithm

The basic idea of a black hole is simply a region of space that has so much mass concentrated in it that there is no way for a nearby object to escape its gravitational pull. Anything falling into a black hole, including light, is forever gone from our universe.

4.2.1 Terminology of Black Hole Algorithm

Black Hole: In black hole algorithm, the best candidate among all the candidates at each iteration is selected as a black hole.

Stars: All the other candidates form the normal stars. The creation of the black hole is not random and it is one of the real candidates of the population.

Movement: Then, all the candidates are moved towards the black hole based on their current location and a random number.

1. Black hole algorithm (black hole) starts with an initial population of candidate solutions to an optimization problem and an objective function that is calculated for them.
2. At each iteration of the Black Hole, the best candidate is selected to be the black hole and the rest form the normal stars. After the initialization process, the black hole starts pulling stars around it.
3. If a star gets too close to the black hole it will be swallowed by the black hole and is gone forever. In such a case, a new star (candidate solution) is randomly generated and placed in the search space and starts a new search.

4.3 Calculation of Fitness Value for Black Hole Algorithm

1. Initial Population: $P(x) = \{x_1^t, x_2^t, x_3^t, \ldots, x_n^t\}$ randomly generated population of candidate solutions (the stars) are placed in the search space of some problem or function.
2. Find the total Fitness of population:

$$f_i = \sum_{i=1}^{pop\_size} eval(p(t) \tag{1}$$

3.

$$f_{BH} = \sum_{i=1}^{pop\_size} eval(p(t)$$

where f_i and f_{BH} are the fitness values of black hole and i_{th} star in the initialized population. The population is estimated and the best candidate (from remaining stars) in the population, which has the best fitness value, f_i is selected to be the black hole and the remaining form the normal stars. The black hole has the capability to absorb the stars that surround it. After initializing the first black hole and stars, the black hole starts absorbing the stars around it and all the stars start moving towards the black hole.

4.3.1 Absorption Rate of Stars by Black Hole

The black hole starts absorbing the stars around it and all the stars start moving towards the black hole. The absorption of stars by the black hole is formulated as follows:

$$X_i(t) = X_i(t) + rand \times (X_{BH} - X_i(t)) \tag{3}$$

where i = 1, 2, 3, …n, X_i^t and X_i^{t+1} are the locations of the *i*th star at iterations t and (t + 1) respectively. X_{BH} is the location of the black hole in the search space and rand is a random number in the interval [0, 1]. N is the number of stars (candidate solutions). While moving towards the black hole, a star may reach a location with lower cost than the black hole. In such a case, the black hole moves to the location of that star and vice versa. Then the black hole algorithm will continue with the black hole in the new location and then stars start moving towards this new location.

4.3.2 Probability of Crossing the Event Horizon During Moving Stars

In block hole algorithm, the probability of crossing the event horizon of black hole during moving stars towards the black hole is used to gather the more optimal data point from search space of the problem. Every star (candidate solution) crosses the event horizon of the black hole will be sucked by the black hole and every time a candidate (star) dies it means it sucked in by the black hole, another candidate solution (star) is populated and distributed randomly over the search space of the defined problem and go for a new search in the search solution space. It is completed to remain the number of candidate solutions constant. The next iteration takes place after all the stars have been moved. The radius of the event horizon in the black hole algorithm is calculated using the following equation: The radius of horizon (R) of black hole is demonstrated as follow:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \tag{4}$$

where f_i and f_{BH} are the fitness values of black hole and *i*th star. N is the number of stars (candidate solutions).When the distance between a candidate solution and the black hole (best candidate) is less than R, that candidate is collapsed and a new candidate is created and distributed randomly in the search space.

4.4 Pseudo Code for Black Hole Algorithm

```
1. Initialize a population of stars with random locations in the search space $P(t) = \{x_1^t, x_2^t, x_3^t \ldots x_n^t\}$. Randomly generated population of candidate solutions (the stars) are placed in the search space of some problem or function.
Loop
2. For each ith star, evaluate the objective function
       $f_i = \sum_{i=1}^{pop\_size} eval(p(t)$
       $f_{BH} = \sum_{i=1}^{pop\_size} eval(p(t)$
3. Select the best star that has the best fitness value as the black hole.
4. Change the location of each star according to Eq. (3) as
       $X_i(t) = X_i(t) + rand \times (X_{BH} - X_i(t))$
5. If a star reaches a location with lower cost than the black hole, exchange their locations.
6. If a star crosses the event horizon of the black hole
7. Calculate the event horizon radius (R)
       $R_{EventHorizon} = \frac{f_{BH}}{\sum_{i=1}^{N} f_i}$
8. When the distance between a candidate solution and the black hole (best candidate) is less than R, that candidate is collapsed and a new candidate is created and distributed randomly in the search space.
9. Replace it with a new star in a random location in the search space
10. else
    break
11. If a termination criterion (a maximum number of iterations or a sufficiently good fitness) is met exit the loop.
```

The candidate solution to the clustering problem corresponds to one dimensional (1-D) array while applying black hole algorithm for data clustering. Every candidate solution is considered as k initial cluster centers and the individual unit in the array as the cluster center dimension. Figure 1 illustrates a candidate solution of a problem with three clusters and all the data objects have four features.

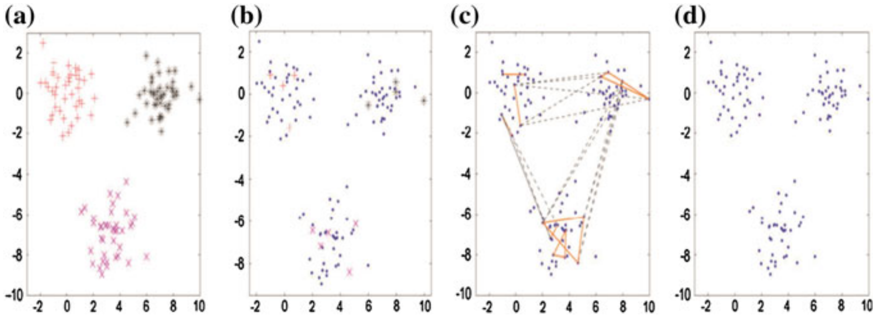


Fig. 1 Learning problems: dots correspond to points without any labels. Points with labels are denoted by *plus signs*, *asterisks*, and *crosses*. In (**c**), the must-link and cannot link constraints are denoted by *solid* and *dashed lines*, respectively [50]. **a** Supervised. **b** Partially labelled. **c** Partially constrained. d Unsupervised

4.5 Advantage of Black Hole Algorithm

- It has a simple structure and it is easy to implement.
- It is free from tuning parameter issues like genetic algorithm local search utilizes the schemata(S) theorem of higher order O(S) (compactness) and longer defining length δ(S). In Genetic Algorithm, to improve the fine tuning capabilities of a genetic algorithm, which is a must for high precision problem over the traditional representation of binary string of chromosomes? It was required a new mutation operator over the traditional mutation operator however, it only use only local knowledge i.e. it stuck into local minimum optimal value.

The Black Hole algorithm converges to global optimum in all the runs while the other heuristic algorithms may get trapped in local optimum solutions like genetic algorithm, Ant colony Optimization algorithm simulated Annealing algorithm.

5 Application of Black Hole Metaheuristic Algorithm

Nature-inspired metaheuristic algorithms have been used in many fields such as computer science [51–53] clustering analysis, industry [54] agriculture [55], computer vision [56–58] is about computing visual properties from the real world and automatic circle detection in digital still images has been considered an important and complex task for the computer vision community that has devoted a tremendous amount of research, seeking for an optimal circle detector. Electro-magnetism Optimization (EMO) bio-inspired algorithm based circle detector method which assumes the overall detection process as optimization problem, forecasting [59], medicine and biology [60], scheduling [61], data mining [62, 63], economy [64] and engineering [65]. There are following applications of black hole algorithm in data clustering and its performance analysis.

5.1 Cluster Analysis

Clustering is an important unsupervised classification approach, where a set of patterns are usually vectors (observations, data items, or feature vectors) into in multi-dimensional space are grouped into clusters or groups, based on some similarity metrics between data objects; the distance measurement is used to find out the similarity and dissimilarity of different object of our database [66]. The main idea is to classify a given data set through a certain number of clusters by minimizing distances between objects inside each cluster. Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity [50, 67]. Cluster is often used for different type of application in image processing, data statistical analysis, medical imaging analysis and other field of science and engineering research field. In addition, it is a main task of exploratory data mining and a common technique for statistical data analysis used in many fields including machine learning, pattern recognition, image analysis, information retrieval and bioinformatics.

5.1.1 Data Analysis Problem Is Specified as Follows

Given N objects, assign each object to one of K clusters and minimize the sum of squared euclidean distances between each object and the center of the cluster that belongs to every allocated object:

$$F(O,Z) = \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{K} Wij(O_i - Z_j)^2$$

where $(O_i - Z_j)^2 = \|\mathrm{O_i} - \mathrm{Z_j}\|$ is the Euclidean distance between a data object O_i and the cluster center Z_j. N and K are the number of data objects and the number of clusters, respectively. W_{ij} is the association weight of data object $O_i$with cluster j.

$$W_{ij} = \begin{cases} 1 & \text{if object i is assign to cluster j.} \\ 0 & \text{if object i is not assigned to cluster j.} \end{cases}$$

5.2 Data Clustering

The goal of data clustering also known as cluster analysis is to discover the natural grouping of a set of patterns, points or objects. An operational definition of clustering can be stated as follows: Given a representation of n objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are

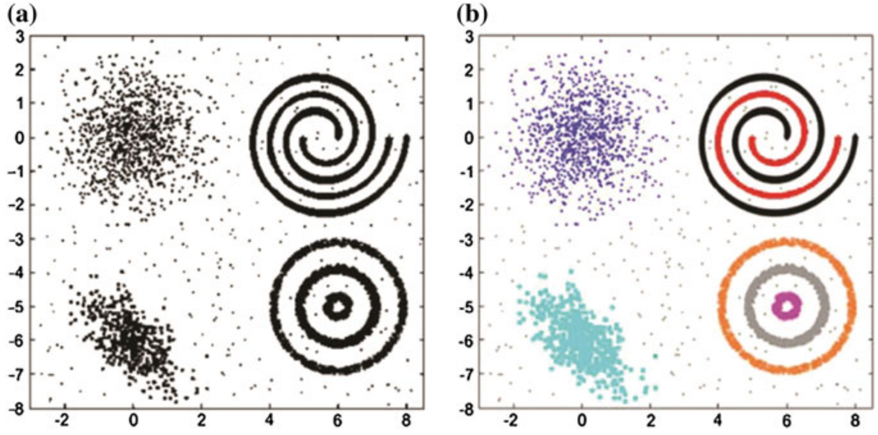


Fig. 2 Diversity of clusters. The seven clusters in (**a**) [denoted by seven different colors in 1(**b**)] differ in *shape*, *size* and *density*. Although these clusters are apparent to a data analyst, none of the available clustering algorithms can detect all these clusters. (*Source* [50]. **A** Input data. **b** Desired clustering

low. Figure 2 demonstrates that clusters may differ in terms of their shape, size, and density. The presence of noise in the data makes the detection of the clusters even more difficult and ideal cluster can be defined as a set of points that is compact and isolated. While humans are excellent cluster seekers in two and possibly three dimensions, we need automatic algorithms for high-dimensional data. It is this challenge along with the unknown number of clusters for the given data that has resulted in thousands of clustering algorithms that have been published and that continue to appear. An example of clustering is shown in Fig. 2. In pattern recognition, data analysis is concerned with predictive modeling: given some training data and to predict the behavior of the unseen test data. This task is also referred to as learning.

5.2.1 Classification of Machine Learning

Classification of data based on machine algorithms is follow:

Supervised Learning and Unsupervised Learning

Supervised learning is the machine learning approach of inferring a function from labeled training data. The training data consist of a set of training examples. Let a set of labeled training data $X = [(x_n, y_n)] \in X \times Y \leq n \leq N$ where x is input space and y a finite label set. It is assumed that each $[(x_n, y_n)]$ is drawn independently from a fixed, but unknown probability distribution p, where $[(x_n, y_n)] \in p(x, y)$. Unfortunately, supervised learning method is very expensive and time consuming to collect a huge amount of labeled data $[(x_n, y_n)]$. One of learning approach to deal such issues is to exploit unsupervised learning. The main aim object is to learn a classification model from both labeled $X = [(x_n, y_n)] \in X \times Y \le n \le N$ and $[x_j]_{j=N+1}^{N+M}$ unlabelled data where $N \le M$. Clustering is a more difficult and challenging problem than classification.

Semi-supervised Learning

In semi-supervised classification, the labels of only a small portion of the training data set are available. The unlabeled data, instead of being discarded, are also used in the learning process. In semi-supervised clustering, instead of specifying the class labels, pair-wise constraints are specified, which is a weaker way of encoding the prior knowledge. A pair-wise must-link constraint corresponds to the requirement that two objects should be assigned the same cluster label, whereas the cluster labels of two objects participating in a cannot-link constraint should be different [50, 68]. Constraints can be particularly beneficial in data clustering, where precise definitions of underlying clusters are absent. Figure 1 illustrates this spectrum of different types of learning problems of interest in pattern recognition and machine learning.

5.3 K-means Clustering

Cluster analysis is prevalent in any discipline that involves analysis of multivariate data. Clustering algorithm K-means was first published in 1955. It is difficult to exhaustively list the numerous scientific fields and applications that have utilized clustering techniques as well as the thousands of published algorithms. Image segmentation an important problem in computer vision, can be formulated as a clustering problem. Documents can be clustered to generate topical hierarchies for efficient information access. Clustering is also used to group customers into different types for efficient marketing to group services delivery engagements for workforce management and planning as well as to study genome data in biology [50]. Data clustering has been used for the following three main purposes.

- Underlying structure to gain insight into data generates hypotheses, detect anomalies, and identify salient features.
- Natural classification to identify the degree of similarity among forms or organisms (phylogenetic relationship).
- Compression: as a method for organizing the data and summarizing it through cluster prototypes.

Among the classical clustering algorithms, K-means is the most well known algorithm due to its simplicity and efficiency.

5.3.1 Classification of Clustering Algorithms

Clustering algorithms are classified into can be broadly divided into two categories: (1): Hierarchical clustering and Partitional clustering.

Hierarchical Algorithm

Hierarchical clustering constructs a hierarchy of groups by splitting a large cluster into small ones and merging smaller cluster into a large cluster centroid [69]. In this, there are two main approaches:(1) *the divisive approach*, which splits a large cluster into two or more smaller clusters; (2) *the agglomerative approach,* which builds a larger cluster by merging two or more smaller clusters by recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy. Input to a hierarchical algorithm is an n × n similarity matrix, where n is the number of objects to be clustered [50].

Partitioned Algorithm

Partitioned clustering algorithms find all the clusters simultaneously as a partition of the data without hierarchical structure. The most widely used partitional clustering approaches are prototype-based clustering algorithm where each cluster is demonstrated by its centre. The objective function or square error function is sum of distance from the pattern to the centre [70]. Partitioned algorithm can use either an n × d pattern matrix, where n objects are embedded in a d-dimensional feature space, or an n × n similarity matrix. Note that a similarity matrix can be easily derived from a pattern matrix but ordination methods such as multi-dimensional scaling (MDS) are needed to derive a pattern matrix from a similarity matrix. The most well-known hierarchical algorithms are single-link and complete-link. The most popular and the simplest partitioned algorithm is K-means. Since partitioned algorithms are preferred in pattern recognition due to the nature of available data, our coverage here is focused on these algorithms [50].

5.4 K-means Algorithm

K-means has a rich and diverse history as it was independently discovered in different scientific fields by [71], Lloyd. It is a popular partitional clustering algorithm and essentially a function minimization technique, where the main objective function is the square error.

Let $\mathrm{X} = \{x_i\}, i = 1, 2, 3, 4\ldots, n$ be the set of n d-dimensional points to be clustered into a set of K clusters, $C = \{c_k\}$ where k = 1, 2, 3, 4….K. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of cluster C_k. The squared error between μ_k and the points in cluster C_k is defined as $J(c_k) = \sum_{x_i \in c_k} (X_i - \mu_k)^2$. The goal of K-means is to minimize the sum of the squared error over all K clusters, $J(C) = \sum_{k=1}^{K} \sum_{xi \in c_k}^{N} (X_i - \mu_k)^2$. Minimizing this objective function is known to be an NP-hard problem (even for K = 2). Thus K-means which is a greedy algorithm and can only converge to a local minimum, even though recent study has shown with a large probability, K-means could converge to the global optimum when clusters are well separated. K-means starts with an initial partition with K clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error always decreases with an increase in the number of clusters K (with J(C) = 0 when K = n), it can be minimized only for a fixed number of clusters [64]. Recently efficient hybrid evolutionary and bio-inspired metaheuristic methods and K-means to overcome local problems in clustering are used [72, 73] (Niknam and Amiri 2010).

5.4.1 Steps of K-Means Algorithm

1. Select an initial partition with K clusters.
2. Repeat steps 2 and 3 until cluster membership stabilizes.
3. Generate a new partition by assigning each pattern to its closest cluster center.
4. Compute new cluster centers (Figs. 3, 4).

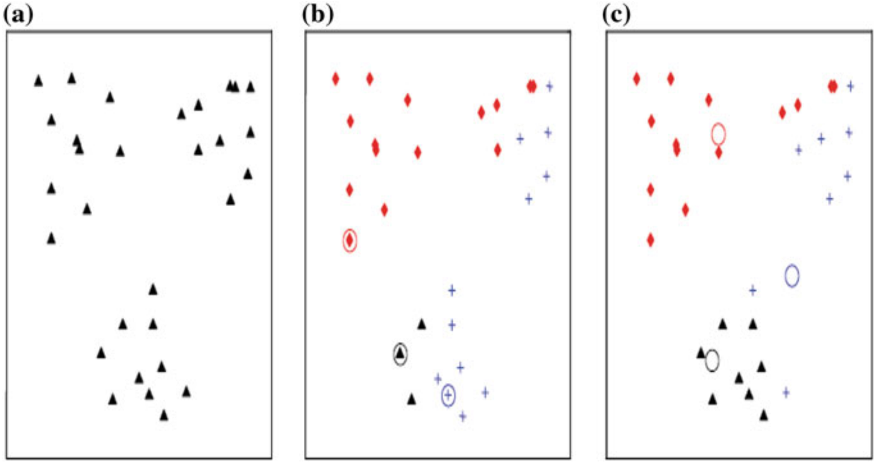


Fig. 3 Illustration of K-means algorithm **a** Two-dimensional input data with three clusters; **b** three seed points selected as cluster centers and initial assignment of the data points to clusters; and **c** updates intermediate cluster labels

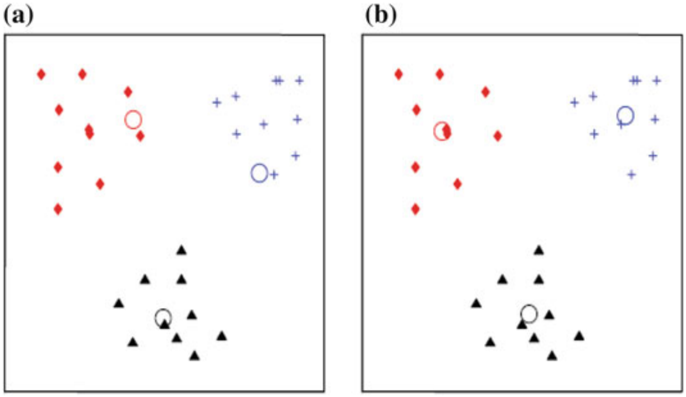


Fig. 4 **a** and **b** intermediate iterations updating cluster labels and their centers an final clustering obtained by K-means

The K-means algorithm requires three user-specified parameters: number of clusters K, cluster initialization, and distance metric. The most critical choice is K. While no perfect mathematical criterion exists, a number of heuristics are available for choosing K. Typically, K-means is run independently for different values of K and the partition that appears the most meaningful to the domain expert is selected. Different initializations can lead to different final clustering because K-means. One way to overcome the local minima is to run the K-means algorithm, for a given K, with multiple different initial partitions and choose the partition with the smallest squared error. K-means is typically used with the Euclidean metric for computing the distance between points and cluster centers. As a result, K-means finds spherical or ball-shaped clusters in data. K-means with mahalanobis distance metric has been used to detect hyper ellipsoidal clusters [74].

5.5 Advantages and Disadvantages of the K-Means Clustering

Among the all classical clustering algorithms, K-means clustering algorithm is the well known algorithm due to their simplicity and efficiency. It suffers from two problems: It needs number of cluster before starting i.e. the number of cluster must be known a priori. In addition, its performance strongly depends on the initial centroids and may get stuck in local optima solutions and its convergence rate are affected [56]. In order to overcome the shortcomings of K-means many heuristic approaches have been applied in the last two decades.

5.6 *Evolutionary Computation Algorithms for Cryptography and Cryptanalysis*

Cryptography is a methodology and study of techniques for secure communication in the presence of third parties and cryptanalysis is the study of analyzing information systems in order to study the hidden aspects of the systems. The cryptanalysis of different cipher problems can be formulated as NP-hard combinatorial problem. Metaheuristic algorithm provides a very powerful tool for the cryptanalysis of simple substitution ciphers using a cipher text only attack and automated cryptanalysis of classical simple substitution ciphers [75–77].

Recently, efficient hybrid evolutionary optimization algorithms based on combining evolutionary methods and swarm intelligence has significant role in the field of cryptography and demonstrates a dynamical system which is sensitive to initial condition and generates apparently random behavior but at the same time the system is completely deterministic. Hussein et al. presents a new encryption scheme based on a new chaotic map derived from a simplified model of Swarm Intelligence (SI) [78] and overcome the problem of cryptograph [79–81].

5.7 *Short-Term Scheduling Based System Optimization by Black Hole Algorithm*

Recently, a major challengeable subject that are facing the electric power system operator and how to manage optimally the power generating units over a scheduling horizon of one day considering all of the practical equality inequality and dynamic constraints. These constraints of system are comprised of load plus transmission losses balance, valve-point effects, prohibited operating zones, multi-fuel options, line flow constraints, operating reserve and minimum on/off time. There is not available any optimization for the short-term thermal generation scheduling (STGS). It has high-dimensional, high-constraints, non-convex, non-smooth and non-linear nature and needs an efficient algorithm to be solved. Then, a new optimization approach, known as gradient-based modified teaching–learning-based optimization combined with black hole (MTLBO–BH) algorithm has been planned to seek the optimum operational cost [82–84].

6 Discussion

The field of metaheuristic approaches for the application to combinatorial optimization problems is a rapidly growing field of research. This is due to a great importance of combinatorial optimization problems for the scientific as well as the industrial world. Since the last decade metaheuristics approaches have a significant role for solving the complex problem of different applications in science, computer vision, computer science, data analysis, data clustering, and mining, clustering analysis, industrial forecasting of weather, medical and biological research, economy and different multi-disciplinary engineering research field. In addition, metaheuristics are useful in computer vision, image processing, machine learning and pattern recognition of any subject which can be deployed for finding the optimal set of discriminant values in form of Eigen vector (face recognition, fingerprint and other biometric characteristics) and incorporate these values for identification purpose. In biometrics and computer vision, face recognition has always been a major challenge for machine learning researchers and pattern recognition. Introducing the intelligence in machines to identifying humans from their face images (which is stored in template data base) deals with handling variations due to illumination condition, pose, facial expression, scale and disguise etc., and hence becomes a complex task in computer vision. Face recognition demonstrates a classification problem for human recognition. Face recognition classification problems can be solved by a technique for the design of the *Radial Basis Functions* neural network with metaheuristic approaches (like firefly, particle swarm intelligence and black hole algorithm). These algorithms can be used at match score level in biometrics and select most discriminant set of optimal features for identification of face and their classification.

Recently black hole methodology plays a major role of modeling and simulating natural phenomena for solving complex problems. The motivation for new heuristic optimization algorithm is based on the black hole phenomenon. Further, it has a simple structure and it is easy to implement and it is free from parameter tuning issues like genetic algorithm. The black hole algorithm can be applied to solve the clustering problem and can run on different benchmark datasets. In future research, the proposed algorithm can also be utilized for many different areas of applications. In addition, the application of BH in combination with other algorithms may be effective. Meta-heuristics support managers in decision-making with robust tools that provide high-quality solutions to important applications in business, engineering, economics and science in reasonable time horizons.

7 Conclusion and Future Direction

We conclude that new black hole algorithm approach is population based same as particle swarm optimization, firefly, genetic algorithm, BAT algorithm and other evolutionary methods. It is free from parameter tuning issues like genetic algorithm and other. It does not suffer from premature convergence problem. This implies that black hole is potentially more powerful in solving NP-hard (e.g. data clustering problem) problems which is to be investigated further in future studies. The further improvement on the convergence of the algorithm is to vary the randomization parameter so that it decreases gradually as the optima are approaching. In wireless sensor network, density of deployment, scale, and constraints in battery, storage device, bandwidth and computational resources create serious challenges to the developers of WSNs. The main issues of the node deployment, coverage and mobility are often formulated as optimization problems and moth optimization techniques suffer from slow or weak convergence to the optimal solutions for high performance optimization methods that produce high quality solutions by using minimum resources. Bio-inspired black hole algorithm can give a model to solve optimization problems in WSNs due to its simplicity, best solution, fast convergence and minimum computational complexity. These can be form important topics for further research in computer network. Furthermore, as a relatively straight forward extension, the black hole algorithm can be modified to solve multi objective optimization problems. In addition, the application of black hole in combination with other algorithms may form an exciting area for further research.

References

1. Tan, X., Bhanu, B.: Fingerprint matching by genetic algorithms. Pattern Recogn. **39**, 465–477 (2006)
2. Karakuzu, C.: Fuzzy controller training using particle swarm optimization for nonlinear system control. ISA Trans. **47**(2), 229–239 (2008)
3. Rajabioun, R.: Cuckoo optimization algorithm. Elsevier Appl. Soft Comput. **11**, 5508–5518 (2011)
4. Tsai Hsing, C., Lin, Yong-H: Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. Appl. Soft Comput. Elsevier **1**, 5367–5374 (2011)
5. Baojiang, Z., Shiyong, L.: Ant colony optimization algorithm and its application to neu ro-fuzzy controller design. J. Syst. Eng. Electron. **18**, 603–610 (2007)
6. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B **26**(1), 29–41 (1996)
7. Farmer, J.D., et al.: The immune system, adaptation and machine learning. Phys. D Nonlinear Phenom. Elsevier **22**(1–3), 187–204 (1986)
8. Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. Inf. Sci. **177**, 3918–3937 (2007)
9. Kirkpatrick, S., Gelatto, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
10. Tang, K.S., Man, K.F., Kwong, S., He, Q.: Genetic algorithms and their applications. IEEE Sig. Process. Mag. **3**(6), 22–37 (1996)
11. Du, Weilin, Li, B.: Multi-strategy ensemble particle swarm optimization for dynamic optimization. Inf. Sci. **178**, 3096–3109 (2008)
12. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**, 82–102 (1999)
13. Liu, Y., Yi, Z., Wu, H., Ye, M., Chen, K.: A tabu search approach for the minimum sum-of-squares clustering problem. Inf. Sci. **178**(12), 2680–2704 (2008)
14. Kim, T.H., Maruta, I., Sugie, T.: Robust PID controller tuning based on the constrained particle swarm optimization. J. Autom. Sciencedirect **44**(4), 1104–1110 (2008)
15. Cordon, O., Santamarı, S., Damas, J.: A fast and accurate approach for 3D image registration using the scatter search evolutionary algorithm. Pattern Recogn. Lett. **27**, 1191–1200 (2006)
16. Yang, X.S.: Firefly algorithms for multimodal optimization, In: Proceeding of Stochastic Algorithms: Foundations and Applications (SAGA), 2009 (2009)

17. Kalinlia, A., Karabogab, N.: Artificial immune algorithm for IIR filter design. Eng. Appl. Artif. Intell. **18**, 919–929 (2005)
18. Lin, Y.L., Chang, W.D., Hsieh, J.G.: A particle swarm optimization approach to nonlinear rational filter modeling. Expert Syst. Appl. **34**, 1194–1199 (2008)
19. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
20. Jackson, D.E., Ratnieks, F.L.W.: Communication in ants. Curr. Biol. **16**, R570–R574 (2006)
21. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the Argentine ant. Naturwissenschaften **76**, 579–581 (1989)
22. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks **4**, 1942–1948 (1995)
23. Yang, X. S.: 2010, ‘Nature-inspired metaheuristic algorithms’, Luniver Press
24. Tarasewich, p, McMullen, P.R.: Swarm intelligence: power in numbers. Commun. ACM **45**, 62–67 (2002)
25. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study. Swarm Evol. Comput. **1**(3), 164–171 (2011)
26. Yang, X.S.: Firefly algorithm. Engineering Optimization, pp. 221–230 (2010)
27. Yang, X.S.: Bat algorithm for multi-objective optimization. Int. J. Bio-inspired Comput. **3**(5), 267–274 (2011)
28. Tripathi, P.K., Bandyopadhyay, S., Pal, S.K.: Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf. Sci. **177**, 5033–5049 (2007)
29. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report TR06,Erciyes University (2005)
30. Ellabib, I., Calamari, P., Basir, O.: Exchange strategies for multiple ant colony system. Inf. Sci. **177**, 1248–1264 (2007)
31. Hamzaçebi, C.: Improving genetic algorithms performance by local search for continuous function optimization. Appl. Math. Comput. **96**(1), 309–317 (2008)
32. Lozano, M., Herrera, F., Cano, J.R.: Replacement strategies to preserve useful diversity in steady-state genetic algorithms. Inf. Sci. **178**, 4421–4433 (2008)
33. Lazar, A.: Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets, *Heuristic and Optimization for Knowledge Discovery*, IGI Global, pp. 263–278 (2014)
34. Russell, S.J., Norvig, P.: Artificial Intelligence a Modern Approach. Prentice Hall, Upper Saddle River (2010). 1132
35. Fred, W.: Glover, Manuel Laguna, Tabu Search, 1997, ISBN: 079239965X
36. Christian, B., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surveys (CSUR) **35**(3), 268–308 (2003)
37. Gazi, V., Passino, K.M.: Stability analysis of social foraging swarms. IEEE Trans. Syst. Man Cybern. Part B **34**(1), 539–557 (2008)
38. Deb, K.: Optimization for Engineering Design: Algorithms and Examples, Computer-Aided Design. PHI Learning Pvt. Ltd., New Delhi (2009)
39. Rashedi, E.: Gravitational Search Algorithm. M.Sc. Thesis, Shahid Bahonar University of Kerman, Kerman (2007)
40. Shah-Hosseini, H.: The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. Int. J. Bio-inspired Comput. **1**(1), 71–79 (2009)
41. Dos Santos, C.L., et al.: A multiobjective firefly approach using beta probability. IEE Trans. Magn. **49**(5), 2085–2088 (2013)
42. Talbi, E.G.: Metaheuristics: from design to implementation, vol. 74, p. 500. Wiley, London (2009)
43. Giacconi, R., Kaper, L., Heuvel, E., Woudt, P.: Black hole research past and future. In: Black Holes in Binaries and Galactic Nuclei: Diagnostics. Demography and Formation, pp. 3–15. Springer, Berlin, Heidelberg (2001)
44. Pickover, C.: Black Holes: A Traveler’s Guide. Wiley, London (1998)
45. Frolov, V.P., Novikov, I.D.: Phys. Rev. D. **42**, 1057 (1990)

46. Schutz, B. F.: Gravity from the Ground Up. Cambridge University Press, Cambridge. ISBN 0-521-45506-5 (2003)
47. Davies, P.C.W.: Thermodynamics of Black Holes. Reports on Progress in Physics, Rep. Prog. Phys. vol. 41 Printed in Great Britain (1978)
48. Heusler, M.: Stationary black holes: uniqueness and beyond. Living Rev. Relativity **1**(1998), 6 (1998)
49. Nemati, M., Momeni, H., Bazrkar, N.: Binary black holes algorithm. Int. J. Comput. Appl. **79** (6), 36–42 (2013)
50. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
51. Akay, B., Karaboga, D.: A modified artificial bee colony algorithm for real-parameter optimization. Inf. Sci. **192**, 120–142 (2012)
52. El-Abd, M.: Performance assessment of foraging algorithms vs. evolutionary algorithms. Inf. Sci. **182**, 243–263 (2012)
53. Ghosh, S., Das, S., Roy, S., Islam, M.S.K., Suganthan, P.N.: A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization. Inf. Sci. **182**, 199–219 (2012)
54. Fox, B., Xiang, W., Lee, H.: Industrial applications of the ant colony optimization algorithm. Int. J. Adv. Manuf. Technol. **31**, 805–814 (2007)
55. Geem, Z., Cisty, M.: Application of the harmony search optimization in irrigation. Recent Advances in Harmony Search Algorithm', pp. 123–134. Springer, Berlin (2010)
56. Selim, S.Z., Ismail, M.A.: K-means-type algorithms: a generalized convergence theorem and characterization of local optimality pattern analysis and machine intelligence. IEEE Trans. PAMI **6**, 81–87 (1984)
57. Wang, J., Peng, H., Shi, P.: An optimal image watermarking approach based on a multi-objective genetic algorithm. Inf. Sci. **181**, 5501–5514 (2011)
58. Picard, D., Revel, A., Cord, M.: An application of swarm intelligence to distributed image retrieval. Inf. Sci. **192**, 71–81 (2012)
59. Chaturvedi, D.: Applications of genetic algorithms to load forecasting problem. Springer, Berlin, pp. 383–402 (2008) (Journal of Soft Computing)
60. Christmas, J., Keedwell, E., Frayling, T.M., Perry, J.R.B.: Ant colony optimization to identify genetic variant association with type 2 diabetes. Inf. Sci. **181**, 1609–1622 (2011)
61. Guo, Y.W., Li, W.D., Mileham, A.R., Owen, G.W.: Applications of particle swarm optimization in integrated process planning and scheduling. Robot. Comput.-Integr. Manuf. Elsevier **25**(2), 280–288 (2009)
62. Rana, S., Jasola, S., Kumar, R.: A review on particle swarm optimization algorithms and their applications to data clustering. Artif. Intell. Rev. **35**, 211–222 (2011)
63. Yeh, W.C.: Novel swarm optimization for mining classification rules on thyroid gland data. Inf. Sci. **197**, 65–76 (2012)
64. Zhang, Y., Gong, D.W., Ding, Z.: A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. Inf. Sci. **192**, 213–227 (2012)
65. Marinakis, Y., Marinaki, M., Dounias, G.: Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. Inf. Sci. **181**, 4684–4698 (2011)
66. Anderberg, M.R.: Cluster analysis for application. Academic Press, New York (1973)
67. Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)
68. Valizadegan, H., Jin, R., Jain, A.K.: Semi-supervised boosting for multi-class classification. 19th European Conference on Machine Learning (ECM), pp. 15–19 (2008)
69. Chris, D., Xiaofeng, He: Cluster merging and splitting in hierarchical clustering algorithms. Proc. IEEE ICDM **2002**, 1–8 (2002)
70. Leung, Y., Zhang, J., Xu, Z.: Clustering by scale-space filtering. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1396–1410 (2000)
71. Révész, P.: On a problem of Steinhaus. Acta Math. Acad. Scientiarum Hung. **16**(3–4), 311–331(1965)

72. Niknam, T., et al.: An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering. J. Zhejiang Univ. Sci. A **10**(4), 512–519 (2009)
73. Niknam, T., Amiri, B.: An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Appl. Soft Comput. **10**(1), 183–197 (2011)
74. Ding, C., He, X.: K-means clustering via principal component analysis. Proceedings of the 21th international conference on Machine learning, pp. 29 (2004)
75. Uddin, M.F., Youssef, A.M.: Cryptanalysis of simple substitution ciphers using particle swarm optimization. IEEE Congress on Evolutionary Computation, pp. 677–680 (2006)
76. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**, 58–73 (2002)
77. Danziger, M., Amaral Henriques, M.A.: Computational intelligence applied on cryptology: a brief review. Latin America Transactions IEEE (Revista IEEE America Latina) **10**(3), 1798–1810 (2012)
78. Chee, Y., Xu, D.: Chaotic encryption using discrete-time synchronous chaos. Phys. Lett. A **348**(3–6), 284–292 (2006)
79. Hussein, R.M., Ahmed, H.S., El-Wahed, W.: New encryption schema based on swarm intelligence chaotic map. Proceedings of 7th International Conference on Informatics and Systems (INFOS), pp. 1–7 (2010)
80. Chen, G., Mao, Y.: A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solutions Fractals **21**, 749–761 (2004)
81. Hongbo, Liu: Chaotic dynamic characteristics in swarm intelligence. Appl. Soft Comput. **7**, 1019–1026 (2007)
82. Azizipanah-Abarghooeea, R., et al.: Short-term scheduling of thermal power systems using hybrid gradient based modified teaching–learning optimizer with black hole algorithm. Electric Power Syst. Res. Elsevier **108**, 16–34 (2014)
83. Bard, J.F.: Short-term scheduling of thermal-electric generators using Lagrangian relaxation. Oper. Res. **36**(5), 756–766 (1988)
84. Yu, I.K., Song, Y.H.: A novel short-term generation scheduling technique of thermal units using ant colony search algorithms. Int. J. Electr. Power Energy Syst. **23**, 471–479 (2001)

Genetic Algorithm Based Multiobjective Bilevel Programming for Optimal Real and Reactive Power Dispatch Under Uncertainty

Papun Biswas

Abstract This chapter presents how multiobjective bilevel programming (MOBLP) in a hierarchical structure can be efficiently used for modeling and solving optimal power generation and dispatch problems via genetic algorithm (GA) based Fuzzy Goal Programming (FGP) method in a power system operation and planning horizon. In MOBLP formulation of the proposed problem, first the objectives of real and reactive power (P-Q) optimization are considered as two optimization problems at two individual levels (top level and bottom level) with the control of more than one objective at each level. Then the hierarchically ordered problem is fuzzily described to accommodate the impression in P-Q optimization simultaneously in the decision making context. In the model formulation, the concept of membership functions in fuzzy sets for measuring the achievement of highest membership value (unity) of the defined fuzzy goals to the extent possible by minimising their under-deviational variables on the basis of their weights of importance is considered. The aspects of FGP are used to incorporate the various uncertainties in power generation and dispatch. In the solution process, the GA is used in the framework of FGP model in an iterative manner to reach a satisfactory decision on the basis of needs and desires of the decision making environment. The GA scheme is employed at two different stages. At the first stage, individual optimal decisions of the objectives are determined for fuzzy goal description of them. At the second stage, evaluation of goal achievement function for minimization of the weighted under-deviational variables of the membership goals associated with the defined fuzzy goals is considered for achieving the highest membership value (unity) of the defined fuzzy goals on the basis of hierarchical order of optimizing them in the decision situation. The proposed approach is tested on the standard IEEE 6-Generator 30-Bus System to illustrate the potential use of the approach.

P. Biswas (✉)
Department of Electrical Engineering, JIS College of Engineering,
West Bengal University of Technology, West Bengal, India
e-mail: papun.biswas.2008@ieee.org

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_8

Keywords Multiobjective bilevel programming · Fuzzy goal programming · Genetic algorithm · Real and reactive power (P-Q) optimization · Voltage deviation · Transmission-losses

1 Introduction

The thermal power operation and management problems [36] are actually optimization problems with multiplicity of objectives and various system constraints. The most important optimization problem in power system operation and planning is real power optimization i.e. economic-dispatch problem (EDP). It is to be noted that more than 75 % of the power plants throughout the world are thermal plants, where fossil fuel is used as source for power generation. In most of the thermal power plant coal is used as main electric power generation source. But, generation of electric power by burning coal leads to produce various harmful pollutants like oxides of carbon, oxides of nitrogen and oxides of sulphur. These byproducts not only affect the human but also the entire living beings in this world. So, economic-dispatch problem of electric power plant is actually a combined optimization problem where real-power generation cost and environmental emission from the plant during the generating of power has to optimize simultaneously when several operational constraints must be satisfied.

Actually the practical loads in electrical system may have resistance, inductance and capacitance or their combinations. Most of the loads in electrical power system are reactive (inductive or capacitive) in nature. Due to the presence of reactive loads, the reactive power will also always be present in the system. Figure 1 represents the voltage and current waveform in inductive load.

Figure 2 diagrammatically represents the real (P) and reactive (Q) power. The vector sum of real and reactive power is known as apparent power (S). Electrical system designers have to calculate the various parameters based on the apparent power which is greater than the real power.

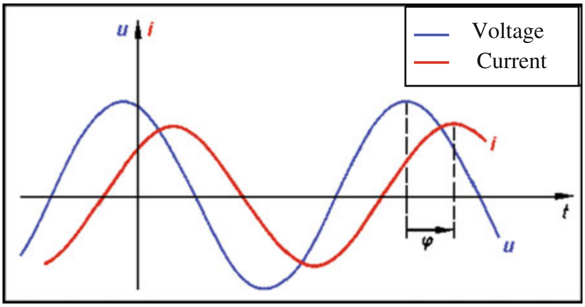


Fig. 1 Voltage and current waveform in inductive load

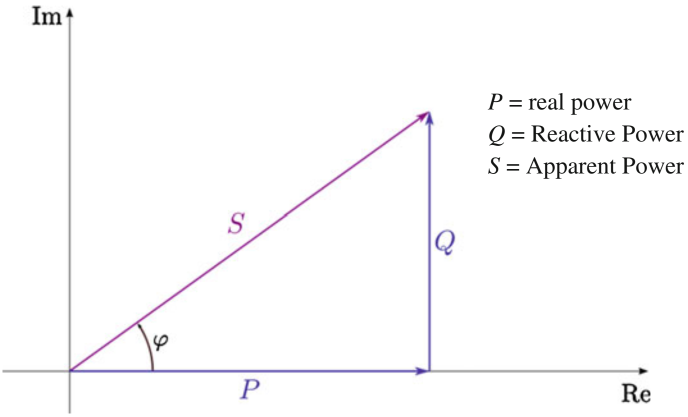


Fig. 2 Power triangle

Now, reactive power is very essential for flowing of active power in the circuit. So, reactive power dispatch (RPD) is also a very important objective in power system operation and planning. The main objective of reactive power planning is to maintain a proper voltage profile and satisfaction of operational constrains in all the load buses.

Reactive power, termed as *Volt-Amps-Reactive* (VAR), optimization is one of the major issues of modern energy management system. Reactive power optimization also has significant influence on economic operations of power systems.

The purpose of optimal reactive power dispatch (ORPD) is to minimize the real power loss of the system and improve voltage profiles by satisfying load demand and operational constraints. In power systems, ORPD problem is actually an optimization problem with multiplicity of objectives. Here the reactive power dispatch problem involves best utilization of the existing generator bus voltage magnitudes, transformer tap setting and the output of reactive power sources so as to optimize the loss and to enhance the voltage stability of the system. Generally, the load bus voltages can be maintained within their permissible limits by adjusting transformer taps, generator voltages, and switchable VAR sources. Also, the system losses can be minimized via redistribution of reactive power in the system.

Now, in power system operation and planning the simultaneous combined optimization of the above two problems (EDP and RPD) is very essential for proper planning of the system design and operation.

In most of the practical optimization the decision parameters of problems with multiplicity of objectives are inexact in nature. This inexactness is due to the inherent impressions in parameter themselves as well as imprecise in nature of human judgments during the setting of the parameter values.

To cope with the above situations and to overcome the shortcomings of the classical approaches, the concept of membership functions in fuzzy sets theory (FST) [69] has appeared as a robust tool for solving the optimization problems.

Also the concept of fuzzy logic [5, 6], deals with approximate reasoning rather than fixed and exact, has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false.

In this situation, Fuzzy programming (FP) approach [73] based on FST can be applied for achieving the solution of the real-life optimization problem. The conventional FP approaches discussed by Zimmerman [74] have been further extended by Shih and Lee [57], Sakawa and Nishizaki [55, 56], and others to solve hierarchical decision problems from the view point of making a balance of decision powers in the decision making environment.

But, the main drawback of conventional FP approach is that there is a possibility of rejecting the solution repeatedly due to dissatisfaction with the solution. Due to this repeated rejection and further calculation the computational load and computing time increases and the decision is frequently found not closer to the highest membership value (unity) to reach the optimal solution.

To overcome the above difficulty in solving FP problems with multiplicity of objectives, fuzzy goal programming (FGP) as an extension of conventional goal programming (GP) [35, 52] which is based on the goal satisficing philosophy exposed by Simon [58], has been introduced by Pal and Moitra [47] to the field of conventional FP for making decision with regard to achieving multiple fuzzy goals efficiently in the decision making environment.

In FGP approach, achievement of fuzzy goals to their aspired levels in terms of achieving the highest membership value (unity) of each of them is considered. In FGP model formulation, the membership functions of the defined fuzzy goals are transformed into flexible membership goals by assigning the highest membership value (unity) as the aspiration levels and introducing under- and over-deviational variables to each of them in an analogous to conventional GP approach. Here, in the goal achievement function, only the under-deviational variables of the membership goals are minimized, which can easily be realized from the characteristics of membership functions defined for the fuzzy goals.

Now, in the context of solving the optimization problems, although conventional multiobjective decision making (MODM) methods have been studied extensively in the past, it would have to be realized to the fact that the objectives of the proposed problem are inherently incommensurable owing to the two opposite interests of economic power generation and emission reduction for protection of environment from pollution, because emission amount depends on quality of fossil-fuel used in thermal plants.

To overcome the above situation, bilevel programming (BLP) formulation of the problem in a hierarchical decision [37] structure can be reasonably taken into account to make an appropriate decision in the decision making environment.

The concept of hierarchical decision problem was introduced by Burton [10] for solving decentralized planning problems. The concept of BLP was first introduced by Candler and Townsley [12]. Bilevel programming problem (BLPP) is a special case of Multilevel programming problem (MLPP) of a hierarchical decision system. In a BLPP, two decision makers (DMs) are located at two different hierarchical levels, each independently controlling one set of decision variables and with different and perhaps conflicting objectives.

In a hierarchical decision process, the lower-level DM (the follower) executes his/her decision powers, after the decision of the higher-level DM (the leader). Although the leader independently optimizes its own benefits, the decision may be affected by the reactions of the follower. As a consequence, decision deadlock arises frequently and the problem of distribution of proper decision power is encountered in most of the practical decision situations. In MOBLP, there will be more than one objective in both the hierarchical levels.

Further, to overcome the computational difficulty with nonlinear and competitive in nature of objectives, genetic algorithms (GAs) [25] based on natural selection and natural genetics, initially introduced by Holland [31, 32], have appeared as global solution search tools to solve complex real-world problems. The deep study on GA based solution methods made in the past century has been well documented by Michalewicz [44] and Deb [16] among others. The GA based solution methods [23, 72] to fuzzy multiobjective optimization problems have been well documented by Sakawa [54].

The GA based solution approach to BLPPs in crisp decision environment was first studied by Mathieu et al. [43]. Thereafter, the computational aspects of using GAs to fuzzily described hierarchal decision problems have been investigated [29, 46, 54, 55] in the past. The potential use of GAs to quadratic BLPP model has also been studied by Pal and Chakraborti [49] in the recent past.

2 Related Works

Demand of electric power has increased in an alarming way in the recent years owing to rapid growth of human development index across the countries in modern world. Here it is to be mentioned that the main source of electric energy supply is thermal power plants, where fossil-fuel is used as resource for power generation. The thermal power system planning and operation problems are actually optimization problems with various system constraints in the environment of power generation and dispatch on the basis of needs in society.

The general mathematical programming model for optimal power generation was introduced by Dommel and Tinney [19]. The first mathematical model for optimal control of reactive power flow was introduced by Peschon et al. [50]. Thereafter, various mathematical models for reactive power optimization have been developed [22, 53]; Lee and Yang [40]; Quintana and Santos-Nieto [17, 51]. The study on environmental power dispatch models developed from 1960s to 1970s was surveyed by Happ [28]. Thereafter, different classical optimization models developed in the past century for environmental-economic power generation (EEPG) problems have been surveyed [14, 42, 62] in the past.

Now, in the context of thermal power plant operations, it is worthy to mention that coal as fossil-fuel used to generate power produces atmospheric emissions, namely Carbon oxides (CO_x), Sulfur oxides (SO_x) and oxides of Nitrogen (NO_x), are the major and harmful gaseous pollutants. Pollution affects not only humans, but also other species including plants on the earth.

The constructive optimization model for minimization of thermal power plant emissions was first introduced by Gent and Lament [24]. Thereafter, the field was explored by Sullivan and Hackett [61] among other active researchers in the area of study.

Now, consideration of both the aspects of economic power generation and reduction of emissions in a framework of mathematical programming was initially studied by Zahavi and Eisenberg [70], and thereafter optimization models for EEPG problems were investigated [11, 63] in the past.

During 1990s, emissions control problems were seriously considered and different strategic optimization approaches were developed with the consideration of 1990s Clean Air Amendment by the active researchers in the field and well documented [20, 30, 60] in the literature. Different mathematical models for reactive power optimization have also been developed during 1990 by the eminent researchers [27, 41, 67]; Bansilal and Parthasarathy [7, 40] in this field. Thereafter, different mathematical programming approaches for real and reactive power optimizations have been presented [9, 15, 21, 26, 33, 34, 59]; Abou et al. [3] and widely circulated in the literature. Here, it is to be mentioned that in most of the previous approaches the inherent multiobjective decision making problems are solved by transforming them into single objective optimization problems. As a result, decision deadlock often arises there concerning simultaneous optimization of both the objectives.

To overcome the above difficulty, GP as a robust and flexible tool for multiobjective decision analysis and which is based on the satisficing (coined by the noble laureate [58] philosophy has been studied [45] to obtain the goal oriented solution of economic-emission power dispatch problems.

During the last decade, different multiobjective optimization methods for EEPG problems have been studied [1, 4, 64] by considering the Clean Air Act Amendment.

The traditional stochastic programming (SP) approaches to EEPG problems was studied [18, 68] in the past. FP approach to EEPG problems has been discussed [8, 66]. But, the extensive study in this area is at an early stage.

To solve thermal power planning problems, consideration of both the aspects of real and reactive power generation and dispatch in the framework of a mathematical programming model was initially studied by Jolissaint, Arvanitidis and Luenberger [38], and thereafter optimization models for combined real and reactive power management problems were investigated [39, 41, 65] in the past. But, the deep study in this area is at an early stage.

GA for solving the large-scale economic dispatch was first studied by Chen and Chang [13]. Then, several soft computing approaches to EEPG problems have also been studied [2, 9, 26] by the active researchers in this field.

Now, it is to be observed that the objectives of power system operation and control are highly conflict each other. As an essence, optimization of objectives in a hierarchical structure on the basis of needs of DMs can be considered. As such, bilevel programming (BLP) [49] in hierarchical decision system might be an effective one for solving the problems. Although, the problem of balancing thermal power supply and market demand have been studied [71] in the recent past, but the study in this area is yet to be explore in the literature. Moreover, the MOBLP approach to optimal real and reactive power management problem by employing GA based FGP method is yet to appear in the literature.

In this paper, the GA base FGP approach is used to formulate and solve MOBLP for combined optimal P-Q management problem. In the model formulation, the *minsum* FGP [48] the most widely used and simplest version of FGP is used to achieve a rank based power generation decision in an inexact decision environment. In the decision making process, a GA scheme is employed at two different stages. At the first stage, individual optimal decisions of the objectives are determined for fuzzy goal description of them. At the second stage, evaluation of goal achievement function for minimization of the weighted under-deviational variables of the membership goals associated with the defined fuzzy goals is considered for achieving the highest membership value (unity) of the defined fuzzy goals on the basis of hierarchical order of optimizing them in the decision situation. A case example of IEEE 6-Generator 30-Bus System is considered to illustrate the potential use of the approach.

The chapter is organizing as follows. Section 2 contains the description of proposed problem by defining the objectives and constraints in power generation system. Section 3 provides the MOBLP model formulation by defining the leaders and followers objectives and decision vector. In Sect. 4, computational steps of the proposed GA scheme for modeling and solving the problem is presented. In Sect. 5 the FGP Model formulation of the proposed problem is presented. Section 6 gives an illustrative case example in order to demonstrate the feasibility and efficiency of the proposed approach. Finally, Sect. 7 provides some general conclusions and future research.

3 Problem Description

Let there be N number of generators present in the system. $P_{gi,}$ V_i and T_i be the decision variable of generating power, generator voltages and transformer tap setting in the system. Then, let P_D be the total demand of power (in p.u.), T_L be the total transmission-loss (in p.u), P_L be the real power losses in the system and V_D be the load-bus voltage deviation associated with the system.

The objectives and constraints of the proposed P-Q management problem are discussed as follows.

3.1 Description of Objective Functions

3.1.1 Fuel-Cost Function

The total fuel-cost ($/hr) function associated with generation of power from all generators of the system can be expressed as:

$$F_C = \sum_{i=1}^{N} (a_i + b_i P_{gi} + c_i P_{gi}^2), \tag{1}$$

where a_i, b_i and c_i are the estimated cost-coefficients associated with generation of power from i-th generator.

3.1.2 Emission Function

In a thermal power plant operational system, various types of pollutions are discharged to the earth's Environment due to burning of coal for generation of power.

The total emission (ton/hr) can be expressed as:

$$E = \sum_{i=1}^{N} 10^{-2}(\alpha_i + \beta_i P_{gi} + \gamma_i P_{gi}^2) + \zeta_i \exp(\lambda_i P_{gi}), \tag{2}$$

where $\alpha_i, \beta_i, \gamma_i, \zeta_i, \lambda_i$ are emission-coefficients associated with generation of power from i-th generator.

3.1.3 Power-Losses Function

The real power losses in MW can be expressed as:

$$P_L = \sum_{k=1}^{nl} g_k [V_i^2 + V_j^2 - 2V_i V_j cos(\delta_i - \delta_j)], \tag{3}$$

where 'nl' is the number of transmission lines, g_k is the conductance of the kth line, V_i and V_j are the voltage magnitude and δ_i and δ_j are the voltage phase angle at the end buses i and j of the kth line respectively.

3.1.4 Voltage Profile (VP) Improvement Function

The improvement of voltage profile is nothing but minimizing the bus voltage deviation (V_D) from 1.0 per unit

The objective function can be expressed as:

$$V_D = \sum_{i \in NL} |V_i - 1.0|, \tag{4}$$

where NL is the number of load buses.

3.2 Description of System Constraints

The system constraints which are commonly involved with the problem are defined as follows.

3.2.1 Power Balance Constraint

The generation of total power must cover the total demand (P_D) and total transmission-loss (T_L) inherent to a thermal power generation system.

The total power balance constraint can be obtained as:

$$\sum_{i=1}^{N} P_{gi} - (P_D + T_L) = 0, \tag{5}$$

The transmission-loss can be modeled as a function of generators' output and that can be expressed as:

$$T_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_{gi} B_{ij} P_{gj} + \sum_{i=1}^{N} B_{0i} P_{gi} + B_{00}, \tag{6}$$

where B_{ij}, B_{0i} and B_{00} are called Kron's loss-coefficients or B-coefficients [66] associated with the power transmission network.

The Real power balance is as follows:

$$P_{gi} - P_{di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)] = 0 \tag{7}$$

The Reactive power balance is:

$$Q_{gi} - Q_{di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \sin(\delta_i - \delta_j) + B_{ij} \cos(\delta_i - \delta_j)] = 0, \tag{8}$$

where i = 1, 2, …, NB; ‘NB’ is the number of buses; P_{gi} and Q_{gi} are the i-th generator real and reactive power respectively; P_{di} and Q_{di} are the i-th load real and reactive power respectively; G_{ij} and B_{ij} are the transfer conductance and susceptance between bus i and bus j respectively.

3.2.2 Generator Constraints

In an electric power generation and dispatch system, the constraints on the generators can be considered as:

$$\begin{aligned} &P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max}, \\ &Q_{g_i}^{\min} \le Q_{g_i} \le Q_{g_i}^{\max}, \\ &V_{g_i}^{\min} \le V_{g_i} \le V_{g_i}^{\max}, \quad i = 1,2,\ldots,N \end{aligned} \tag{9}$$

where P_{gi}, Q_{gi} and V_{gi} are the active power, reactive power and generator bus voltage, respectively. ‘N’ is the number of generators in the system.

3.2.3 Transformer Constraints

Transformer tap settings (T) are bounded as follows:

$$T_i^{\min} \le T_i \le T_i^{\max}, i = 1,\ldots,NT \tag{10}$$

where ‘NT’ is the number of transformers.

3.2.4 Shunt VAR Constraints

Shunt VAR compensations are restricted by their limits as:

$$Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max}, i = 1,\ldots,N_c \tag{11}$$

where ‘N_c’ is the number of shunt VAR.

3.2.5 Security Constraints

These include the constraints of voltages at load buses V_L as follows

$$V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max}, i = 1,\ldots,NL \tag{12}$$

Now, MOBLP formulation of the proposed problem for minimizing the objective functions is presented in the following Sect. 4.

4 MOBL Formulation of the Problem

In MOBLP formulation of the proposed problem, minimization of total fuel-cost and minimization of real power losses, the most important objectives in power system operation and planning, are considered as leader's problem and minimization of total environmental emission and voltage deviation are considered as follower's problem in the hierarchical decision system.

Now, the MOBLP model formulation of the proposed problem is presented in the following Sect. 4.1.

4.1 MOBLP Model Formulation

In a BLP model formulation, the vector of decision variables are divided into two distinct vectors and assigned them separately to the DMs for controlling individually.

Let **D** be the vector of decision variables in a thermal power supply system. Then, let $\mathbf{D_L}$ and $\mathbf{D_F}$ be the vectors of decision variables controlled independently by the leader and follower, respectively, in the decision situation, where L and F stand for leader and follower, respectively.

Then, BLP model of the problem appears as [49]:

Find $\mathbf{D}(\mathbf{D_L},\mathbf{D_F})$ so as to:

$$\underset{\mathbf{D_L}}{\text{Minimize}}\ \mathrm{F_C} = \sum_{i=1}^{N} (a_i + b_i P_{gi} + c_i P_{gi}^2)$$

$$\underset{\mathbf{D_L}}{\text{Minimize}}\ \mathrm{P_L} = \sum_{k=1}^{nl} g_k[V_i^2 + V_j^2 - 2V_i V_j cos(\delta_i - \delta_j)]$$

(Leader's problem) and, for given $\mathbf{D_L}, \mathbf{D_F}$ solves

$$\underset{\mathbf{D_F}}{\text{Minimize}}\ \mathrm{E} = \sum_{i=1}^{N} 10^{-2}(\alpha_i + \beta_i P_{gi} + \gamma_i P_{gi}^2) + \zeta_i \exp(\lambda_i P_{gi})$$

$$\underset{\mathbf{D_F}}{\text{Minimize}}\ \mathrm{V_D} = \sum_{i \in NL} |V_i - 1.0|$$

(Follower's problem) subject to the system constraints in (5)–(12)

$$\underset{\mathbf{D_L}}{\text{Minimize}}\ F_C = \sum_{i=1}^{N}(a_i + b_i P_{gi} + c_i P_{gi}^2)$$

$$\underset{\mathbf{D_L}}{\text{Minimize}}\ P_L = \sum_{k=1}^{nl} g_k[V_i^2 + V_j^2 - 2V_i V_j cos(\delta_i - \delta_j)]$$

$$\text{(Leader'sproblem)}$$

and, for given $\mathrm{D_L}$, $\mathrm{D_F}$solves

$$\underset{\mathbf{D_F}}{\text{Minimize}}\ E = \sum_{i=1}^{N} 10^{-2}(\alpha_i + \beta_i P_{gi} + \gamma_i P_{gi}^2) + \zeta_i exp(\lambda_i P_{gi})$$

$$\underset{\mathbf{D_F}}{\text{Minimize}}\ V_D = \sum_{i \in NL} | V_i - 1.0 |$$

$$\text{(Follower's problem)} \qquad (13)$$

where $\mathbf{D_L} \cap \mathbf{D_F} = \varphi$, $\mathbf{D_L} \cup \mathbf{D_F} = \mathbf{D}$ and $\mathbf{D} = \{\mathbf{P_g}; \mathbf{V_g} \text{ and } \mathbf{T}\} \in S(\neq \varphi)$, and where S denotes the feasible solution set, ∩ and ∪ stand for the mathematical operations 'intersection' and 'union', respectively.

Now, the GA scheme employed for modeling and solving the problem in (13) in the framework of an FGP approach is presented in the following Sect. 5.

5 GA Scheme for the Problem

In the literature of GAs, there is a variety of schemes [25, 44] for generating new population with the use of different operators: selection, crossover and mutation.

In the present GA scheme, binary representation of each candidate solution is considered in the genetic search process. The initial population (the initial feasible solution individuals) is generated randomly. The fitness of each feasible solution individual is then evaluated with the view to optimize an objective function in the decision making context.

The basic steps of the GA scheme with the core functions adopted in the solution search process are presented in the following algorithmic steps.

5.1 Steps of the GA Algorithm

Step 1. *Representation and initialization*

Let E denote the double vector representation of a chromosome in a population as $E = \{x_1, x_2, \ldots, x_n\}$. The population size is defined by pop_size, where pop_size chromosomes are randomly initialized in the search domain.

Step 2. *Fitness function*

The fitness value of each chromosome is determined by evaluating an objective function. The fitness function is defined as

$$eval(E_v) = (Z_k)_v, \quad k = 1, 2;\ v = 1, 2, \ldots, \text{pop\_size},$$

where, Z_k represents the objective function of the k-th level DM, and where the subscript v is used to indicate the fitness value of the v-th chromosome, $v = 1, 2, \ldots,$ pop_size.

The best chromosome with largest fitness value at each generation is determined as:

$$E^* = \max\{eval(E_v)|v = 1, 2, \ldots, \text{pop\_size}\}, \text{or},$$
$$E^* = \min\{eval(E_v)|v = 1, 2, \ldots, \text{pop\_size}\}$$

which depends on searching of the maximum or minimum value of an objective function.

Step 3. *Selection*

The simple roulette-wheel scheme [25] is used for the selection of two parents for mating purpose in the genetic search process.

Step 4. *Crossover*

The parameter p_c is defined as the probability of crossover. The single-point crossover in [25] of a genetic system is applied here from the view point that the resulting offspring always satisfy the system constraints set S. Here, a chromosome is selected as a parent for a defined random number $r \in [0, 1]$, if $r < p_c$ is satisfied.

Step 5. *Mutation*

Mutation mechanism is applied over the population after performing crossover operation. It alters one or more genes of a selected chromosome to re-introduce the genetic material to gain extra variability for fitness strength in the population. As in the conventional GA scheme, a parameter p_m of the genetic system is defined as the probability of mutation. The mutation operation is performed on a bit-by-bit basis, where for a random number $r \in [0, 1]$ a chromosome is selected for mutation provided $r < p_m$.

Step 6. *Termination*

The execution of the whole process terminates when the fittest chromosome is reported at a certain generation number in the solution search process.

The pseudo code of the standard genetic algorithm is presented as:

```
Initialize population of chromosomes $E(x)$
Evaluate the initialized population by computing its fitness measure
While not termination criteria do
$x := x + 1$
Select $E(x+1)$ from $E(x)$
Crossover $E(x+1)$
Mutate $E(x+1)$
Evaluate $E(x+1)$
End While
```

Now, FGP formulation of the problem in (13) by defining the fuzzy goals is presented in the next Sect. 6.

6 FGP Model Formulation of the Problem

In a power generation decision context, it is assumed that the objectives in both the levels are motivated to cooperative to each other and each optimizes his/her benefit by paying an attention to the benefit of other one. Here, since leader is in the leading position to make own decision, relaxation on the decision of leader is essentially needed to make a reasonable decision by follower to optimize the objective function to a certain level of satisfaction. Therefore, relaxation of individual optimal values of both the objectives as well as the decision vector $\mathbf{D_L}$ controlled by leader up to certain tolerance levels need be considered to make a reasonable balance of execution of decision powers of the DMs.

To cope with the above situation, a fuzzy version of the problem in (13) would be an effective one in the decision environment.

The fuzzy description of the problem is presented as follows Section.

6.1 Description of Fuzzy Goals

In a fuzzy decision situation, the objective functions are transformed into fuzzy goals by means of assigning an imprecise aspiration level to each of them.

In the sequel of making decision, since individual minimum values of the objectives are always acceptable by each DM, the independent best solutions of leader and follower are determined first as $(\mathbf{D}_{\mathbf{L}}^{lb}, \mathbf{D}_{\mathbf{F}}^{lb}; \mathrm{F}_{\mathrm{C}}^{lb}, \mathrm{P}_{\mathrm{L}}^{lb})$ and $(\mathbf{D}_{\mathbf{L}}^{fb}, \mathbf{D}_{\mathbf{F}}^{fb}; \mathrm{E}^{fb}, \mathrm{V}_{\mathrm{D}}^{fb})$, respectively, by using the GA scheme, where *lb* and, *fb* stand for leader's best and follower's best, respectively.

Then, the fuzzy goals of the leader and follower can be successively defined as:

$$\begin{aligned} &\mathrm{F_C} \underset{\sim}{<} \mathrm{F_C^{\mathit{lb}}} \text{ and } \mathrm{P_L} \underset{\sim}{<} \mathrm{P_L^{\mathit{lb}}} \\ &\mathrm{E} \underset{\sim}{<} \mathrm{E^{\mathit{fb}}} \text{ and } \mathrm{V_D} \underset{\sim}{<} \mathrm{V_D^{\mathit{fb}}} \end{aligned} \tag{14}$$

where '$\underset{\sim}{<}$' Refers to the fuzziness of an aspiration level and it is to be understood as 'essentially less than' [73].

Again, since maximum values of the objectives when calculated in isolation by the DMs would be the most dissatisfactory ones, the worst solutions of leader and follower can be obtained by using the same GA scheme as $(\mathbf{D_L^{\mathit{lw}}}, \mathbf{D_F^{\mathit{lw}}}; \mathrm{F_C^{\mathit{lw}}},\ \mathrm{P_L^{\mathit{lw}}})$ and $(\mathbf{D_L^{\mathit{fw}}}, \mathbf{D_F^{\mathit{fw}}}; \mathrm{E^{\mathit{fw}}}, \mathrm{V_D^{\mathit{fw}}})$, respectively, where *lw* and, *fw* stand for leader's worst and follower's worst, respectively.

Then, $\mathrm{F_C^{\mathit{lw}}}, \mathrm{P_L^{\mathit{lw}}}, \mathrm{E^{\mathit{fw}}}$ and $\mathrm{V_D^{\mathit{fw}}}$would be the upper-tolerance limits of achieving the aspired levels of $\mathrm{F_C}, \mathrm{P_L}$,EandV$_{\mathrm{D}}$, respectively.

The vector of fuzzy goals associated with the control vector $\mathbf{D_L}$ can be defined as:

$$\mathbf{D_L} \underset{\sim}{<} \mathbf{D_L^{\mathit{lb}}} \tag{15}$$

In the fuzzy decision situation, it may be noted that the increase in the values of fuzzily described goals defined by the goal vector in (15) would never be more than the corresponding upper-bounds of the power generation capacity ranges defined in (9).

Let $\mathbf{D_L^t}, (\mathbf{D_L^t} < \mathbf{D_L^{max}})$, be the vector of upper-tolerance limits of achieving the goal levels of the vector of fuzzy goals defined in (15).

Now, the fuzzy goals are to be characterized by the respective membership functions for measuring their degree of achievements in a fuzzy decision environment.

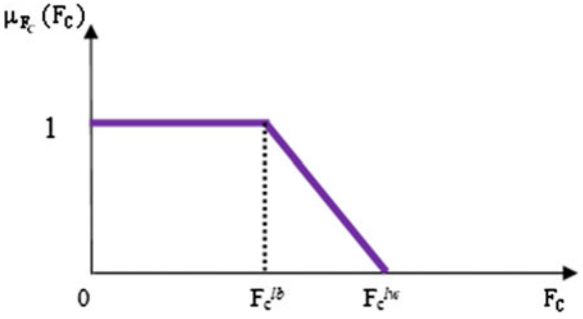
6.2 Characterization of Membership Function

The membership function representation of the fuzzy objective goal of fuel cost function under the control of leader appears as:

$$\mu_{\mathrm{F_C}}[\mathrm{F_c}] = \begin{cases} 1, & \text{if } \mathrm{F_C} \le \mathrm{F_C^{\mathit{lb}}} \\ \dfrac{\mathrm{F_c^{\mathit{lw}}} - \mathrm{F_C}}{\mathrm{F_C^{\mathit{lw}}} - \mathrm{F_C^{\mathit{lb}}}}, & \text{if } \mathrm{Z_1^{\mathit{lb}}} < \mathrm{F_C} \le \mathrm{F_C^{\mathit{lw}}} \\ 0, & \text{if } \mathrm{F_C} > \mathrm{F_C^{\mathit{lw}}} \end{cases} \tag{16}$$

where $(\mathrm{F_C^{\mathit{lw}}} - \mathrm{F_C^{\mathit{lb}}})$ is the tolerance range for achievement of the fuzzy goal defined in (14).

Fig. 3 The membership function of fuzzy goal in (16)



The graphical representation of the above membership functions in (16) is displayed in Fig. 3.

The membership function representation of the fuzzy objective goal of power-loss function under the control of leader appears as:

$$\mu_{P_L}[P_L] = \begin{cases} 1, & \text{if } P_L \leq P_L^{lb} \\ \frac{\mathrm{P}_L^{lw} - P_L}{\mathrm{P}_L^{lw} - P_L^{lb}}, & \text{if } P_L^{lb} < P_L \leq \mathrm{P}_L^{lw} \\ 0, & \text{if } \mathrm{F_C} > \mathrm{P}_L^{lw} \end{cases} \tag{17}$$

where $(P_L^{lw} - P_L^{lb})$ is the tolerance range for achievement of the fuzzy goal defined in (14).

Similarly, the membership function representations of the fuzzy objective goals of emission and voltage profile improvement function under the control of follower are successively appear as:

$$\mu_E[E] = \begin{cases} 1, & \text{if } E \leq E^{fb} \\ \frac{\mathrm{E}^{fw} - E}{\mathrm{E}^{fw} - E^{fb}}, & \text{if } E^{fb} < E \leq \mathrm{E}^{fw} \\ 0, & \text{if } E > \mathrm{E}^{fw} \end{cases} \tag{18}$$

where $(E^{fw} - E^{fb})$ is the tolerance range for achievement of the fuzzy goal defined in (14).

$$\mu_{V_D}[V_D] = \begin{cases} 1, & \text{if } V_D \leq V_D^{fb} \\ \frac{V_D^{fw} - V_D}{V_D^{fw} - V_D^{fb}}, & \text{if } V_D^{fb} < V_D \leq V_D^{fw} \\ 0, & \text{if } V_D > V_D^{fw} \end{cases} \tag{19}$$

where $(V_D^{fw} - V_D^{fb})$ is the tolerance range for achievement of the fuzzy goal defined in (14).

The membership function of the fuzzy decision vector $\mathbf{D_L}$ of the leader appears as:

$$\mu_{\mathbf{D_L}}[\mathbf{D_L}] = \begin{cases} 1, & \text{if } \mathbf{D_L} \leq \mathbf{D_L^{\mathit{lb}}} \\ \frac{\mathbf{D_L^t} - \mathbf{D_L}}{\mathbf{D_L^t} - \mathbf{D_L^{\mathit{lb}}}}, & \text{if } \mathbf{D_L^{\mathit{lb}}} < \mathbf{D_L} \leq \mathbf{D_L^t} \\ 0, & \text{if } \mathbf{D_L} > \mathbf{D_L^t} \end{cases} \tag{20}$$

where $(\mathbf{D_L^t} - \mathbf{D_L^{\mathit{lb}}})$ is the vector of tolerance ranges for achievement of the fuzzy decision variables associated with $\mathbf{D_L}$ defined in (15).

Note 1: $\mu[.]$represents membership function.

Now, *minsum* FGP formulation of the proposed problem is presented in the following section.

6.3 Minsum FGP Model Formulation

In the process of formulating FGP model of a problem, the membership functions are transformed into membership goals by assigning the highest membership value (unity) as the aspiration level and introducing under- and over-deviational variables to each of them. In minsum FGP, minimization of the sum of weighted under-deviational variables of the membership goals in the goal achievement function on the basis of relative weights of importance of achieving the aspired goal levels is considered.

The *minsum* FGP model can be presented as [48]:

Find $\mathrm{D}(\mathrm{D_L}, \mathrm{D_F})$ so as to:

$$\text{Minimize} : \mathrm{Z} = \sum_{\mathrm{k}=1}^{4} \mathrm{w_k^- d_k^-} + \mathrm{w_5^- d_5^-}$$

and satisfy

$$\begin{aligned} &\mu_{\mathrm{F_C}} : \frac{\mathrm{F_C^{lw}} - \mathrm{F_C}}{\mathrm{F_C^{lw}} - \mathrm{Z_1^{lb}}} + \mathrm{d_1^-} - \mathrm{d_1^+} = 1, \\ &\mu_{\mathrm{P_L}} : \frac{\mathrm{P_L^{lw}} - \mathrm{P_L}}{\mathrm{P_L^{lw}} - \mathrm{P_L^{lb}}} + \mathrm{d_2^-} - \mathrm{d_2^+} = 1, \\ &\mu_{\mathrm{E}} : \frac{\mathrm{E^{fw}} - \mathrm{E}}{\mathrm{E^{fw}} - \mathrm{E^{fb}}} + \mathrm{d_3^-} - \mathrm{d_3^+} = 1, \\ &\mu_{\mathrm{V_D}} : \frac{\mathrm{V_D^{fw}} - \mathrm{V_D}}{\mathrm{V_D^{fw}} - \mathrm{V_D^{fb}}} + \mathrm{d_4^-} - \mathrm{d_4^+} = 1, \\ &\mu_{\mathrm{D_L}} : \frac{\mathbf{D_L^t} - \mathbf{D_L}}{\mathbf{D_L^t} - \mathbf{P_{GL}}^{\mathbf{lb}}} + \mathbf{d_5^-} - \mathbf{d_5^+} = \mathbf{I} \end{aligned} \tag{21}$$

subject to the set of constraints defined in (5)–(12)

where $d_k^-, d_k^+ \geq 0$, (k = 1, …, 4) represent the under- and over-deviational variables, respectively, associated with the respective membership goals. $d_5^-, d_5^+ \geq 0$ represent the vector of under- and over-deviational variables, respectively, associated with the membership goals defined for the vector of decision variables in $\mathbf{D_L}$, and where I is a column vector with all elements equal to 1 and the dimension of it depends on the dimension of $\mathbf{D_L}$. Z represents goal achievement function, $w_k^- > 0$, k = 1, 2, 3, 4 denote the relative numerical weights of importance of achieving the aspired goal levels, and $w_5^- > 0$ is the vector of numerical weights associated with d_5^-, and they are determined by the inverse of the tolerance ranges [48] for achievement of the goal levels in the decision making situation.

Now, the effective use of the *minsum* FGP model in (21) is demonstrated via a case example presented in the next section.

7 A Demonstrative Case Example

The standard IEEE 30-bus 6-generator test system [1] is considered to illustrate the potential use of the approach.

The pictorial representation of single-line diagram of IEEE 30-bus test system is shown in the Fig. 4.

The system shown in Fig. 4 has 6 generators and 41 lines and the total system demand for the 21 load buses is 2.834 p.u. The detailed data are given in Tables 1, 2, 3, 4 and 5.

The *B-coefficients* [66] are presented as follows:

$$B = \begin{bmatrix} 0.1382 & -0.0299 & 0.0044 & -0.0022 & -0.0010 & -0.0008 \\ -0.0299 & 0.0487 & -0.0025 & 0.0004 & 0.0016 & 0.0041 \\ 0.0044 & -0.0025 & 0.0182 & -0.0070 & -0.0066 & -0.0066 \\ -0.0022 & 0.0004 & -0.0070 & 0.0137 & 0.0050 & 0.0033 \\ -0.0010 & 0.0016 & -0.0066 & 0.0050 & 0.0109 & 0.0005 \\ -0.0008 & 0.0041 & -0.0066 & 0.0033 & 0.0005 & 0.0244 \end{bmatrix}$$

$$B_0 = [-0.0107 \quad 0.0060 \quad -0.0017 \quad 0.0009 \quad 0.0002 \quad 0.0030], \quad B_{00} = 9.8573E-4$$

Now, in the proposed MOBLP formulation of the problem, without loss of generality it is assumed that the power generation, $\mathbf{P_G}(P_{g1},P_{g2},P_{g3},P_{g4},P_{g5}P_{g6})$ are under the control of the leader, and the generator bus voltages, $\mathbf{V}$ $(V_1,V_2,V_5,V_8,V_{11}, V_{13})$ and transformer tap-setting, $\mathbf{T}$ $(T_{11}, T_{12}, T_{15}, T_{36})$ are assigned to the follower.

Using the data Tables, the individual best and worst solution of each of the objectives under leader and follower of the proposed MOBLP can be calculated as follows.

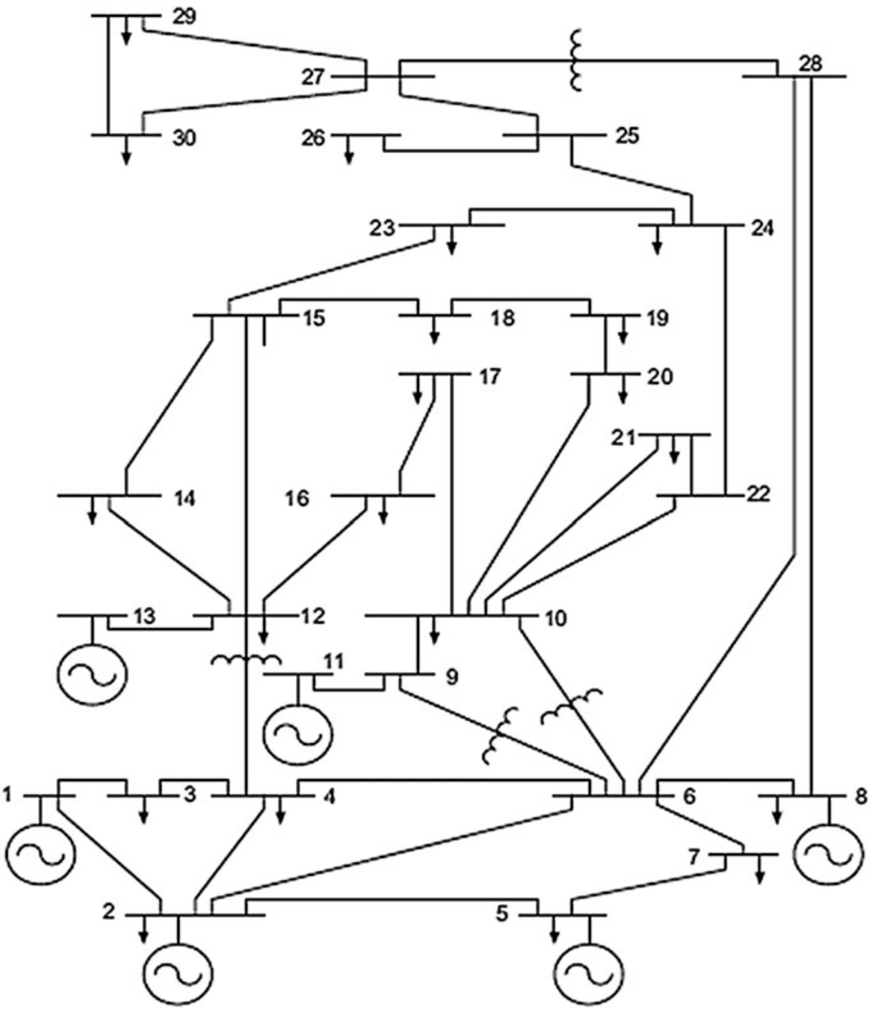


Fig. 4 Single-line diagram of IEEE 30-bus test system

Find $\boldsymbol{P_G}(P_{g1}, P_{g2}, P_{g3}, P_{g4}, P_{g5}P_{g6})$ so as to:

$$\begin{aligned} \mathit{Minimize}\, F_C(\boldsymbol{P_G}) = {} & (10 + 200P_{g1} + 100P_{g1}^2 + 10 + 150P_{g2} + 120P_{g2}^2 + 20 \\ & + 180P_{g3} + 40P_{g3}^2 + 10 + 100P_{g4} + 60P_{g4}^2 + 20 + 180P_{g5} \\ & + 40P_{g5}^2 + 10 + 150P_{g6} + 100P_{g6}^2) \end{aligned}$$

(leader's objective 1)

Table 1 Data description of power generation costs and emission-coefficients

Power generation	g_1	g_2	g_3	g_4	g_5	g_6
Cost coefficients						
a	10	10	20	10	20	10
b	200	150	180	100	180	150
c	100	120	40	60	40	100
Emission coefficients						
α	4.091	2.543	4.258	5.426	4.258	6.131
β	−5.554	−6.047	−5.094	−3.550	−5.094	−5.555
γ	6.490	5.638	4.586	3.380	4.586	5.151
ζ	2.0E– 4	5.0E– 4	1.0E– 6	2.0E– 3	1.0E– 6	1.0E−5
λ	2.857	3.333	8.000	2.000	8.000	6.667

Table 2 Data description of generator limit (in p.u)

Generator no.	$P_{gi\ min}$	$P_{gi\ max}$	$Q_{gi\ min}$	$Q_{gi\ max}$	$V_{i\ min}$	$V_{i\ max}$
1	0.05	0.50	− 0.15	0.45	1.000	1.071
2	0.05	0.60	− 0.10	0.40	1.000	1.082
3	0.05	1.00	− 0.15	0.50	1.000	1.010
4	0.05	1.20	− 0.15	0.625	1.000	1.010
5	0.05	1.00	− 0.2	0.6	1.000	1.045
6	0.05	0.60	–	–	1.000	1.060

Table 3 Description of load data

Bus no.	Load		Bus no.	Load	
	P (p.u.)	Q (p.u.)		P (p.u.)	Q (p.u.)
1	0.000	0.000	16	0.035	0.018
2	0.217	0.127	17	0.090	0.058
3	0.024	0.012	18	0.032	0.009
4	0.076	0.016	19	0.095	0.034
5	0.942	0.190	20	0.022	0.007
6	0.000	0.000	21	0.175	0.112
7	0.228	0.109	22	0.000	0.000
8	0.300	0.300	23	0.032	0.016
9	0.000	0.000	24	0.087	0.067
10	0.058	0.020	25	0.000	0.000
11	0.000	0.000	26	0.035	0.023
12	0.112	0.075	27	0.000	0.000
13	0.000	0.000	28	0.000	0.000
14	0.062	0.016	29	0.024	0.009
15	0.082	0.025	30	0.106	0.019

Table 4 Description of line data

Line no.	From bus	To bus	Line impedance	
			R (p.u.)	X (p.u.)
1	1	2	0.0192	0.0575
2	1	3	0.0452	0.0852
3	2	4	0.0570	0.1737
4	3	4	0.0132	0.0379
5	2	5	0.0472	0.1982
6	2	6	0.0581	0.1763
7	4	6	0.0119	0.0414
8	5	7	0.0460	0.0060
9	6	7	0.0267	0.0820
10	6	8	0.0120	0.0420
11	6	9	0.0000	0.2080
12	6	10	0.0000	0.5560
13	9	11	0.0000	0.2080
14	9	10	0.0000	0.1100
15	4	12	0.0000	0.2560
16	12	13	0.0000	0.1400
17	12	14	0.1231	0.2559
18	12	15	0.0662	0.1304
19	12	16	0.0945	0.1987
20	14	15	0.2210	0.1997
21	16	17	0.0824	0.1932
22	15	18	0.1070	0.2185
23	18	19	0.0639	0.1292
24	19	20	0.0340	0.0680
25	10	20	0.0936	0.2090
26	10	17	0.0324	0.0845
27	10	21	0.0348	0.0749
28	10	22	0.0727	0.1499
29	21	22	0.0116	0.0236
30	15	23	0.1000	0.2020
31	22	24	0.1150	0.1790
32	23	24	0.1320	0.2700
33	24	25	0.1885	0.3292
34	25	26	0.2544	0.3800
35	25	27	0.1093	0.2087
36	28	27	0.0000	0.3960
37	27	29	0.2198	0.4153
38	27	30	0.3202	0.6027
39	29	30	0.2399	0.4533
40	8	28	0.6360	0.2000
41	6	28	0.0169	0.0599

Table 5 Description of control variables

Control variable	Minimum value	Maximum value
P_{G1}	50	200
P_{G2}	20	80
P_{G5}	15	50
P_{G8}	10	35
P_{G11}	10	30
P_{G13}	12	40
Q_{G2}	−0.2	0.6
Q_{G5}	−0.15	0.625
Q_{G8}	−0.15	0.5
Q_{G11}	−0.10	0.40
Q_{G13}	−0.15	0.45
V_1	0.95	1.1
V_2	0.95	1.1
V_5	0.95	1.1
V_8	0.95	1.1
V_{11}	0.95	1.1
V_{13}	0.95	1.1
T_{11}	0.9	1.1
T_{12}	0.9	1.1
T_{15}	0.9	1.1
T_{36}	0.9	1.1
Q_{c10}	0.0	5.0
Q_{c12}	0.0	5.0
Q_{c15}	0.0	5.0
Q_{c17}	0.0	5.0
Q_{c20}	0.0	5.0
Q_{c21}	0.0	5.0
Q_{c23}	0.0	5.0
Q_{c24}	0.0	5.0
Q_{c29}	0.0	5.0

subject to

$$P_{g1} + P_{g2} + P_{g3} + P_{g4} + P_{g5} + P_{g6} - (2.834 + L_T) = 0,$$

(Power balance constraint)

and

$$\begin{gathered}0.05 \le P_{g1} \le 0.50, 0.05 \le P_{g2} \le 0.60,\\ 0.05 \le P_{g3} \le 1.00, 0.05 \le P_{g4} \le 1.20,\\ 0.05 \le P_{g5} \le 1.00, \quad 0.05 \le P_{g6} \le 0.60,\end{gathered}$$

(Power generator capacity constraints)

Table 6 The parameter values used in GA

Parameter	Value
Number of individual in the initial population	50
Selection	Roulette-wheel
Crossover function	Single point
Crossover probability	0.8
Mutation probability	0.06
Maximum generation number	100

where
$$\begin{aligned} L_T = {} & 0.1382P_{g1}^2 + 0.0487P_{g2}^2 + 0.0182P_{g3}^2 + 0.0137P_{g4}^2 + 0.0109P_{g5}^2 + 0.0244P_{g6}^2 \\ & - 0.0598P_{g1}P_{g2} + 0.0088P_{g1}P_{g3} - 0.0044P_{g1}P_{g4} - 0.0020P_{g1}P_{g5} - 0.0016P_{g1}P_{g6} \\ & - 0.0050P_{g2}P_{g3} + 0.0008P_{g2}P_{g4} + 0.0032P_{g2}P_{g5} + 0.0082P_{g2}P_{g6} - 0.140P_{g3}P_{g4} \\ & - 0.0132P_{g3}P_{g5} - 0.0132P_{g3}P_{g6} + 0.010P_{g4}P_{g5} + 0.0066P_{g4}P_{g6} + 0.0010P_{g5}P_{g6} \\ & - 0.0107P_{g1} + 0.0060P_{g2} - 0.0017P_{g3} + 0.0009P_{g4} + 0.0002P_{g5} + 0.0030P_{g6} \\ & + 9.8573x10^{-4} \end{aligned}$$

Now, employing the proposed GA scheme the individual best and least solutions of the leader's objectives are determined.

The computer program developed in MATLAB and GAOT (Genetic Algorithm Optimization Toolbox) in MATLAB-Ver. R2010a is used together for the calculation to obtain the results. The execution is made in Intel Pentium IV with 2.66 GHz. Clock-pulse and 4 GB RAM.

Now, the following GA parameter values are introduced during the execution of the problem in different stages.

The parameter values used in genetic algorithm solution are given in Table 6.

The individual best and worst solution of the Fuel-cost function in leader's objectives is obtained as (Table 7):

The individual best and worst solution of the real-power loss function is obtained as:

Table 7 Best and worst solution of fuel-cost

Decision variables and objective functions	Best cost	Worst cost
P_{g1}	0.1220	0.5000
P_{g2}	0.2863	0.6000
P_{g3}	0.5832	0.1397
P_{g4}	0.9922	0.0500
P_{g5}	0.5236	1.000
P_{g6}	0.3518	0.6000
Fuel-cost ($/h)	**595.9804**	**705.2694**
Total transmission losses	0.0255	0.0556

Find $\{\mathbf{V}_{\boldsymbol{k}},\mathbf{T}_{\boldsymbol{l}},(\mathbf{Q}_{\mathbf{c}})_{\boldsymbol{m}}$; k = 1,2,5,8,11,13; l = 11,12,15,36, m = 10,12,15,17,20,21, 24,29$\}$Minimize $\mathrm{P_L} = \sum_{\mathrm{k}=1}^{\mathrm{nl}} \mathrm{g_k}[\mathrm{V}_\mathrm{i}^2 + \mathrm{V}_\mathrm{j}^2 - 2\mathrm{V_i V_j} cos(\delta_\mathrm{i} - \delta_\mathrm{j})]$ (leader's objective 2) subject to

$$\mathrm{P_{gi}} - \mathrm{P_{di}} - \mathrm{V_i}\sum_{\mathrm{j}=1}^{\mathrm{NB}} \mathrm{V_j}[\mathrm{G_{ij}}\cos(\mathrm{delta_i} - \delta_\mathrm{j}) + \mathrm{B_{ij}}\sin(\delta_\mathrm{i} - \delta_\mathrm{j})] = 0$$

$$\mathrm{Q_{gi}} - \mathrm{Q_{di}} - \mathrm{V_i}\sum_{\mathrm{j}=1}^{\mathrm{NB}} \mathrm{V_j}[\mathrm{G_{ij}}\sin(\delta_\mathrm{i} - \delta_\mathrm{j}) + \mathrm{B_{ij}}\cos(\delta_\mathrm{i} - \delta_\mathrm{j})] = 0,$$

where i = 1,2,…,30; $\mathrm{P_{gi}}$ and $\mathrm{Q_{gi}}$ are the i-th generator real and reactive power respectively; $\mathrm{P_{di}}$ and $\mathrm{Q_{di}}$ are the i-th load real and reactive power respectively; $\mathrm{G_{ij}}$ and $\mathrm{B_{ij}}$ are the transfer conductance and susceptance between bus i and bus j respectively.

$$\begin{aligned}&\mathrm{P_{gi}^{min}} \le \mathrm{P_{gi}} \le \mathrm{P_{gi}^{max}},\\ &\mathrm{Q_{g_i}^{min}} \le \mathrm{Q_{g_i}} \le \mathrm{Q_{g_i}^{max}},\\ &\mathrm{V_{g_i}^{min}} \le \mathrm{V_{g_i}} \le \mathrm{V_{g_i}^{max}}, \mathrm{i} = 1,2,\ldots,6\end{aligned}$$

(Generator Constraints)

$$\mathrm{T_i^{min}} \le \mathrm{T_i} \le \mathrm{T_i^{max}}, \quad \mathrm{i} = 11,\ 12,\ 15,\ 36.$$

(Transformer Constraints)

$$\mathrm{Q_{ci}^{min}} \le \mathrm{Q_{ci}} \le \mathrm{Q_{ci}^{max}}, \quad \mathrm{i} = 11,\ 12,\ 15,\ 36.$$

(Shunt VAR Constraints)

$$\mathrm{V_{L_i}^{min}} \le \mathrm{V_{L_i}} \le \mathrm{V_{L_i}^{max}}, \mathrm{i} = 1,\ldots,\mathrm{NL}$$

(Security Constraints)

Now, employing the proposed GA scheme the individual best and least solutions of the objective is determined (Table 8).

Similarly the individual best and worst solutions of the Emission function in the follower's problem can be obtained as:

Table 8 Best and worst solution of power-losses

Decision variables and objective functions	Best power-losses	Worst power-losses
V_1	1.1000	1.0993
V_2	1.0931	1.0967
V_5	1.0736	1.0990
V_8	1.0756	1.0346
V_{11}	1.1000	1.0993
V_{13}	1.1000	0.9517
T_{11}	1.0465	0.9038
T_{12}	0.9097	0.9029
T_{15}	0.9867	0.9002
T_{36}	0.9689	0.9360
Q_{c10}	5.0000	0.6854
Q_{c12}	5.0000	4.7163
Q_{c15}	5.0000	4.4931
Q_{c17}	5.0000	4.5100
Q_{c20}	4.4060	4.4766
Q_{c21}	5.0000	4.6075
Q_{c23}	2.8004	3.8806
Q_{c24}	5.0000	4.2854
Q_{c29}	2.5979	3.2541
Power losses (MW)	**4.5550**	**7.0733**

Find $\boldsymbol{P_G}(P_{g1}, P_{g2}, P_{g3}, P_{g4}, P_{g5}P_{g6})$so as to:

$$\begin{aligned}
\mathit{Minimize}\, Z_2(\boldsymbol{P_{GL}}, \boldsymbol{P_{GF}}) = {} & 10^{-2}(4.091 - 5.554P_{g1} + 6.490P_{g1}^2) + 2.0E - 4\exp(2.857P_{g1}) \\
& + 10^{-2}(2.543 - 6.047P_{g2} + 5.638P_{g2}^2) + 5.0E - 4\exp(3.333P_{g2}) \\
& + 10^{-2}(4.258 - 5.094P_{g3} + 4.586P_{g3}^2) + 1.0E - 6\exp 8.000P_{g3}) \\
& + 10^{-2}(5.326 - 3.550P_{g4} + 3.380P_{g4}^2) + 2.0E - 3\exp(2.000P_{g4}) \\
& + 10^{-2}(4.258 - 5.094P_{g5} + 4.586P_{g5}^2) + 1.0E - 6\exp(8.000P_{g5}) \\
& + 10^{-2}(6.131 - 5.555P_{g6} + 5.151P_{g6}^2) + 1.0E - 5\exp(6.667P_{g6})
\end{aligned}$$

(follower's objective 1)

subject to

$$P_{g1} + P_{g2} + P_{g3} + P_{g4} + P_{g5} + P_{g6} - (2.834 + L_T) = 0,$$

(Power balance constraint)

Table 9 Best and worst solution of emission

Decision variables and objective functions	Best emission	Worst emission
P_{g1}	0.4108	0.5000
P_{g2}	0.4635	0.6000
P_{g3}	0.5442	0.4816
P_{g4}	0.3902	1.2000
P_{g5}	0.5443	0.0500
P_{g6}	0.5153	0.0500
Emission (ton/h)	**0.1952**	**0.2533**
Total Transmission losses	0.0344	0.0476

and

$$
\begin{aligned}
&0.05 \le P_{g1} \le 0.50, 0.05 \le P_{g2} \le 0.60,\\
&0.05 \le P_{g3} \le 1.00, 0.05 \le P_{g4} \le 1.20,\\
&0.05 \le P_{g5} \le 1.00, 0.05 \le P_{g6} \le 0.60,
\end{aligned}
$$

(Power generator capacity constraints)

where
$$
\begin{aligned}
L_T = {} & 0.1382P_{g1}^2 + 0.0487P_{g2}^2 + 0.0182P_{g3}^2 + 0.0137P_{g4}^2 + 0.0109P_{g5}^2 + 0.0244P_{g6}^2\\
& - 0.0598P_{g1}P_{g2} + 0.0088P_{g1}P_{g3} - 0.0044P_{g1}P_{g4} - 0.0020P_{g1}P_{g5} - 0.0016P_{g1}P_{g6}\\
& - 0.0050P_{g2}P_{g3} + 0.0008P_{g2}P_{g4} + 0.0032P_{g2}P_{g5} + 0.0082P_{g2}P_{g6} - 0.140P_{g3}P_{g4}\\
& - 0.0132P_{g3}P_{g5} - 0.0132P_{g3}P_{g6} + 0.010P_{g4}P_{g5} + 0.0066P_{g4}P_{g6} + 0.0010P_{g5}P_{g6}\\
& - 0.0107P_{g1} + 0.0060P_{g2} - 0.0017P_{g3} + 0.0009P_{g4} + 0.0002P_{g5} + 0.0030P_{g6}\\
& + 9.8573X10^{-4}
\end{aligned}
$$

Now, employing the proposed GA scheme the individual best and least solutions of the objective is determined.

The individual best and worst solution of the voltage profile improvement function in follower's problem is obtained as

Find $\{\mathbf{V}_k, \mathbf{T}_l, (\mathbf{Q}_c)_m$; $k = 1,2,5,8,11,13$; $l = 11,12,15,36$, $m = 10,12,15,17,20,21,24,29\}$

$$
\begin{aligned}
\text{Minimize}\, \mathrm{V_D} = {} & \sum |\mathrm{V_i} - 10.0|, i\\
& = 2,3,4,5,7,8,10,12,14,15,16,17,18,19,20,21,23,24,26,29,30
\end{aligned}
$$

(follower's objective 2)

subject to

$$
\mathrm{P_{gi}} - \mathrm{P_{di}} - \mathrm{V_i}\sum_{\mathrm{j}=1}^{\mathrm{NB}} \mathrm{V_j}[\mathrm{G_{ij}}\cos(\delta_\mathrm{i} - \delta_\mathrm{j}) + \mathrm{B_{ij}}\sin(\delta_\mathrm{i} - \delta_\mathrm{j})] = 0
$$

$$
\mathrm{Q_{gi}} - \mathrm{Q_{di}} - \mathrm{V_i}\sum_{\mathrm{j}=1}^{\mathrm{NB}} \mathrm{V_j}[\mathrm{G_{ij}}\sin(\delta_\mathrm{i} - \delta_\mathrm{j}) + \mathrm{B_{ij}}\cos(\delta_\mathrm{i} - \delta_\mathrm{j})] = 0,
$$

where i = 1,2,…,30; P_{gi} and Q_{gi} are the i-th generator real and reactive power respectively; P_{di} and Q_{di} are the i-th load real and reactive power respectively; G_{ij} and B_{ij} are the transfer conductance and susceptance between bus i and bus j respectively.

$$\begin{aligned}
&P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max},\\
&Q_{g_i}^{\min} \le Q_{g_i} \le Q_{g_i}^{\max},\\
&V_{g_i}^{\min} \le V_{g_i} \le V_{g_i}^{\max}, \quad i = 1,2,\ldots,6
\end{aligned}$$

(Generator Constraints)

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 11,\ 12,\ 15,\ 36.$$

(Transformer Constraints)

$$Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max}, \quad i = 11,\ 12,\ 15,\ 36.$$

(Shunt VAR Constraints)

$$V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max}, i = 2,3,4,5,7,8,10,12,14,15,16,17,18,19,20,21,23,24,26,29,30$$

(Security Constraints) Now, employing the proposed GA scheme the individual best and least solutions of the objective is determined.

Now, considering all the objectives of leader and follower simultaneously and following the procedure, the executable FGP model of the problem is obtained as:

Find $\mathbf{P_G}, \mathbf{V}$ and $\mathbf{T}$so as to

$$\begin{aligned}
\text{Minimize}\, Z = {}& 0.009d_1^- + 0.5208d_2^- + 17.211d_3^- + 3.33d_4^- + 7.812d_5^- + 15.689d_6^- + 2.85d_7^- \\
& + 2.74d_8^- + 3.62d_9^- + 3.33d_{10}^-
\end{aligned}$$

and satisfy $\mu_{F_c} : \dfrac{705.2694 - \sum_{i=1}^{N}(a_i + b_iP_{gi} + c_iP_{gi}^2)}{705.2694 - 595.9804} + d_1^- - d_1^+ = 1$

$$\mu_{P_L} : \frac{6.47 - \sum_{k=1}^{41} g_k[V_i^2 + V_j^2 - 2V_iV_j\cos(\delta_i - \delta_j)]}{6.47 - 4.55} + d_2^- - d_2^+ = 1,$$

$$\mu_E : \frac{0.2533 - \sum_{i=1}^{N} 10^{-2}(\alpha_i + \beta_iP_{gi} + \gamma_iP_{gi}^2) + \zeta_i\exp(\lambda_iP_{gi})}{0.2533 - 0.1952} + d_3^- - d_3^+ = 1$$

$$\mu_{VD} : \frac{1.95 - VD}{1.95 - 0.09} + d_4^- - d_4^+ = 1,$$

where $VD = \sum_{i \in NL} |V_i - 1.0|$, i = 2,3,4,5,7,8,10,12,14,15,16,17,18,19,20,21, 23,24,26,29 and 30

Table 10 Best and worst solution of voltage deviation

Decision variables and objective functions	Best voltage deviation	Worst voltage deviation
V_1	1.0100	1.1000
V_2	0.9918	1.0931
V_5	1.0179	1.0736
V_8	1.0183	1.0756
V_{11}	1.0114	1.1000
V_{13}	1.0282	1.1000
T_{11}	1.0265	1.0465
T_{12}	0.9038	0.9097
T_{15}	1.0114	0.9867
T_{36}	0.9635	0.9689
Q_{c10}	4.9420	5.0000
Q_{c12}	1.0885	5.0000
Q_{c15}	4.9985	5.0000
Q_{c17}	0.2395	5.0000
Q_{c20}	4.9958	4.4060
Q_{c21}	4.9075	5.0000
Q_{c23}	4.9863	2.8004
Q_{c24}	4.9663	5.0000
Q_{c29}	2.2325	2.5979
Voltage deviation	**0.0911**	**1.9589**

$$\begin{aligned}
\mu_{P_{g1}} &: \frac{0.25 - P_{g1}}{0.25 - 0.1220} + d_5^- - d_5^+ = 1\\
\mu_{P_{g2}} &: \frac{0.35 - P_{g2}}{0.35 - 0.2863} + d_6^- - d_6^+ = 1\\
\mu_{P_{g3}} &: \frac{0.9321 - P_{g3}}{0.9321 - 0.5823} + d_7^- - d_7^+ = 1\\
\mu_{P_{g4}} &: \frac{0.9901 - P_{g4}}{0.9901 - 0.6254} + d_8^- - d_8^+ = 1\\
\mu_{P5} &: \frac{0.8 - P_{g5}}{0.8 - 0.5236} + d_9^- - d_9^+ = 1\\
\mu_{P5} &: \frac{0.8 - P_{g5}}{0.8 - 0.5236} + d_9^- - d_9^+ = 1\\
\mu_{P6} &: \frac{0.45 - P_{g6}}{0.45 - 0.3518} + d_{10}^- - d_{10}^+ = 1\\
& d_i^-, d_i^+ \geq 0, \quad i = 1, 2, \ldots, 8.
\end{aligned} \tag{22}$$

subject to the given system constraints in (5)–(12).

Now, employing the GA scheme, the problem in (22) is solved to reach the optimal decision. The resultant solution is presented in the Table 11.

The achieved membership values of the objectives are

Table 11 Solutions under the proposed model

Decision variables and objective functions	Achieved solution	
Decision of power generation	Pg_{1}	0.1613
	Pg_{2}	0.3566
	Pg_{3}	0.5519
	Pg_{4}	0.8525
	Pg_{5}	0.3217
	Pg_{6}	0.6000
Decision of generator voltages	V_1	1.062
	V_2	1.021
	V_5	1.045
	V_8	1.064
	V_{11}	1.041
	V_{13}	1.063
Decision of transfor-mar tap setting	T_{11}	0.951
	T_{12}	0.906
	T_{15}	0.972
	T_{36}	0.950
Decision of Shunt VAR compensations	Q_{c10}	3.90
	Q_{c12}	4.15
	Q_{c15}	4.81
	Q_{c17}	3.73
	Q_{c20}	4.617
	Q_{c21}	4.824
	Q_{c23}	3.781
	Q_{c24}	4.512
	Q_{c29}	2.690
Objective functions	Total generation cost ($/h)	612.731
	Power losses (MW)	5.171
	Total emission (ton/h)	0.21269
	Voltage deviation	0.331

$$(\mu_{F_c}, \mu_{P_L}, \mu_E, \mu_{VD}) = (0.7703, 0.7292, 0.7216, 0.7154)$$

The result shows that a satisfactory decision is achieved here from the view point of balancing the decision powers of the DMs on the basis of order of hierarchy adopted in the decision making situation (Fig. 5).

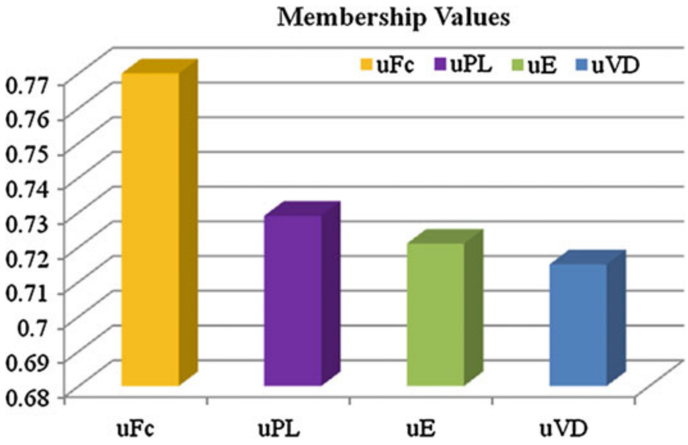


Fig. 5 Membership values of the four objectives in two different levels

8 Conclusions

In this chapter, an GA based FGP approach for modeling and solving optimal real and reactive power flow management problem in the framework of MOBLP in a hierarchical decision structure is presented.

The main advantage of the proposed approach is that the BLP formulation of the problem within the framework of multiobjective decision making model leads to take individual decisions regarding optimization of objectives on the basis of hierarchy assigned to them.

Again, computational load and approximation error inherent to conventional linearization approaches can be avoided here with the use of the GA based solution method.

In the framework of the proposed model, consideration of other objectives and environmental constraints may be taken into account and the possible aspects of formulating MLPP within hierarchical decision structure for power plant operations may be a problem in future study.

Further, sensitivity analysis with assignment of objectives to DMs along with controlling of decision variables at different hierarchical levels on the basis of needs in the decision horizon may be a open problem in future.

Finally, it is hoped that the solution approach presented here may lead to future research for proper planning of electric power generation and dispatch.

References

1. Abido, M.A.: A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. Electr. Power Syst. Res. **65**(1), 71–81 (2003)
2. Abido, M.A.: Multiobjective evolutionary algorithms for electric power dispatch problem. IEEE Trans. Evol. Comput. **10**(3), 315–329 (2006)
3. Abou El Ela, A.A., Abido, M.A., Spea, S.R.: Differential evolution algorithm for optimal reactive power dispatch. Electr. Power Syst. Res. **81**, 458–468 (2011)
4. AlRashidi, M.R., El-Hawary, M.E.: Pareto fronts of the emission-economic dispatch under different loading conditions. Int. J. Electr. Electron. Eng. **2**(10), 596–599 (2008)
5. Azar, A.T.: Fuzzy Systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
6. Azar, A.T.: Overview of type-2 fuzzy logic systems. Int. J. Fuzzy Syst. Appl. **2**(4), 1–28 (2012)
7. Bansilal, A., Thukaram, D., Parthasarathy, K.: Optimal reactive power dispatch algorithm for voltage stability improvement. Int. J. Electr. Power Energy Syst. **18**(7), 461–468 (1996)
8. Basu, M.: An interactive fuzzy satisfying-based simulated annealing technique for economic emission load dispatch with nonsmooth fuel cost and emission level functions. Electr. Power Compon. Syst. **32**(2), 163–173 (2004)
9. Basu, M.: Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. Int. J. Electr. Power Energy Syst. **30**(2), 140–149 (2008)
10. Burton, R.M.: The multilevel approach to organizational issues of the firm—a critical review. Omega **5**(4), 395–414 (1977)
11. Cadogan, J.B., Eisenberg, L.: Sulfur oxide emissions management for electric power systems. IEEE Trans. Power Appar. Syst. **96**(2), 393–401 (1977)
12. Candler, W., Townsley, R.: A linear two level programming problem. Comput. Oper. Res. **9** (1), 59–76 (1982)
13. Chen, P.H., Chang, H.C.: Large-scale economic dispatch by genetic algorithm. IEEE Trans. Power Syst. **10**(4), 1919–1926 (1995)
14. Chowdhury, B.H., Rahman, S.: A review of recent advances in economic dispatch. IEEE Trans. Power Syst. **5**(4), 1248–1259 (1990)
15. Das, B., Patvardhan, C.: A new hybrid evolutionary strategy for reactive power dispatch. Electr. Power Res. **65**, 83–90 (2003)
16. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, US (2002)
17. Deeb, N., Shahidehpour, S.M.: Linear reactive power optimization in a large power network using the decomposition approach. IEEE Trans. Power Syst. **5**(2), 428–438 (1990)
18. Dhillon, J.S., Parti, S.C., Kothari, D.P.: Stochastic economic emission load dispatch. Electr. Power Syst. Res. **26**(3), 179–186 (1993)
19. Dommel, H.W., Tinney, W.F.: Optimal power flow solutions. IEEE Trans. Power Appar. Syst. **87**(10), 1866–1876 (1968)
20. El-Keib, A.A., Ma, H., Hart, J.L.: Economic dispatch in view of the clean air act of 1990. IEEE Trans. Power Syst. **9**(2), 972–978 (1994)
21. Farag, A., Al-Baiyat, S., Cheng, T.C.: Economic load dispatch multiobjective optimization procedures using linear programming techniques. IEEE Trans. Power Syst. **10**(2), 731–738 (1995)
22. Fernandes, R.A., Lange, F., Burchett, R.C., Happ, H.H., Wirgau, K.A.: Large scale reactive power planning. IEEE Trans Power Apparatus Syst. **102**(5), 1083–1088 (1983)
23. Gen, M., Ida, K., Lee, J., Kim, J.: Fuzzy nonlinear goal programming using genetic algorithm. Comput. Ind. Eng. **33**(1–2), 39–42 (1997)
24. Gent, M.R., Lament, JWm: Minimum-emission dispatch. IEEE Trans. Power Appar. Syst. **90** (6), 2650–2660 (1971)
25. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)

26. Gong, D., Zhang, Y., Qi, C.: Environmental/economic power dispatch using a hybrid multi-objective optimization algorithm. Electr. Power Energy Syst. **32**, 607–614 (2010)
27. Granville, S.: Optimal reactive power dispatch through interior point methods. IEEE Trans. Power Syst. **9**(1), 98–105 (1994)
28. Happ, H.H.: Optimal power dispatch—a comprehensive survey. IEEE Trans. Power Appar. Syst. **96**(3), 841–854 (1977)
29. Hejazi, S.R., Memariani, A., Jahanshahloo, G., Sepehri, M.M.: Linear bilevel programming solution by genetic algorithm. Comput. Oper. Res. **2**(29), 1913–1925 (2002)
30. Hobbs, B.F.: Emission dispatch under the underutilization provision of the 1990 U.S. Clean air act amendments: models and analysis. IEEE Trans. Power Syst. **8**(1), 177–183 (1993)
31. Holland, J.H.: Genetic algorithms and optimal allocation of trials. SIAM J. Comput. **2**, 88–105 (1973)
32. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)
33. Huang, C.M., Yang, H.T., Huang, C.L.: Bi-objective power dispatch using fuzzy satisfaction-maximizing decision approach. IEEE Trans. Power Syst. **12**(4), 1715–1721 (1997)
34. Iba, K.: Reactive power optimization by genetic algorithms. IEEE Trans. Power Syst. **9**(2), 685–692 (1994)
35. Ignizio, J.P.: Goal Programming and Extensions. D. C Health, Lexington (1976)
36. Jizhong, Z.: Optimization of Power System Operation. Wiley, Hoboken (2009)
37. Lai, Y.J.: Hierarchical optimization: a satisfactory solution. Fuzzy Sets Syst. **77**(3), 321–335 (1996)
38. Jolissaint, C.H., Arvanitidis, N.V., Luenberger, D.G.: Decomposition of real and reactive power flows: a method suited for on-line applications. IEEE Trans. Power Appar. Syst. **91**(2), 661–670 (1972)
39. Lee, K.Y., Park, Y.M., Ortiz, J.L.: A united approach to optimal real and reactive power dispatch. IEEE Trans Power Apparatus Syst. **104**(5), 1147–1153 (1985)
40. Lee, K.Y., Yang, F.F.: Optimal reactive power planning using evolutionary programming: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm and linear programming. IEEE Trans. Power Syst. **13**(1), 101–108 (1998)
41. Mangoli, M.K., Lee, K.Y.: Optimal real and reactive power control using linear programming. Electr. Power Syst. Res. **26**, 1–10 (1993)
42. Momoh, J.A., El-Hawary, M.E., Adapa, R.: A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods. IEEE Trans. Power Syst. **14**(1), 105–111 (1999)
43. Mathieu, R., Pittard, L., Anandalingam, G.: Genetic algorithm based approach to bilevel linear programming. Oper. Res. **28**(1), 1–21 (1994)
44. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs', 3rd edn. Spriger, Berlin (1996)
45. Nanda, J., Kothari, D.P., Lingamurthy, K.S.: Economic-emission load dispatch through goal programming techniques. IEEE Trans. Energy Convers. **3**(1), 26–32 (1988)
46. Nishizaki, I., Sakawa, M.: Computational methods through genetic algorithms for obtaining Stackelberg solutions to two-level mixed zero-one programming problems. Cybernetics and Systems **31**(2), 203–221 (2000)
47. Pal, B.B., Moitra, B.N.: A goal programming procedure for solving problems with multiple fuzzy goals using dynamic programming. Eur. J. Oper. Res. **144**(3), 480–491 (2003)
48. Pal, B.B., Moitra, B.N., Maulik, U.: A goal programming procedure for fuzzy multiobjective linear fractional programming problem. Fuzzy Sets Syst. **139**(2), 395–405 (2003)
49. Pal, B.B., Chakraborti, D.: Using genetic algorithm for solving quadratic bilevel programming problems via fuzzy goal programming. Int. J. Appl. Manage. Sci. **5**(2), 172–195 (2013)
50. Peschon, J., Piercy, D.S., Tinney, W.F., Tveit, O.J., Cuenod, M.: Optimum control of reactive power flow. IEEE Trans. Power Appar. Syst. **87**(1), 40–48 (1968)
51. Quintana, V.H., Santos-Nieto, M.: Reactive-power dispatch by successive quadratic programming. IEEE Trans. Energy Convers. **4**(3), 425–435 (1989)

52. Romero, C.: Handbook of Critical Issues in Goal Programming. Pergamon Press, Oxford (1991)
53. Sachdeva, S.S., Billinton, R.: Optimum network VAR planning by nonlinear programming. IEEE Trans. Power Apparatus Syst. **92**, 1217–1225 (1973)
54. Sakawa, M.: Genetic Algorithms and Fuzzy Multiobjective Optimization. Kluwer Academic Publishers, Boston (2001)
55. Sakawa, M., Nishizaki, I.: Interactive fuzzy programming for decentralized two-level linear programming problems. Fuzzy Sets Syst. **125**(3), 301–315 (2002)
56. Sakawa, M., Nishizaki, I.: Interactive fuzzy programming for two-level Nonconvex programming problems with fuzzy parameters through genetic algorithms. Fuzzy Sets Syst. **127**(2), 185–197 (2002)
57. Shih, H.S., Lee, S.: Compensatory fuzzy multiple level decision making. Fuzzy Sets Syst. **114** (1), 71–87 (2000)
58. Simon, H.A.: Administrative Behavior. Fress Press, New York (1957)
59. Sonmez, Y.: Multi-objective environmental/economic dispatch solution with penalty factor using artificial bee colony algorithm. Sci. Res. Essays **6**(13), 2824–2831 (2011)
60. Srinivasan, D., Tettamanzi, A.G.B.: An evolutionary algorithm for evaluation of emission compliance options in view of the clean air act amendments. IEEE Trans. Power Syst. **12**(1), 336–341 (1997)
61. Sullivan, R.L., Hackett, D.F.: Air quality control using a minimum pollution-dispatching algorithm. Environ. Sci. Technol. **7**(11), 1019–1022 (1973)
62. Talaq, J.H., El-Hawary, F., El-Hawary, M.E.: A summary of environmental/ economic dispatch algorithms. IEEE Trans. Power Syst. **9**(3), 1508–1516 (1994)
63. Tsuji, A.: Optimal fuel mix and load dispatching under environmental constraints. IEEE Trans. Power Appar. Syst. **PAS 100**(5), 2357–2364 (1981)
64. Vanitha, M., Thanushkodi, K.: An efficient technique for solving the economic dispatch problem using biogeography algorithm. Eur. J. Sci. Res. **5**(2), 165–172 (2011)
65. Vlachogiannis, J.G., Lee, K.Y.: Quantum-inspired evolutionary algorithm for real and reactive power dispatch. IEEE Trans. Power Syst. **23**(4), 1627–1636 (2008)
66. Wang, L.F., Singh, C.: Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. Electr. Power Syst. Res. **77**(12), 1654–1664 (2007)
67. Wu, Q.H., Ma, J.T.: Power system optimal reactive power dispatch using evolutionary programming. IEEE Trans. Power Syst. **10**(3), 1243–1249 (1995)
68. Yokoyama, R., Bae, S.H., Morita, T., Sasaki, H.: Multiobjective optimal generation dispatch based on probability security criteria. IEEE Trans. Power Syst. **3**(1), 317–324 (1988)
69. Zadeh, L.A.: Fuzzy Sets. Inf. Control **8**(3), 338–353 (1965)
70. Zahavi, J., Eisenberg, L.: Economic-environmental power dispatch. IEEE Trans. Syst. Man Cybern. SMC **5**(5), 485–489 (1975)
71. Zhang, G., Zhang, G., Gao, Y., Lu, J.: Competitive strategic bidding optimization in electricity markets using bilevel programming and swarm technique. IEEE Trans. Industr. Electron. **58** (6), 2138–2146 (2011)
72. Zheng, D.W., Gen, M., Ida, K.: Evolution program for nonlinear goal programming. Comput. Ind. Eng. **31**(3/4), 907–911 (1996)
73. Zimmermann, H.-J.: Fuzzy Sets Decision Making and Expert Systems. Kluwer Academic Publisher, Boston (1987)
74. Zimmermann, H.-J.: Fuzzy Set Theory and Its Applications, 2nd Revised edn. Kluwer Academic Publishers, Boston (1991)

A Monitoring-Maintenance Approach Based on Fuzzy Petri Nets in Manufacturing Systems with Time Constraints

Anis Mhalla and Mohamed Benrejeb

Abstract Maintenance and its integration with control and monitoring systems enable the improvement of systems functioning, regarding availability, efficiency, productivity and quality. This paper proposes a monitoring-maintenance approach based on fuzzy Petri Nets (PN's) for manufacturing job-shops with time constraints. In such systems, operation times are included between a minimum and a maximum value. In this context, we propose a new fuzzy Petri net called Fuzzy Petri Net for maintenance (FPNM). This tool is able to identify and select maintenance activities of a discrete event system with time constraints, using a temporal fuzzy approach. The maintenance module is consists of P-time PNs and fault tree. The first is used for modelling of normal behaviour of the system by temporal spectrum of the marking. The second model corresponds to diagnosis activities. Finally, to illustrate the effectiveness and accuracy of proposed maintenance approach, two industrial examples are depicted.

Keywords Fuzzy numbers · P-time PN · Fault tree · Alpha-cut · FPNM · Recovery

1 Introduction

The demands for products with higher quality and competitive prices have led to the development of complex manufacturing systems. A consequence is that the number of failures tends to increase as well as the time required to locate and repair them. The occurrence of failures during nominal operation can deeply modify the

---

A. Mhalla (✉) · M. Benrejeb
Unité de Recherche LARA Automatique, National School of Engineering of Tunis,
BP 37, 1002 Tunis, Belvédère, Tunisia
e-mail: anis.mhalla@enim.rnu.tn

M. Benrejeb
e-mail: mohamed.benrejeb@enit.rnu.tn

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_9

flexible manufacturing systems (FMS's) performance or its availability. Thus it is imperative to implement a maintenance strategy allocated to the FMS's.

There have been many works proposed for maintenance of FMS. Among many related researches, Yang and Liu [26] use Petri net coupled with parameter trend and fault tree analysis to perform early failure detection and isolation for Preventive maintenance. A fault diagnosis system for district heating is used as an example to demonstrate the proposed Petri net method. Adamyan and He [1] presented a very useful method that utilized Petri net techniques for modeling the system dynamics to identify possible failure sequences. With their method the reliability and safety of the manufacturing systems were also assessed by the reachability trees of markings. Consequently, the method can overcome the limitations of sequential failure logic (SFL) which is used for assessing the probability of failure.

Based on assembly matrix and Petri nets, a virtual disassembly Petri nets (VDPN) model including maintenance resource and maintenance cost of FMS was proposed by Zhang et al. [27]. Cassady et al. [7] introduce the concept of selective maintenance and develop a generalized modeling framework for optimizing selective maintenance decisions. This framework is applicable to systems that perform identical missions and have limited time between missions for maintenance.

Laures et al. [22] propose a control-monitoring-maintenance architecture (CMM) for FMS based on Petri nets with objects (PNO), where stochastic rates are associated to the modeling of maintenance planning. This framework is based on a modular and hierarchic model structured in CMM modules. The integration is based on a development methodology in which the maintenance aspects and policies are taken into account from the conception (modeling) stage.

Another approach for maintenance scheduling of a railway system in a finite planning horizon is proposed in [28]. In this approach, a Petri-net model was used to describe the stochastic and dynamic behavior of the component deterioration and maintenance process. It evaluates the total cost including possession cost, window cost and penalty cost as the objective which can be calculated by the simulation of the Petri-net. Other results on maintenance of FMS's are reported in [8, 12, 13].

In this paper, we propose a new maintenance approach based on the study of effective sojourn time of the token in places and the evaluation of the "failure probability of the top event", in manufacturing systems with staying time constraints. In the category of the workshops concerned by this paper, the operations have temporal constraints which must be imperatively respected. The violation of these constraints can affect the health of the consumers. Thus, the detection of a constraint violation must automatically cause the stop of the production. Maintenance and its integration with control and monitoring systems, enable the improvement of manufacturing systems, regarding availability, efficiency, productivity and quality [22]. Thus, it is possible to implement corrective and preventive actions in manufacturing systems with time constrains in order to make repairs and servicing easier over the process elements, as well as a better control provision of tools and repair parts.

This paper is organised as follows. The second section begins by presenting the formal definition of P-TPN as a modelling tool and summarizes the classes of uncertainties in manufacturing workshops with time constraints. Section 3, introduce the fuzzy probabilistic approach to evaluate failure probability of the top event, when there is an uncertainty about the components failure probabilities. Afterward, the problem of maintenance of manufacturing systems is tackled. An original recovery approach based on PN's, is presented.

In Sect. 5, two academic examples (workshops with/without assembling tasks) are then used to illustrate the different steps of the proposed approach. Finally, a conclusion is presented with some perspectives.

2 Representation of Uncertainty in Manufacturing Workshops

2.1 P-Time Petri Nets

From the modelling point of view, P-TPNs were introduced in 1996 in order to model Dynamic Discrete Event System (DDES) including sojourn time constraints.

Definition 1 [14] The formal definition of a P-TPN is given by a pair $\langle R; I \rangle$ where:

- R is a marked Petri net,
- $I : P \rightarrow Q^+ \times (Q^+ \cup \{+\infty\})$

$$p_i \rightarrow IS_i = [a_i,\ b_i] \text{ with } 0 \le a_i \le b_i.$$

IS_i defines the static interval of staying time of a mark in the place p_i belonging to the set of places P (Q^+ is the set of positive rational numbers). A mark in the place p_i is taken into account in transition validation when it has stayed in p_i at least a duration a_i and no longer than b_i. After the duration b_i the token will be dead.

In manufacturing job-shops with time constraints, for each operation is associated a time Interval ($[a_i, b_i]$ with u.t: unit time). Its lower bound indicates the minimum time needed to execute the operation and the upper bound sets the maximum time not to exceed in order to avoid the deterioration of the product quality. Consequently P-TPNs have the capability of modelling time intervals and deducing a set of scenarios, when time constraints are violated.

2.2 Uncertainty in Manufacturing Workshops

The production is subject to many uncertainties arising from the processes, the operators or the variations of quality of the products. A production is seldom perfectly repetitive, due to uncertainties on the process. However, a regular production is required in order to maintain product quality.

All authors, who treated uncertainties, studied mainly two disturbances: disturbances on the equipment and more particularly the breakdowns machine or the disturbances concerning work and more particularly the change in the operational durations [24]. For all these reasons, a function of possibilities, representing uncertainty over the effective residence time (q_i) of a token in a place p_i, is proposed. This function makes it possible to highlight zones of certainty for an operational duration and helps the human agent (or supervisor) in charge of detecting failures and deciding reconfiguration/repair actions [18].

2.2.1 Graphical Representation of Effective Sojourn Time Uncertainty

In order to quantify to a set of possible sojourn time of the token in the place p_i, a fuzzy set A, representing the uncertainty on the effective sojourn time of the token in the place p_i (q_i) is proposed (Fig. 1).

This quantification allows us to define a measure of the possibility with which the sojourn time q_i, is verified. These results, Fig. 1, make it possible to highlight zones of certainty for operation durations; a high value of effective sojourn time can guarantee a normal behaviour of monitored system. Instead, a low value implies the possibility of detecting of failure symptom (behavioural deviation).

Based on fuzzy model, Fig. 1, all system scenarios are developed. The scenarios consider all possible deviations. Deviations can occur due to the failure of

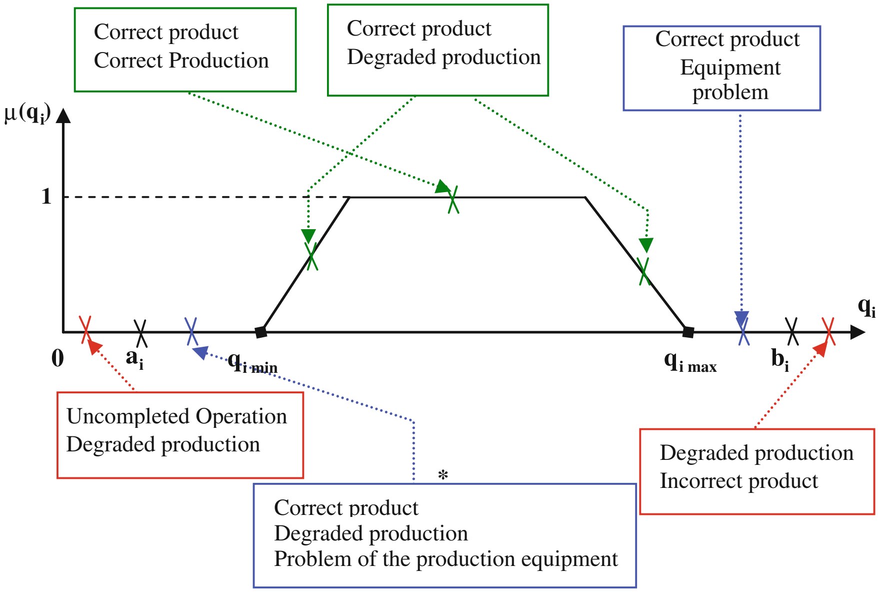


Fig. 1 Function of possibility associated with an effective sojourn time (q_i)

components. Then from fuzzy model, we deduce a set of scenarios (events sequences) bringing the system to erroneous situations (failure).

2.2.2 Control Objects Application in the P-TPN

The control objects can be connected to the P-TPN model of the manufacturing system, and can respectively be applied to perform checking and validation of the adequateness and correctness of all operations that are introduced in the system.

Two general types of control objects can be utilized for the purpose:

Watch-Dogs Objects

These control objects are able to restrict the maximum and the minimum time periods, that are allowed for the execution of the particular operations. In cases, if the time restrictions are violated, the system is capable to generate an immediate reaction (similar to the alarm cases).

Time-Out Objects

These control objects represent particular kind of watch-dogs that restrict only the maximal time periods, allowed for a particular operation, and react in the same way as the watch-dogs.

In the P-TPN, watch-dog model can be connected to the places. In that model the beginning and the end of all operation, which needs strict control of their execution time periods. Thus, in manufacturing system with time constraints, any detection of a constraint violation (possible defects) can be modelled by specific mechanisms named watch dogs.

2.2.3 Constraints Violation and Recovery Task

In manufacturing workshops with time constraints, the fuzzy model associated to effective sojourn time (q_i), monitors the system evolutions through the time durations verification (operating or transfer durations for example) [19]. These durations represent interval constraints. When the interval constraints are exceeded, there is an error.

An error is defined as a discrepancy between an observed or measured value and the true or theoretically correct value or condition [26]. In our study, an error means a gap between measured and computed time intervals by the scheduling task.

Based on the above statements, an error is sometimes referred to as an incipient failure [26]. Therefore maintenance action is taken when the system is still in an error condition, i.e. within acceptable deviation and before failure occurs. Thus, this study employs uncertainty of sojourn time in order to perform early failure detection.

3 Estimation of Failure Probability by Fuzzy Fault Tree Analysis

3.1 Preliminary Definitions

Definition 1 [11] A fault tree FT is a directed acyclic graph defined by the tuple $\{E_i, G_i, D_i, TOP_i\}$. The union of the sets G_i (logical gates) and E_i (events) represents the nodes of the graph; D_i is a set of directed edges, each of which can only connect an event to the input of a logical gate or the output of a logical gate to an event.

A top event TOP_i is an event of the fault tree FT_i that is not the input of any logic gate, i.e. there are no edges that come out of the top event. The nodes of a fault tree are connected through logical gates, in this paper; we consider only static fault trees, i.e. fault trees in which the time variable does not appear. Therefore, only the AND and the OR gate will be treated in this paper.

Definition 2 [11] Let us suppose AND_i is an AND gate with n inputs IN_kAND_i, $1 < k < n$ and output $OUTAND_i$. Let $P_{in}(k, i)$ be the probability associated with the input IN_kAND_i and $P_{OUT}AND_i$ be the probability associated with the output of AND_i. If the inputs to the AND gate is mutually independent, the probability associated with the output can be calculated as follows:

$$P_{OUT}AND_i = \prod_{k=1}^{n} P_{in}(k,\ i) \tag{1}$$

Definition 3 [11] Let us suppose OR_i is an OR gate with n inputs IN_kOR_i, $1 < k<n$ and output $OUTOR_i$. Let $P_{in}(k, i)$ be the probability associated with the input IN_kOR_i and $P_{OUT}OR_i$ be the probability associated with the output of OR_i.

If the inputs to the OR gate all mutually exclusive, the output can be calculated as follows:

$$P_{OUT}OR_i = 1 - \prod_{k=1}^{n} (1 - P_{in}(k,\ i)) \tag{2}$$

3.2 Fuzzy Approach for Uncertainty Analysis

The fuzzy probabilistic approach aims to quantitatively evaluate the reliability of manufacturing workshops with time constraints. But, as mentioned previously, studies are under uncertainty. The goal of the paper is to take into account these uncertainties in the evaluation. So, we investigate the use of the fuzzy set theory to determine the probability of the top event of the fault tree associated to workshops with time constraint.

3.2.1 Fuzzy Numbers

Let x be a continuous variable restricted to a distribution function $\mu(x)$, which satisfy the following assumptions [23]:

- $\mu(x)$ is a piecewise continuous,
- $\mu(x)$ is a convex fuzzy set,
- $\mu(x)$ is a normal fuzzy set.

A fuzzy set which satisfies these requirements is called a fuzzy number.

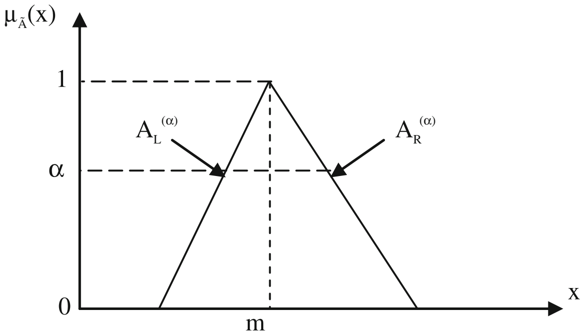
For any fuzzy number $\tilde{A}$ which has the membership function $\mu_{\tilde{A}}(x)$, an interval bounded by two points at each α-level ($0 \leq \alpha \leq 1$) can be obtained using the α-cut method [17]. The symbols $A_L^{(\alpha)}$ and $A_R^{(\alpha)}$ have been used in this paper to represent the $\mu_{\tilde{A}}(x)$ left-end-point and right-end-point of this interval.

As it is shown in Fig. 2, we can express a fuzzy number, using the following form [23]:

$$\tilde{A} \to [A_L{}^{(\alpha)},\ A_R{}^{(\alpha)}] \text{ with } \quad 0 \leq \alpha \leq 1$$

For each α-level of the fuzzy number which represents a probability, the model is run to determine the minimum and maximum possible values of the output. This information is then directly used to construct the corresponding membership function of the output.

Fig. 2 Bounds points for α-level set interval of μ_A (x) [23]



3.2.2 Fuzzy Probabilities

A fuzzy probability is represented by a fuzzy number between 0 and 1 assigned to the probability of an event occurrence [9, 17, 25].

Our goal is to use fuzzy probabilities to describe occurrence probabilities of events. To this end, we follow the standard approach proposed by Buckley to describe the probabilities of various unions and intersections of these events occurrences.

3.2.3 Buckley Approach

An extension of traditional approaches to take account of vagueness is proposed by Buckley [6] and Buckley and Eslami [5]. The Buckley approach, associate to each input variables a fuzzy number and combine them sequentially by using the concept of α-cut which reduces the problem to a calculation interval.

Let us consider two fuzzy numbers $\tilde{X}$ and $\tilde{Y}$, respectively represented by the two intervals $[\mathrm{X}_{\mathrm{L}}^{(\alpha)}, \mathrm{X}_{\mathrm{R}}^{(\alpha)}]$ and $[\mathrm{Y}_{\mathrm{L}}^{(\alpha)}, \mathrm{Y}_{\mathrm{R}}^{(\alpha)}]$. Arithmetic operations applied to intervals give the following expressions [13]:

$$\tilde{\mathrm{Z}} = \tilde{\mathrm{X}} + \tilde{\mathrm{Y}} \rightarrow [\mathrm{Z}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{Z}^{(\alpha)}{}_{\mathrm{R}}] = [\mathrm{X}^{(\alpha)}{}_{\mathrm{L}} + \mathrm{Y}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{R}} + \mathrm{Y}^{(\alpha)}{}_{\mathrm{R}}] \tag{3}$$

$$\tilde{\mathrm{Z}} = \tilde{\mathrm{X}} \cdot \tilde{\mathrm{Y}} \rightarrow [\mathrm{Z}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{Z}^{(\alpha)}{}_{\mathrm{R}}] \tag{4}$$

$$\text{with} : \begin{cases} \tilde{\mathrm{Z}}_{\mathrm{L}}^{(\alpha)} = \min(\mathrm{X}^{(\alpha)}{}_{\mathrm{L}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{R}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{L}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{R}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{R}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{R}}) \\ \tilde{Z}_{R}^{(\alpha)} = \max(\mathrm{X}^{(\alpha)}{}_{\mathrm{L}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{R}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{L}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{L}} \cdot \mathrm{Y}^{(\alpha)}{}_{\mathrm{R}}, \mathrm{X}^{(\alpha)}{}_{\mathrm{R}} \cdot \mathrm{Y}^{(\alpha)}{}_{R}) \end{cases}$$

3.2.4 Fault Tree Analysis and Maintenance Task

Reliability and life are two major elements of maintenance tasks. Reliability theory can also assists maintenance engineers in judging the operational status of equipment and in developing safe response measures to prevent any accidents during shutdown procedures [15]. Such the FTA provides a basis for further maintenance of manufacturing systems with time constraints, there by, enabling engineers to conduct additional tests to determine a proper reliability distribution model for further analysis and application.

According to the failure records of well-documented manufacturing systems, maintenance tasks are generally categorized as adjustment, repair, and replacement [15]. While failure distribution characteristics are analyzed using fault tree analysis, failure distribution modes and Weibull distributions, are incorporated into a system reliability model and then tested and analyzed to establish a proper reliability distribution model.

4 Fuzzy Petri Nets for Maintenance (FPNM)

In manufacturing system, failure will occur if the degradation level exceeds the permissible value. Therefore, maintenance is defined as a strategy to maintain available or operational conditions of a facility using all possible methods and means, or to restore functions from trouble and failures [22].

Much of development works has been undertaken in certain of the maintenance fields. Recovery tools have been researched [15] and their application to failure prevention is well reviewed. The proposed recovery tool is inspired of the research of Minca [21].

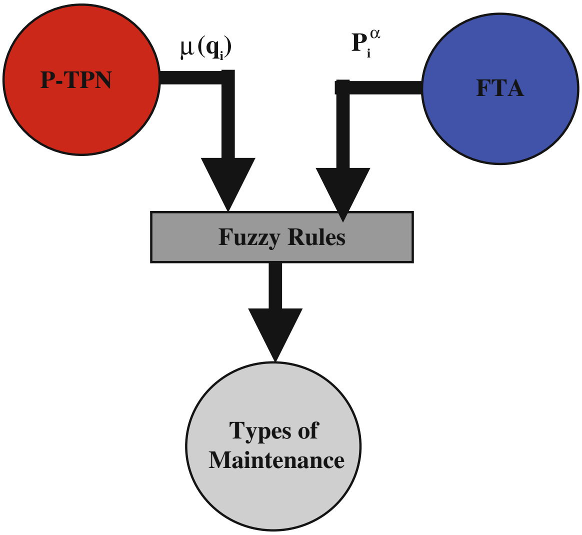
To model the recovery functions, a definition of a fuzzy PN model able to integrate uncertainty on sojourn time (q_i) and fuzzy probabilities of the monitored system (P_i), related to a base of fuzzy logic rules, is given, Fig. 3.

4.1 Definition of FPNM

The fuzzy Petri net for maintenance (FPNM) is defined as being the n-uplet: $\langle P, T, \mathrm{Pr}, Q, RA, F, \Psi, \Omega, \Delta, M_0 \rangle$ with:

$P = p^x \cup p^y$	the finite set of input p^x and output p^y places;
$T = \{t_1, t_2, \ldots, t_n\}$	a collection of transitions. A transition t_i is specialized in inference/aggregation operations of logic rules;

Fig. 3 FPNM structure



$\mathrm{Pr} = \bigcup_{e=1}^{z} \mathrm{Pr}_e$	the finite set of the input variable “fuzzy probability”;
$Q = \bigcup_{f=1}^{r} q_f$	subsets of input variables “sojourn time”;
$RA = \bigcup_{g=1}^{s} ra_g$	subsets of output variables “recovery action”;

Pr (resp Q) and RA are subsets of variables that are respectively in the antecedence and in the consequence of the fuzzy rules F_w;

$F = \bigcup_{w=1}^{\alpha} F_w$	$F_w = \mathrm{Pr} \cup Q \rightarrow RA$: the fuzzy logic rules set
$\Psi = (\Psi_{11}, \Psi_{12}, \ldots, \Psi_{z1}, \Psi_{z2}, \ldots, \Psi_{ze})$	the finite set of membership functions, defined on the universe [0, 1] of the input variables “fuzzy probability”, Pr = (Pr_1, Pr_2, …, Pr_z).“e” represents the number of input variables Pr;
$\Omega = (\Omega_{11}, \Omega_{12}, \ldots, \Omega_{f1}, \Omega_{f2}, \ldots, \Omega_{fr})$	the finite set of membership functions, definite on the universe [0, 1] of the second input variable “sojourn time”. “r” is the number of input variables “sojourn time”;
$\Delta = (\Delta_{11}, \Delta_{12}, \ldots, \Delta_{g1}, \Delta_{g2}, \ldots, \Delta_{gs})$	the finite set of membership functions, definite on the universe [0, 1] of the output variable “recovery action”. “s” is the number of output variables;
M_0	the initial marking of the input places $p_i \in P^x$

Each input or output place of the FPNM is associated to a fuzzy description. For the input places, we describe the marking variable of the place, whereas for the output places we describe recovery action.

In FPNM, each base of logic rules “F” represents the fuzzy implications describing the knowledge base of the expert. Each implication respects the “if-then” model represents the logical dependence of variable Pr (resp. Q and RA} associated to the fuzzy sets Ψ (resp. Ω and Δ).

The proposed FPNM is considered as an adaptive technique dedicated to the recovering of manufacturing systems with time constraints. This model has a double interface, one with the modelling model (based on P-time PN) and the second one with the diagnosis model (fuzzy Fault Tree). To demonstrate the effectiveness of our proposed methodology we present two maintenance realistic examples.

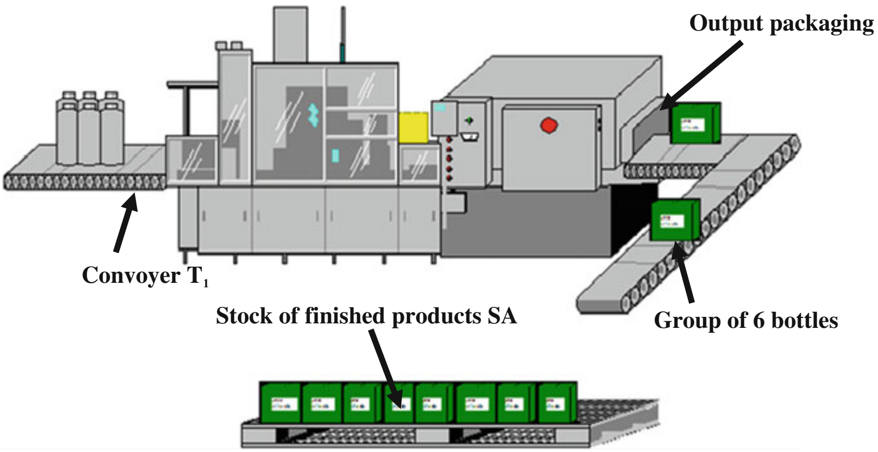


Fig. 4 Packaging machine

5 Illustrative Examples

5.1 Packaging Unit (Job Shop with Assembling Tasks)

5.1.1 Presentation of Packaging Unit

For simplicity, we disregard the nature of the precise operations performed in the packaging unit; therefore we represent a simplified model of the unit.

Figure 4, shows a milk packaging unit: to packing the products (bottles of 1,000 ml), bottles are placed on the conveyor T_1 to supply the packaging machine (M), where they will be wrapped by welding in a group of 6. The finished products are deposited on the output conveyor towards the stock of finished products SA.

5.1.2 Modeling of Packaging Unit

Figure 5, shows a P-time Petri net (G) modeling the packaging machine. Three fuzzy sets, representing the uncertainty on the effective sojourn time of the token in the places p_1, p_2 and p_8, are proposed (Fig. 5). The obtained membership's functions are used to study the maintenance of the machine (M). As the sojourn times in places have not the same functional signification when they are included in the sequential process of a product or when they are associated to a free resource, a decomposition of the Petri net model into two sets is made using [16], Fig. 5, where:

- R_U is the set of places representing the used machines,
- $Trans_C$ is the set of places representing the loaded transport resources.

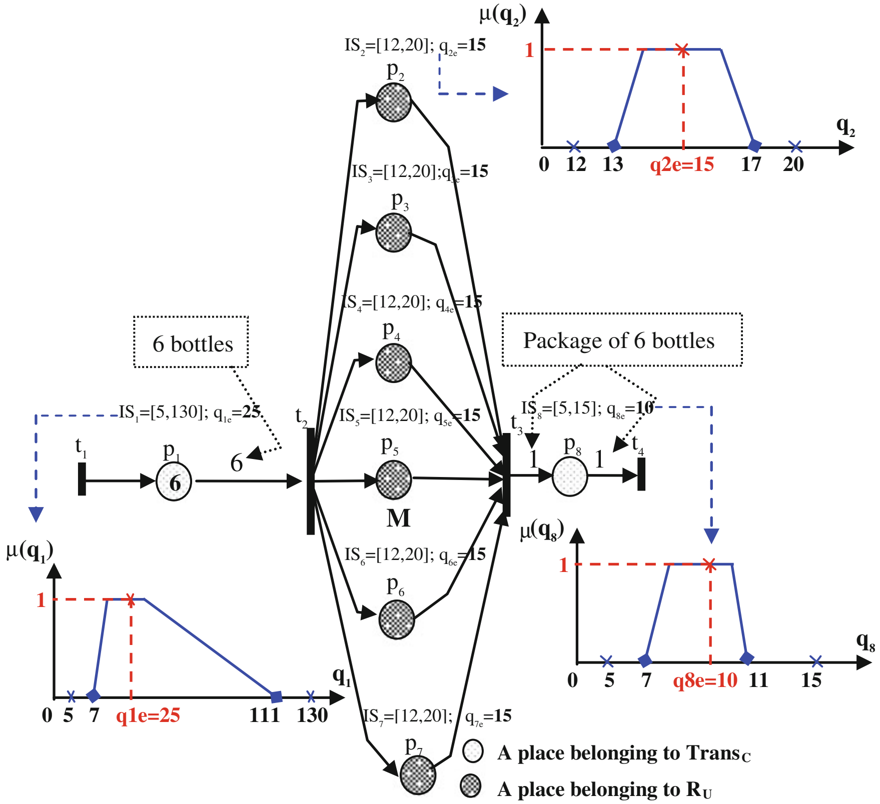


Fig. 5 Packaging machine modeled by a P-time Petri net [19]

In milk manufacturing workshop, the operations have temporal constraints which must be imperatively respected. The violation of these constraints can affect the health of the consumers and can induce some catastrophic consequences (inconsumable product, burnt milk …) [20]. Therefore, the considered recovering approach uses the additional information provided by the knowledge of interval constraints and by the detection a failure symptom.

Let us suppose that we want to monitor the duration of packaging of bottles. According to P-TPN, Fig. 5, the minimum time granted to the operation is 12 u.t, whereas the maximum time is 20 u.t (IS = [12, 20]; q_e = 15). A delay of packaging operation may involve:

- A technical failure of the production tool (conveyor problem for example) requiring to generate a maintenance action,
- The production cycle of a milk bottle can be delayed; in fact a delay can imply the propagation of the failure symptom and can induce some catastrophic consequences on the functioning of the system.

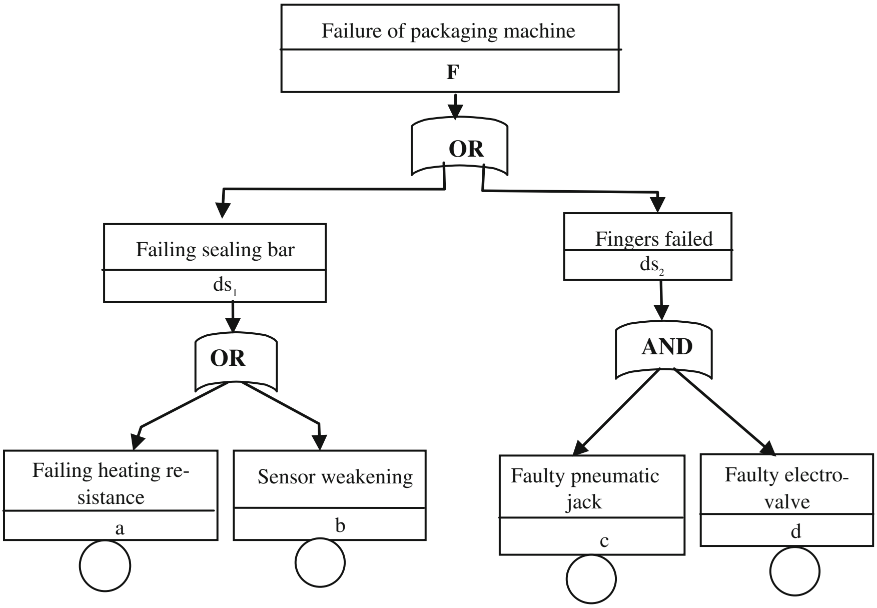


Fig. 6 Fault tree of packaging machine

5.1.3 Diagnosis of Packaging Machine

When a constraint is violated a diagnosis text is generated. The diagnosis text determines failed states (deviations from the normal function) of the packaging machine and its subsystem (failure of sealing bar, failure of fingers …).

When a symptom is claimed, it is imperative to localize the failure by using fault tree as a modeling tool, Fig. 6. The logical expression of top event (F) of the fault tree is: $F = ds_1 + ds_2 = (a + b) + (c \times d)$.

5.1.4 Fuzzy Probabilistic Approach

The Fault Tree analysis (FTA) is based on the fuzzy set theory [2, 3, 4]. So, we can allocate a degree of uncertainty to each value of the failure probability. Thus, according to Eqs. (1) and (2), the fuzzy probability of a system failure (top event occurrence) is determined from the fuzzy probabilities of components failure.

The parameter a_i is the lower bound, the parameter m_i is the modal value, and the parameter b_i is the upper bound for each fuzzy probability of components failure. These parameters are given in Table 1.

Figure 7, provides the representation of computed fuzzy probability associated to the failure F, ds_1 and ds_2. The fuzzy failure probability of the top event (F) is given below:

Table 1 Parameters of fuzzy probabilities

Basic event	m	a	b
a	0.0015	0.00111	0.0023
b	0.0014	0.00121	0.0021
c	0.00413	0.0032	0.0048
d	0.0032	0.0025	0.0033

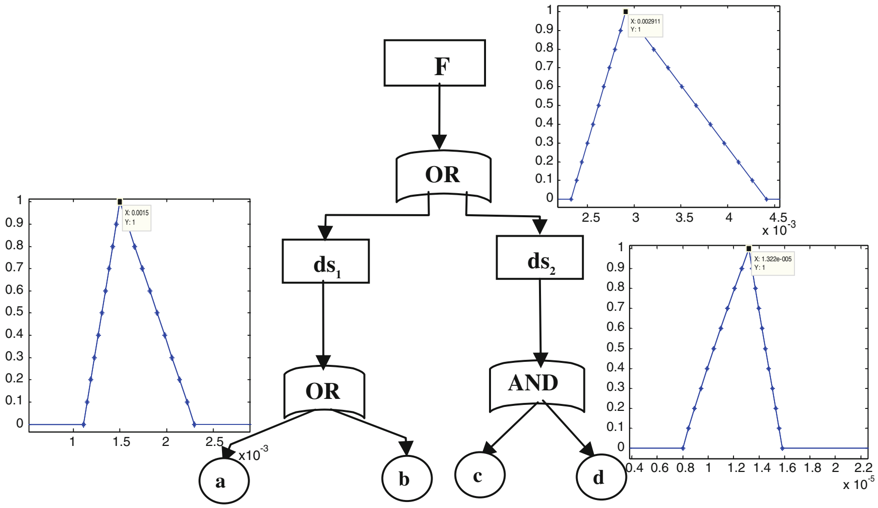


Fig. 7 Fuzzy probability of a component failure (F, ds_1 and ds_2)

- Lower value (a_i) = 0.002327,
- Middle value(m_i) = 0.002911,
- Upper value (b_i) = 0.004411.

5.1.5 Maintenance Approach

To demonstrate the effectiveness and accuracy of the recovery approach, an example with three fuzzy rules is outlined. Consider the following fuzzy rules base:

- Rule 1: IF the sojourn time $q_2 \in [10, 12]$ AND the fuzzy failure probability of the top event “F” $Pr_F \in [0.0023, 0.0044]$ THEN there is a corrective maintenance.
- Rule 2: IF the sojourn time $q_2 \in [13, 17]$ AND the fuzzy failure probability $Pr_F \in [0.0023, 0.0029]$ THEN there is a scheduled preventive maintenance.
- Rule 3: IF the sojourn time $q_2 \in [17, 20]$ AND the fuzzy failure probability $Pr_F \in [0.0044, 0.1]$ THEN there is a corrective maintenance.

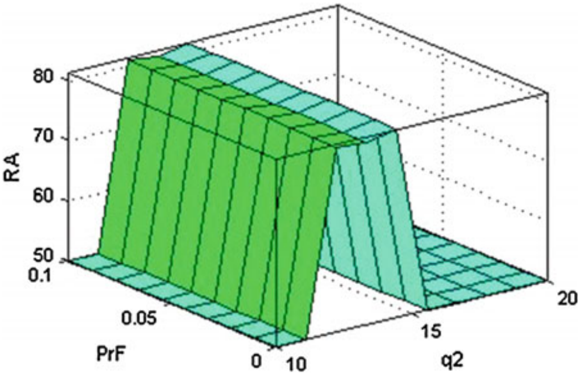


Fig. 8 Three-dimensional trapezoidal membership function [RA = f(q2, PrF)]

– Each rule use the operator "AND" in the premise, since it is an AND operation, the minimum criterion is used (Mamdani inference method), and the fuzzy outputs corresponding to these rules are represented by Fig. 8.
– Next we perform defuzzification to convert our fuzzy outputs to a single number (crisp output), various defuzzification methods were explored. The best one for this particular application: the centre of area (COA) defuzzifier. According to the COA method, the weighted strengths of each output member function are multiplied by their respective output membership function center points and summed. Finally, this area is divided by the sum of the weighted member function strengths and the result is taken as the crisp outputs.

In practice there are two fuzzy outputs to defuzzify (corrective and preventive maintenance). Analysing the data, it is noted that the appropriate technique for recovery is the corrective.

5.2 *Processing Station (Job Shop Without Assembling Tasks)*

5.2.1 Presentation of Processing Station

In the processing station, Fig. 9, workpieces are tested and processed on a rotary indexing table. The rotary indexing table is driven by a DC motor [10]. On the rotary indexing table, the workpieces are tested and drilled in two parallel processes. A solenoid actuator with an inductive sensor checks that the workpieces are inserted in the correct position. During drilling, the workpiece is clamped by a solenoid actuator. Finished workpieces are passed on via the electrical ejector, Fig. 9. The processing station is consists of [10]:

- Rotary indexing table module
- Testing module
- Drilling module
- Clamping module

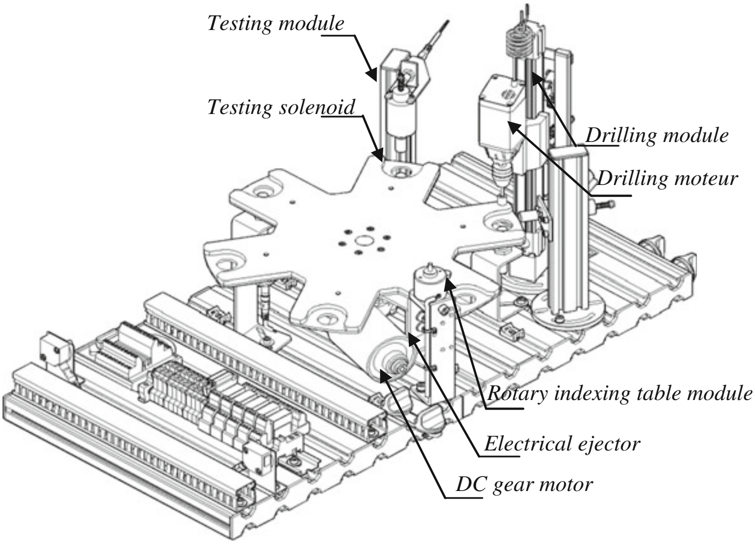


Fig. 9 Processing station

5.2.2 Modeling of Processing Unit

Figure 10, shows a P-time Petri net (G) modeling the production unit. The obtained G is used to study the maintenance of processing unit.

The full set time intervals of operations, in studied unit, are summarized in Table 2 (u.t: unit time).

5.2.3 Monitoring of Processing Unit Based on Effective Sojourn Time

The purpose of the monitoring task is to detect, localise, and identify problems that occur on the system. These problems can be physical (a piece of equipment is down, a cable is cut) or logical (a station is rebooting, a logical connection is down …).

The considered approach uses the additional information provided by the knowledge of the effective sojourn time and allows detecting a failure symptom when a constraint is violated.

Let us suppose that we want to monitor the drilling platform. In this module, a clamping device clamps the workpiece. Once the drilling is completed, the drilling machine is stopped, moved to its upper stop and the clamping device is retracted, Fig. 9.

According to P-TPN, Fig. 10, the minimum time granted to the drilling operation is 5 u.t, whereas the maximum time is 11 u.t ($IS_5 = [5, 11]$; $q_{5e} = 7$).

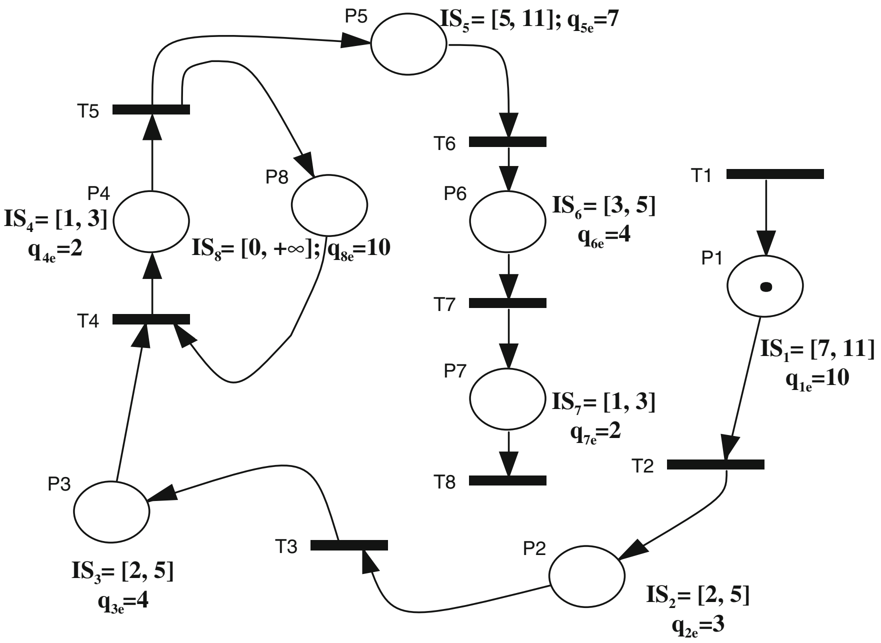


Fig. 10 Processing station modeled by a P-time Petri net

Table 2 Time intervals associated to operations

Places	Action	IS_i (u.t)	q_{ie}
P1	Turn indexing table	[7, 11]	10
P2	Testing	[2, 5]	3
P3	Turn indexing table	[2, 5]	4
P4	Clamping	[1, 3]	2
P5	Drilling	[5, 11]	7
P6	Turn indexing table	[3, 5]	4
P7	Sorting	[1, 3]	3
P8	Retraction of clamping device	[0, +∞]	10

Suppose that the drilling duration is 13 u.t (indicated by the effective sojourn time q_5 = 13 u.t and $q_5 \notin [a_5, b_5]$). This delay of sojourn time, Fig. 11, implies that:

- there is a technical failure of the production tool (clamping device, drilling machine, inductive sensor, …) requiring to generate a maintenance action,
- the quality of the manufactured product is incorrect since $q_5 \notin [a_5, b_5]$).

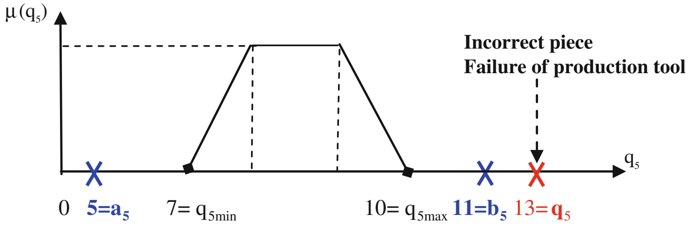


Fig. 11 Function of possibility associated with an effective sojourn time (q_5)

5.2.4 Diagnosis of Processing Station

To establish the causality of failures on the sub-systems that can affect the system status, a fault tree, Fig. 12, was constructed and processing unit failure was defined as the top event of the fault tree (F_0). This diagnostic tree was comprised of 16 basic events.

The calculation of the probability allows us to determine the critical components of the tree and improve system reliability. In addition, this probability guide us in locating basic events that contribute to the vagueness of top event failure rates and thus effectively reduce this imprecision by a feedback on the vagueness of concerned elementary events.

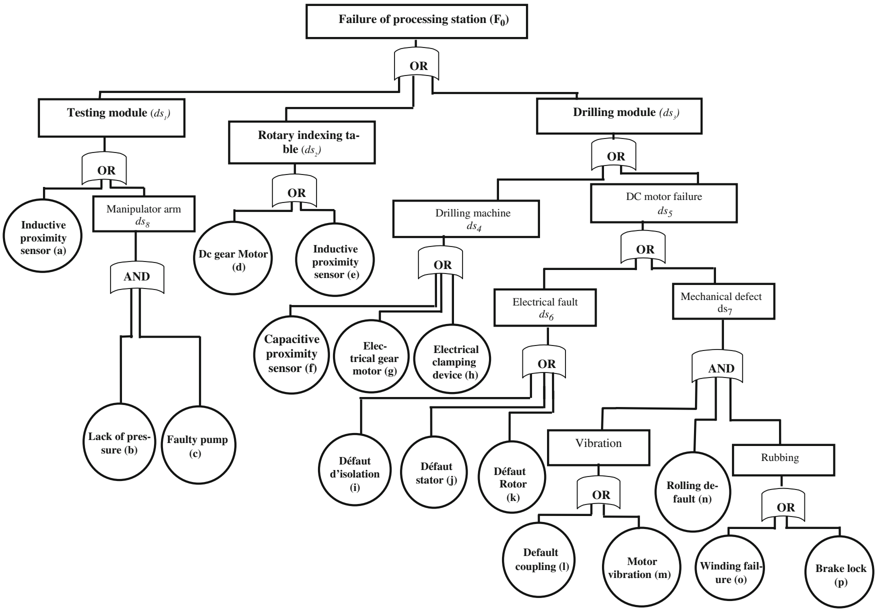


Fig. 12 Fault tree of processing unit

Table 3 Parameters of fuzzy probabilities

Basic event	m_1	m_2	a	b
d_a	0.0013	0.0018	0.00111	0.0023
d_b	0.0014	0.0014	0.00121	0.0021
d_c	0.0041	0.0041	0.0032	0.0048
d_d	0.0028	0.0032	0.0025	0.0033
d_e	0.0024	0.0028	0.0022	0.0038
d_f	0.0034	0.0034	0.0021	0.0043
d_g	0.009	0.009	0.0006	0.012
d_h	0.0012	0.0019	0.0009	0.0019
d_i	0.0015	0.018	0.0010	0.02
d_j	0.00191	0.00191	0.0017	0.00221
d_k	0.0053	0.0063	0.0045	0.0071
d_l	0.00505	0.00505	0.0041	0.0066
d_m	0.0053	0.0056	0.005	0.0061
d_n	0.00365	0.00365	0.0028	0.00676
d_o	0.0033	0.0035	0.0029	0.0042
d_p	0.00364	0.00364	0.0024	0.0044

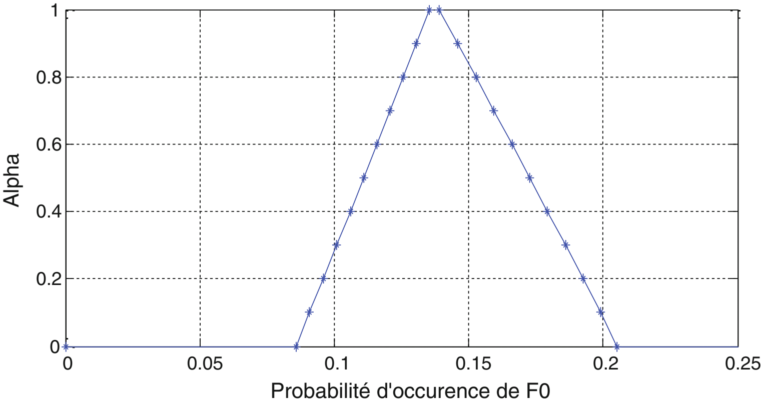


Fig. 13 Membership function for the top event failure probability

The parameter a_i, m_i, b_i are given in Table 3. We choose the trapezoidal shapes because of their mathematical simplicity. Figure 13, gives the fuzzy probability of the top event occurrence (P_{F0}). Analysing the data, it is noted that the most critical events in the fault tree are g, and i, respectively associated respectively to defaults d_g, d_i (greater probability value). Consequently we can deduce the most critical components to system failure; in fact a small variation in the critical component configuration causes a relatively greater change in the estimate of the top event failure probability.

Based on probabilistic measures, the proposed maintenance model is able to evaluate the relative influence of components reliability on the reliability of the system and provide useful information about the maintenance strategy of these elements. The FPNM model is able to trigger one or more preventive or corrective actions. Thus, the failures and repair process is capable of indicating when a failure (is about to) occur, so that repair can be performed before such failure causes damage or capital investment loss.

5.2.5 Maintenance of Processing Station

According to diagnosis information, the role of FPNM associated to the processing unit, Fig. 14, is to modify the control models, activate urgent procedures, finally, decide about the selective maintenance decision. When the maintenance is triggered by the operator after an unresolved fault case—it is the corrective maintenance policy or when triggered by the statistic block—it is the preventive and predictive maintenance.

The full set of linguistic variables associated to each input membership are summarised in Table 4.

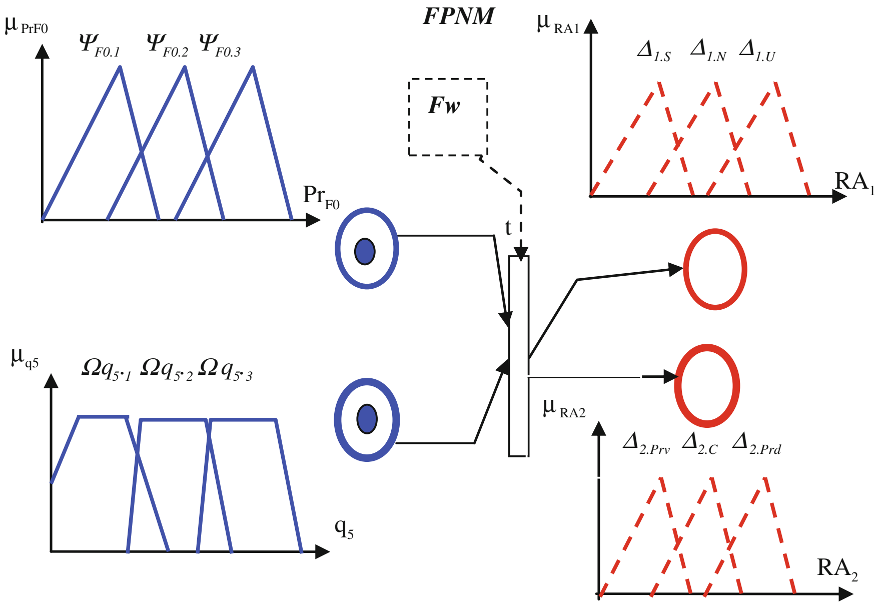

μ_{PrF0}
$\Psi_{F0.1}$ $\Psi_{F0.2}$ $\Psi_{F0.3}$
Pr_{F0}
FPNM
Fw
t
μ_{RA1}
$\Delta_{1.S}$ $\Delta_{1.N}$ $\Delta_{1.U}$
RA_1
μ_{q5}
$\Omega q_{5\cdot 1}$ $\Omega q_{5\cdot 2}$ $\Omega q_{5\cdot 3}$
q_5
μ_{RA2}
$\Delta_{2.Prv}$ $\Delta_{2.C}$ $\Delta_{2.Prd}$
RA_2

Fig. 14 FPNM of processing unit

Table 4 Linguistic variables associated to the inputs

Input	Membership	Linguistic variable
Pr_{F0}	$\Psi_{F0.1}$	Minor
	$\Psi_{F0.2}$	Average
	$\Psi_{F0.3}$	High
q_5	$\Omega q_{5.1}$	Insignificant
	$\Omega q_{5.2}$	Marginal
	$\Omega q_{5.3}$	Critical

Table 5 Linguistic variables associated to the inputs

Output	Membership	Linguistic variable
RA_1	$\Delta_{1.S}$	Slow
	$\Delta_{1.N}$	Normal
	$\Delta_{1.U}$	Urgent
RA_2	$\Delta_{2.Prv}$	Preventive
	$\Delta_{2.C}$	Corrective
	$\Delta_{2.Prd}$	Predictive

Similarly, Table 5 shows linguistic variables associated to the output “recovery action”.

Each input and output places of the proposed FPNM are associate to a fuzzy description. For the input places, we describe the marking variable of the places, whereas the output variables describe a normalized recovery action (resp. control).

In Fig. 14, the places represent the input variables, which are fuzzy probability associated to the top event F_0 “Failure of processing station” (resp sojourn time q_5 associated to the drilling operation). Initially, all of these places contain a token, which means that the truth degrees of these variables are known.

Under such an assumption, if the delay associate to the drilling operation is actually observed, the variable sojourn time “q_5” (resp. probability of the top event F_0) will have a truth degree value which is bigger than 0. On the contrary, if the drilling operation proceeds normally, the truth degree is equal to 0.

In FPNM, the transitions represent rules in which antecedent propositions implicate consequent propositions. Each rule “F_w” is associated with a certainty factor, which describes the confidence level of the rule.

The output places represent the recovery actions. These actions can be slowly, normal or urgent (resp preventive, corrective and predictive).

It is necessary to point out the purpose of the FPNM. As soon as this block is requested by the diagnosis block, it triggers different actions (slow, normal or urgent), Fig. 14. If there is a risk to the operator or the process, the proposed FPNM triggers an emergency procedure.

If the expected state is not coherent with the reference state (operations that are not in conformity with the process state), the system sets to maintenance state: This is the corrective maintenance scenario. In the case of corrective maintenance, it is necessary to “repair the defective material”, eliminate fault effects in order to reach the system’s regular operation status.

When maintenance is required (corrective, preventive or predictive), the FPNM model inhibits all pre-set regular operating conditions at the modelling and diagnosis level. At this point the maintenance module takes up control of the process. The maintenance task is, therefore, synchronized with the modelling and diagnosis model.

6 Conclusion

In this paper, we have proposed a fuzzy Petri net for maintenance, able to analyze monitoring and recovery tasks of manufacturing systems with time constrains. The new recovery approach is based on the study of effective sojourn time of the token in places and the evaluation of the failure probability of fault tree events.

Our study makes the assumption that the supervised system is modeled by P-time Petri nets. The paper proposes an adaptive technique dedicated to the maintenance of manufacturing systems with time constraints. This model has a double interface, one with the modeling model system and the second one with the behavioral model (diagnosis).

At the occurrence of a dysfunction in a milk packaging machine, it is important to react in real time to maintain the productivity and to ensure the safety of the system. It has been shown that the knowledge of the effective sojourn time of the token has a significant contribution regarding this type of problem, since it makes the supervision more efficient by an early detecting of a time constraint violation. This is quite useful for the maintenance task.

We have developed and used a fuzzy probabilistic approach to evaluate the failure probability of the top event, when there is an uncertainty about the components failure probabilities. This approach is based on the use of fuzzy probabilities.

To illustrate the efficiency of the maintenance approach, we have applied it to a packaging process. The proposed Petri net approach can achieve early failure detection and isolation for fault diagnosis. These capabilities can be very useful for health monitoring and preventive maintenance of a system.

It is interesting as further research to incorporate the issues of maintenance and repair strategies into the fuzzy probabilistic approach in order to compute a modified maintenance cost. This last problem needs a specific approach, because of the production loss which occurs when maximum time constraints are not fulfilled anymore.

Based on two workshops topology, it can be claimed that the proposed fuzzy Petri nets for maintenance allows applying various maintenance policies—corrective, preventive and predictive.

References

1. Adamyan, A., He, D.: Analysis of sequential failures for assessment of reliability and safety of manufacturing systems. Reliab. Eng. Syst. Saf. **76**(3), 227–236 (2002)
2. Azar, A.T.: Fuzzy Systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
3. Azar, A.T.: Adaptive Neuro-Fuzzy Systems. In: A.T Azar (ed) Fuzzy Systems. IN-TECH, Vienna, Austria (2010). ISBN 978-953-7619-92-3
4. Azar, A.T.: Overview of type-2 fuzzy logic systems. Int. J. Fuzzy Syst. Appl. **2**(4), 1–28 (2012)
5. Buckley, J.J., Eslami, E.: Fuzzy Markov chains: uncertain probabilities. MathWare Soft Comput. **9**(1), 33–41 (2008)
6. Buckley, J.J.: Fuzzy Probabilities: New Approach and Application, vol. 115, p. 17. Springer, Berlin (2005)
7. Cassady, C.R., Pohl, E.A., Murdock, W.P.: Selective maintenance modeling for industrial systems. J. Qual. Maintenance Eng. **7**(2), 104–117 (2001)
8. Celen, M., Djurdjanovic, D.: Operation-dependent maintenance scheduling in flexible manufacturing systems. CIRP J. Manufact. Sci. Technol. **5**(4), 296–308 (2012)
9. Dunyak, J., Saad, I.W., Wunsch, D.: A theory of independent fuzzy probability for system reliability. IEEE Trans. Fuzzy Syst. **7**(2), 286–294 (1999)
10. Festo: Mechatronics and factory automation, pp. 245–246. MPS Station (2014)
11. Fovino, I.N., Masera, M., De Cian, A.: Integrating cyber attacks within fault trees. J. Reliab. Eng. Syst. Saf. **94**(9), 1394–1402 (2009)
12. Hua, Z., Tao Z.: Dynamic job-shop scheduling with urgent orders based on Petri net and GASA. In: IEEE Chinese Conference of Control and Decision (CCDC'09), pp. 2446–2451 (2009)
13. Khalid, M.N.A., Yusof, U.K., Sabudin, M.: Solving flexible manufacturing system distributed scheduling problem subject to maintenance using harmony search algorithm. In: 4th IEEE Conference on Data Mining and Optimization (DMO), pp. 73–79 (2012)
14. Khansa, W., Denat, J.P., Collart-Dutilleul, S.: P-Time Petri nets for manufacturing systems. In: IEEE Workshop on Discrete Event Systems (WODES'96), pp. 94–102. Edinburgh (1996)
15. Kao Chang, C., Liang Hsiang, C.: Using generalized stochastic Petri nets for preventive maintenance optimization in automated manufacturing systems. J. Qual. **18**(2), 117–129 (2011)
16. Long, J., Descotes-Genon, B.: Flow optimization method for control synthesis of flexible manufacturing systems modeled by controlled timed Petri nets. In: IEEE International Conference on Robotics and Automation, pp. 598–603. USA (1993)
17. Mentes, A., Helvacioglu, I.: An application of fuzzy fault tree analysis for spread mooring systems. J. Ocean Eng. **38**(2), 285–294 (2011)
18. Mhalla, A., Jenheni, O., Collart Dutilleul, S., Benrejeb, M.: Contribution to the monitoring of manufacturing systems with time constraints: application to a surface treatment line. In: 14th International Conference of Sciences and Techniques of Automatic and Computer Engineering, pp. 243–250. Sousse (2013)
19. Mhalla, A., Collart Dutilleul S., Benrejeb, M.: Monitoring of packaging machine using synchronized fuzzy Petri nets. In: Management and Control of Production and Logistics, pp. 337–343. Brazil (2013)
20. Mhalla, A., Collart Dutilleul, S., Craye, E., Benrejeb, M.: Estimation of failure probability of milk manufacturing unit by fuzzy fault tree analysis. J. Intell. Fuzzy Syst. **26**, 741–750 (2014)
21. Minca, E., Racoceanu, D., Zerhouni, N.: Monitoring systems modeling and analysis using fuzzy Petri nets. Stud. Inform. Control **11**(4), 331–338 (2002)
22. Rocha Loures, E., Busetti de Paula, M.A., Portela Santos, E.A.: A control-monitoring-maintenance framework based on Petri net with objects in flexible manufacturing system. In: 3th International Conference on Production Research, pp. 3–6. Brazil (2006)

23. Sallak, M., Simon, C., Aubry, J.F.: Evaluating safety integrity level in presence of uncertainty. In: 4th International Conference on Safety Reliability, p. 5. Krakow (2006)
24. Sitayeb, F.B.: Contribution à l'étude de la performance et de la robustesse des ordonnancements conjoint production/ maintenance : cas du Flowshop. Thèse de Doctorat, Université de Franche Comté, pp. 88–90 (2005)
25. Tanaka, H., Fan, L.T., Lai, F.S., Toguchi, K.: Fault tree analysis by fuzzy probability. IEEE Trans. Reliab. **32**(5), 453–457 (1983)
26. Yang, S.K., Liu, T.S.: A Petri net approach to early failure detection and isolation for preventive maintenance. Int. J. Qual. Reliab. Eng. **14**(5), 319–330 (1998)
27. Zhang, W.W., Su, Q.X., Liu, P.Y.: Study of equipment virtual disassembly Petri net modeling for virtual maintenance. Advances in Computer, Communication, Control and Automation, pp. 361–367. Springer, Berlin (2012)
28. Zhang, T., Andrews, J., Guo, B.: A simulated Petri-net and genetic algorithm based approach for maintenance scheduling for a railway system. In: Advances in Risk and Reliability Technology Symposium. 20th AR2TS, pp. 86–87. Nottingham (2013)

Box and Jenkins Nonlinear System Modelling Using RBF Neural Networks Designed by NSGAII

Kheireddine Lamamra, Khaled Belarbi and Souaad Boukhtini

Abstract In this work, we use radial basis function neural network for modeling nonlinear systems. Generally, the main problem in artificial neural network is often to find a better structure. The choice of the architecture of artificial neural network for a given problem has long been a problem. Developments show that it is often possible to find architecture of artificial neural network that greatly improves the results obtained with conventional methods. We propose in this work a method based on No Sorting Genetic Algorithm II (NSGA II) to determine the best parameters of a radial basis function neural network. The NSGAII should provide the best connection weights between the hidden layer and output layer, find the parameters of the radial function of neurons in the hidden layer and the optimal number of neurons in the hidden layers and thus ensure learning necessary. Two functions are optimized by NSGAII: the number of neurons in the hidden layer of the radial basis function neural network, and the error which is the difference between desired input and the output of the radial basis function neural network. This method is applied to modeling Box and Jenkins system. The obtained results are very satisfactory.

Keywords NSGAII · Radial basis function (RBF) neural networks · Optimization · Modelling · Non linear system · Box and Jenkins system

---

K. Lamamra (✉)
Department of Electrical Engineering, University of Oum El Bouaghi, Oum El Bouaghi, Algeria
e-mail: l_kheir@yahoo.fr

K. Lamamra
Laboratory of Mastering of Renewable Energies, University of Bejaia, Bejaia, Algeria

K. Belarbi · S. Boukhtini
University of Constantine, Constantine, Algeria
e-mail: kbelarbi@yahoo.com

S. Boukhtini
e-mail: sou_boukh@yahoo.fr

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_10

1 Introduction

The first phase of modelling is to bring together the knowledge which we have about the process behaviour, from experiments and/or theoretical analysis of physical phenomena. This knowledge leads to several model assumptions. Each of these dynamic models realizes nonlinear functions between its control variables, state, and output. In the case where these functions are unknown, a black box model is used [56]. If some functions can be fixed from the physical analysis, then we talk about knowledge model [21, 56].

The second phase is to select the best model. This phase is the identification; this involves estimating the parameters of competing models. The estimation of the model parameters is performed by minimizing a cost function determined from the difference between the measured process outputs and the predicted values (prediction error).

The quality of this estimate depends on the wealth of learning sequences and effectiveness of the used algorithm. After the identification of all hypothesis models, we use the hypothesis corresponding to the best obtained predictor; the final model validation is performed according to the performance of its intended use [6, 20, 58].

There are several modelling tools, among them artificial neural networks. Several studies are currently using artificial neural networks in the field of modelling. for example: Grasso et al. [31] proposed “a new neural architecture able to accomplish the identification task. It is based on a relatively new neural algorithm, the multi-valued neural network with complex weights. The main idea is to use a set of measurements or simulations made on the system, taken at different values of geometrical parameters and at different frequencies, to train a multilayer architecture with multi-valued neurons, able to estimate the electrical parameters of the lumped model”.

Badkar et al. [7] presented “a study the Laser transformation hardening of commercially pure titanium, nearer to ASTM grade 3 of chemical composition was investigated using continuous wave 2 kW, Nd: YAG laser. The effect of laser process variables such as laser power, scanning speed, and focused position was investigated using response surface methodology and artificial neural network keeping argon gas flow rate of 10 lpm as fixed input parameter”. They described in their work, “the comparison of the heat input (HI) and ultimate tensile strength (σ) (simply called as tensile strength) predictive models based on artificial neural network and response surface methodology. The performance of the developed artificial neural network models were compared with the second-order RSM mathematical models of HI and σ. There was good agreement between the experimental and simulated values of response surface methodology and artificial neural network”.

Among the advantages of a neural network is its ability to adapt to the conditions imposed by any environment, and ease to change its parameters (weight, number of neurons, etc…) depending on the behaviour of its environment The neural networks are used to model and control dynamic systems linear and nonlinear where conventional methods fail [18, 41].

Research in the field of neural networks is focused on architecture by which neurons are combined and methodologies by which the weight of interconnections are calculated or adjusted.

Currently researchers are divided into two groups, the first is made up of biologists, physicists and psychologists, this group is trying to develop a neural model able to mimic with a given accuracy, the behaviour of the brain, the second group consists of engineers who are concerned with how the artificial neurons are interconnected to form networks with powerful computing capabilities. Actually, studies of neural networks are expanding and their use is still growing rapidly [57, 65].

Usually we associate with an artificial neural network learning algorithm to modify the processing performed in order to achieve a given task. For artificial neural network, learning can be seen as the problem of updating the weights of the connections within the network, in order to succeed the requested task [8, 40].

Generally learning neural networks can be made in two ways. In supervised learning, we have a set of examples (input-output pairs) and we must learn to give the correct output of new inputs. In reinforcement learning: we have inputs describing a situation and we receive a punishment (or error) if we give out is not adequate [11, 16, 44].

In supervised learning, the identification of the parameters of neural network is often performed by the algorithms of back-propagation, based on minimizing the training error and the chaining rule [12, 66]. This algorithm is a gradient descent on a differentiable error function.

This algorithm showed several disadvantages such as slow convergence, sensitivity to local minima and the difficulty to adjust the learning parameters (the number of neurons in the hidden layers, learning step etc…). In some networks using Hebbian learning where the synaptic weights can be adjusted during a learning phase patterns through Hebb’s formula leads to a formula expressing the weights based on grounds recognized [4, 28, 51].

Several approaches have been proposed to improve the method of back-propagation as the modification of learning step, decentralization of learning step algorithms using quasi-Newton [37], genetic algorithms [59] …etc.

In this work we propose the use of No Sorting Genetic Algorithm II (NSGA II) to construct a model using a RBF neural network with optimal structure. In this approach the NSGA II is used to optimize the number of neurons in the hidden layer of neural network, find the best connection weights between the hidden layer and output layer, find the parameters of the radial function of neurons in the hidden layer and ensure learning of neural network.

This paper arbitrary small on a compact region [15, 29, 33, 49] is organized as follows: in the second section we briefly recall the basic principles of neural networks, in the third section we present the NSGA II algorithm, its operating principle and its application in our method, and the fourth section we present the learning of radial basis function neural networks by the NSGAII, and finally we present the simulation results of the developed method in the fifth section.

2 Neural Networks "NN"

Artificial neural networks, is a branch of artificial intelligence research that aims to simulate intelligent behaviour by mimicking the way that biological neural networks work. Most artificial intelligence methods seek to reproduce human intelligence by imitating "what we do". Artificial neural networks seek to reproduce it by imitating "the way that we do it". The origins of artificial neural networks proceeded computers by some decades, but it was not until computers became generally available that real progress could be made in the development of these methods [5, 9].

There was a slight "glitch" of a decade or so following the publication of a book that heavily criticized the possibility of artificial neural networks developing into anything useful; since then, progress has been dramatic and these tools have moved on from being oddities used by specialists to general-purpose algorithms for data analysis and pattern recognition tasks [9].

A neural network is a computational model whose design is inspired schematically the functioning of real neurons [42]. Artificial neural networks are increasingly used and applied in various fields [47, 61, 64].

This concept, as well as genetic algorithms, is linked to the notion of learning that allows computers to learn by example, experience or analogy, forming the foundation for adaptive systems.

An artificial neural network is generally composed of a succession of layers, each of which takes its inputs to the outputs of the previous one. Each layer is composed of neurons, and each synapse is associated a synaptic weight.

A neural network has the ability to adapt to the conditions imposed by any environment, and easy replacement when a there are a change of the parameters of this environment, which allows him to solve problems previously qualified as difficult [24, 39, 63].

2.1 Radial Basis Function RBF NN

The radial basis function neural network is a two-layer network in which the hidden layer performs a nonlinear transformation usually a Gaussian function to map the input space (Fig. 1), then the output layer combines the outputs of the intermediate layer linearly as the outputs of whole network. They may be considered as linearly parameterized networks.

RBF's are Gaussian functions have the form:

$$f_i(x) = e^{\frac{-d(x)}{\sigma_i^2}}$$

With:

$d(x)$	the Euclidian distance given by: $d(x) = \|x - c_i\|$
c_i	centers of Gaussian functions
σ_i	widths of Gaussian functions

The radial basis function network approximation error can be reduced by increasing the number of the adjustable weights. Universal approximation results for neural networks indicate that, if the number of RBF is large enough, then the error can be made arbitrary small on a compact region [15, 29, 33, 49].

The adjustable parameters of a radial basis function neural network are thus the centers c_i and the variance sigma of the radial basis functions and the weights of the connections.

The centers and the variance are usually adjusted off line using for instance the k-means based on the data sets to be approximated while the weights of the connections are adjusted on line, during the approximation process using various methods such as least mean square [36], genetic algorithm [17], particle swarm optimization [32] …etc.

Radial basis functions are powerful techniques for interpolation in multidimensional space. Radial basis functions have been applied in the area of neural network where they may be used as a replacement for the sigmoid hidden layer transfer characteristic in multi-layer Perceptron. Radial basis function network have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian.

In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability [9, 42].

Radial basis function network have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptron. Generally this is because the only parameters that are adjusted in the learning process are the linear mapping

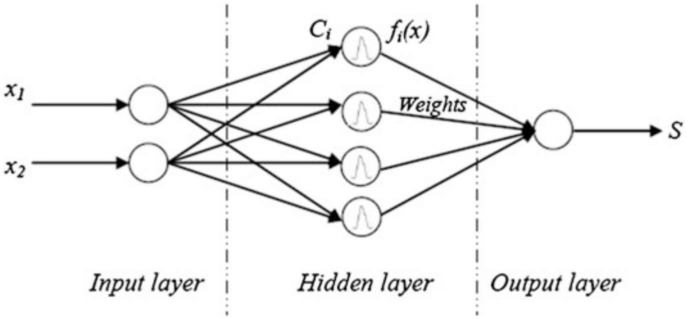


Fig. 1 General structure of a RBF neural network. Example of RBF neural network with two neurons in the input layer, four neurons in the hidden layer and one neuron in the output layer

from hidden layer to output layer. RBF network have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centers are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. In this work, it is to the NSGAII algorithm to find the best RBF centers.

Currently, the RBF neural network are used in many works for different tasks, for example in network traffic identification, where "a method of network traffic identification based on RBF neural network is proposed by analysis of the current status of the network environment. The public data set and the real-time traffic are used for a combination of supervised learning" [62].

Gutiérrez et al. [34] used in their work, the RBF neural networks in a hybrid multi-logistic methodology, called logistic regression, "the process for obtaining the coefficients is carried out in three steps. First, an evolutionary programming (EP) algorithm is applied, in order to produce an RBF neural network with a reduced number of RBF transformations and the simplest structure possible. Then, the initial attribute space (or, as commonly known as in logistic regression literature, the covariate space) is transformed by adding the nonlinear transformations of the input variables given by the RBFs of the best individual in the final generation. Finally, a maximum likelihood optimization method determines the coefficients associated with a multilogistic regression model built in this augmented covariate space".

In the work of Pendharkar [48], radial basis function neural networks are used for classification problems, "a hybrid radial basis function network-data envelopment analysis (RBFN-DEA) neural network is proposed, the procedure uses the radial basis function to map low dimensional input data from input space R to a high dimensional R^+ feature space where DEA can be used to learn the classification function".

Sheikhan et al. [54, 55] presented a "RBF-based Active queue management (AQM) controller is used, RBF as a nonlinear controller is suitable as an AQM scheme to control congestion in transmission control protocol (TCP) communication networks since it has nonlinear behaviour. Particle swarm optimization algorithm is also employed to derive RBF output weights such that the integrated-absolute error is minimized. Furthermore, in order to improve the robustness of RBF controller, an error-integral term is added to RBF equation. The output weights and the coefficient of the integral error term in the latter controller are also optimized by Particle swarm optimization algorithm". Sheikhan et al. [54, 55] "makes the Lorenz hyper-chaos synchronization and its application to improve the security of communication systems. Two methods are proposed to synchronize the general forms of hyper-chaotic systems, and their performance in securing communication application is verified. The first method uses a standard RBF neural controller. Particle swarm optimization algorithm is used to derive and optimize the parameters of the RBF controller. In the second method, with the aim of increasing the robustness of the RBF controller, an error integral term is added to the equations of RBF neural network".

In the work of Xia et al. [60] “an energy-based controller is incorporated with RBF neural network compensation, which is used to swing up the pendubot and raise it to its uppermost unstable equilibrium position”.

Radial basis function neural networks are also used in adaptive control, Jafarnejadsani et al. [38] proposed “an adaptive control based on radial-basis-function neural network for different operation modes of variable-speed variable-pitch wind turbines including torque control at speeds lower than rated wind speeds, pitch control at higher wind speeds and smooth transition between these two modes The adaptive neural network control approximates the nonlinear dynamics of the wind turbine based on input/output measurements and ensures smooth tracking of the optimal tip-speed-ratio at different wind speeds. The robust neural network weight updating rules are obtained using Lyapunov stability analysis”.

The radial basis function neural networks are used also in identification of nonlinear systems, in the work of Chai and Qiao [13], RBF neural networks are used to “model the non linear system when the system runs without a fault, after some input and output data of the system are obtained, the center of the hidden nodes are chosen using clustering technology. Assuming that the system noise and approximation error are unknown but bounded, the output weights of RBF neural network model of the system are determined by a linear-in-parameter set membership estimation. An interval containing the actual output of the system running without a fault can be predicted based on the result of the estimation. If the measured output is out of the predicted interval, it can be determined that a fault has occurred”.

Dos Santos Coelho et al. [52], used a “Radial Basis Function neural network with training combining the Gustafson-Kessel clustering method and a modified differential evolution in order to perform the swimmer velocity profile identification. The main idea is to obtain the dynamic of the velocity profile and to use it to improve the athletes’ swim style. To achieve good performance with differential evolution algorithm, the tuning of control parameters is essential as its performance is sensitive to the choice of the mutation and crossover settings”.

In this work authors “combines the two strategies described above proposing a modified differential evolution algorithm based on the association of a sinusoidal signal and chaotic sequences generated by logistic map for the mutation factor tuning. By using data collected from breaststroke and crawl swim style of an elite female swimmer; the validity and the accuracy of the RBF neural network model have been tested by simulations”.

RBF neural network are used in optimization, in the work of Mukhopadhyay et al. [46] it’s introduced “Discrete Hilbert Transform (DHT)-Neural Model which provides better result than the ARMA-Neural Model. a signal and its’ DHT produces the same Energy Spectrum. Based on this concept DHT is used for Wind Speed forecasting purpose. Thereafter the RBF neural network is used on this to forecast wind power”.

Chen et al. [14] proposed “a novel online modelling algorithm for nonlinear and non-stationary systems using a radial basis function neural network with a fixed number of hidden nodes. Each of the RBF basis functions has a tunable center vector and an adjustable diagonal covariance matrix. A multi-innovation recursive least square (MRLS) algorithm is applied to update the weights of RBF online, while the modelling performance is monitored. When the modelling residual of the RBF network becomes large in spite of the weight adaptation, a node identified as insignificant is replaced with a new node, for which the tunable center vector and diagonal covariance matrix are optimized using the quantum particle swarm optimization (QPSO) algorithm".

2.2 Learning of a Neural Network

Learning is to determine the weights for the output of the neural network to be as close as possible to the target. The main problem is how to build a neural network, how many layers of covers and the number of units (or neurons) in hidden layer required achieving a good approximation. Since a wrong choice can lead to poor network performance matching [23].

The first attempts to solve the problem of determining the architecture have been to test several networks with different architectures to achieve the desired performance [27].

In recent years, many studies have been devoted to developing methods for optimizing the architecture of the neural network. The main algorithms have been proposed can be classified into three families:

1. Pruning algorithms: detect and remove the weights or units that contribute little to the network performance [45].
2. Ascending or constructive algorithms: start from an approximate solution to the problem with a simple network and add if necessary unit or hidden layers to improve network performance [26].
3. The direct algorithms: define a suitable architecture and perform learning or perform both operations simultaneously, such as genetic algorithms [3].

3 Non-dominated Sorting Genetic Algorithm II: NSGA II

The Multi Objective Genetic Algorithm that we used in this work is the NSGAII (Non-dominated Sorting Genetic Algorithm) introduced and enhanced by Deb and Goel [19].

It is one of the most used and most cited in the literature algorithms [53]. It is widely used by many authors, not only in the context of multi-objective optimization, but also for comparison with other algorithms, it is considered as a benchmark by several researchers, for example: Hashmi et al. [35] used this algorithm in "a negotiation Web service that would be used by both the consumer and provider Web services for conducting negotiations for dependent QoS parameters".

Min et al. [43] used it in a "multi-objective history matching model to predict the individual performance".

In the work of Gossard et al. [30] NSGAII is "coupled with an artificial neural network to optimize the equivalent thermo-physical properties of the external walls (thermal conductivity kwall and volumetric specific heat (ρc) wall) of a building in order to improve its thermal efficiency".

Adham et al. [1] used NSGAII "as an optimization technique in combination with a multi-objective general optimization scheme with the thermal resistance model as an analysis, for a potential improvement in the overall performance of a rectangular micro-channel heat sink using a new gaseous coolant namely ammonia gas".

In the work of Prasad and Singru [50] NSGA-II is used to "select the optimum design of turbo-alternator (TA), a real-life TA used in an industry is considered". Domínguez et al. [22] proposed "a high-performance architecture for the NSGA-II using parallel computing, for evaluation functions and genetic operators. In the proposed architecture, the Mishra Fast Algorithm for finding the Non Dominated Set was used; it's proposed a modification in the sorting process for the NSGA-II that improves the distribution of the solutions in the Pareto front".

NSGAII is an algorithm establishing the dominance relationships between individuals and providing a fast sorting method of chromosomes [19]. This algorithm uses a measure of crowding around individuals to ensure diversity in the population. The principle of this algorithm is shown in Fig. 2.

At the beginning, an initial population is randomly generated, and then it undergoes a sorting using the concept of non-domination. Each solution is assigned a strength or rank equal to the level of non-dominance (1 for best, 2 for the next level, etc…). The reproduction step consists of a tournament for the selection of parents.

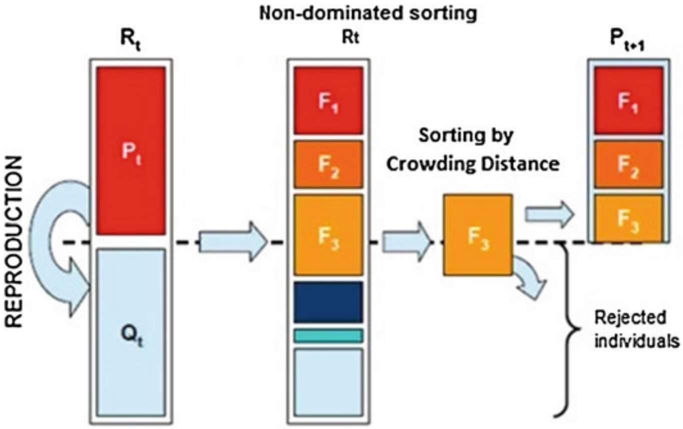


Fig. 2 Operating principle of NSGAII

When two individuals of the population are chosen randomly in the population, the tournament is based on a comparison of the domination with constraints of the two individuals. For a given generation t, we create $R_t = P_t U Q_t$, Q_t is children population of the previous population P_t (generated from the parents through the operators of crossover and mutation), R_t includes individuals of $\mathrm{P_t}$, which ensures the elite nature of the algorithm NSGAII. Population R_t contains 2 N individuals (it is composed of N parents and N children). Then R_t undergoes a sorting using the concept of non-dominance of Pareto. Individuals are grouped into non-dominated fronts such as F_1 represents individuals of rank 1, F_2 individuals of rank 2, etc…

The next objective is to reduce the number of individuals in the 2 N population R_t for a population P_{t+1} of size N. If the size of F_1 is less than N, then all F_1 individuals are retained. It is the same for the other fronts as long as the number of individuals retained does not exceed the size N.

If we take the example of Fig. 2, the fronts F_1 and F_2 are fully retained but the conservation front F_3 will result in exceeding the size N of the population P_{t+1}. It must then make a selection of F_3 individuals to keep.

It is then necessary to make a selection of F_3 individuals to maintain. In this case, NSGAII involves a mechanism for preserving the diversity of the population based on the evaluation of the density of individuals around each solution through a procedure for calculating the “distance proximity”.

A low value of the proximity distance for an individual is an individual “well surrounded”. It then proceeds to a descending sorting according to this distance proximity to retain individuals $\mathrm{F_3}$ front and eliminate individuals from the densest areas. This way we complete the population P_{t+1}. Individuals with extreme values for the criteria are preserved by this mechanism, thereby maintaining the external terminals of the Pareto front.

At the end of this phase, the population P_{t+1} is created. Then a new population Q_{t+1} is generated by reproduction from P_{t+1}. We continue iteratively the procedure described above to the satisfaction of stop criteria set by the user.

Generally, the NSGAII keeps elitism and diversity without adding additional parameters, while using an algorithm attractive in its simplicity with a minimum of parameters.

4 Learning of RBF NN by the NSGAII

The NSGA II is used to optimize the structure and parameters of the RBF NN. The following two objective functions are chosen:

- The first function to be optimized (f_1) is the number of neuron of the hidden layer of the RBF NN.
- The second function (f_2) is the quadratic error which is the difference between the desired input of the RBF NN and its output.

NSGAII must find the best number of neurons in the hidden layer (Nn) and provide the best connection weights between neurons in the hidden layer and the output layer, and find the parameters of the radial function of neurons hidden layer. In this work, we used the radial functions of Gaussian form, and NSGA II must find the best centers (C_i) and the best widths sigma (σ_i) for these functions.

The chromosome then contains the number of neurons in the hidden layer, Gaussian functions centers and widths of the hidden layer neurons, and the weights of connections between the hidden layer and the output layer.

The chromosome contains the following parameters:

$$[N_n C_1 C_2 \ldots C_{Nn} \sigma_1 \sigma_2 \ldots \sigma_{Nn} Z_1 Z_2 \ldots Z_{Nn}] \tag{1}$$

where:

N_n	is the number of neuron in the hidden layer
C_2 $C_1 \ldots$ C_{Nn}	Gaussian functions centers of the hidden layer neurons
σ_1 σ_2 ... σ_{Nn}	widths of the Gaussian functions of the hidden layer neurons
Z_1 $Z_2 \ldots$ Z_{Nn}	the connection weights between the neurons of the hidden layer and the neuron of the output layer

The length of the chromosome (*Lch*) depends only on the number of neurons in the hidden layer (N_n) and the number of neurons of the output layer (N_{ns}) because the inputs are fixed to two.

The general expression of the length of the chromosome (*Lch*) is given by [2]:

$$Lch = (2 + N_{ns}) * \max(N_n) + 1 \tag{2}$$

For example, for a neural network with one output and two neurons in the hidden layer, the length of the chromosome is *Lch* equal to 7: the number of neurons N_n (one allele), Gaussian functions centers neurons of the hidden layer C_1 C_2 (two alleles), the widths "σ" of the Gaussian functions of the hidden layer neurons σ_1 σ_2 (two alleles), the connection weights between the neurons of the hidden layer and the output layer neuron Z_1 Z_2 (two alleles).

The population size T_m (population matrix) is given by [3]:

$$Tm = N * ((2 + N_{ns}) * \max(N_n) + 1) \tag{3}$$

N number of individuals in the population

For example, if the maximum number of neurons in the hidden layer is equal to 20, then the length of the population chromosomes is equal to 61. In this case if the neurons number in the chromosome *i* is equal to 6 ($N_{ni} = 6$), it will be organized as follows:

$$[N_{ni}(=6)\,C_1\ldots C_6\ldots\sigma_1\ldots\sigma_6 Z_1\ldots Z_6 0\,0\ldots 0]\,(Lch\;=\;61) \tag{4}$$

N_{ni} neurons number in the hidden layer of the *i*th individual

Bellow an example of 3 individuals in a population composed of 80 individuals where N_{nmax} is 20:

$$N_{n1}=5;\ N_{n30}=20;\ N_{n80}=2$$

$$\left|\begin{array}{lll}
Nn_1(=5)\ \ C_1\ldots C_5 & \sigma_1\ldots\sigma_5\ Z_1\ldots Z_5\ 0\ 0 & 0 \\
\ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots \\
Nn_{30}(=20)\ \ C_1\ldots C_{20} & \sigma_1\ldots\sigma_{20}\ Z_1 & Z_{20} \\
\ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots \\
Nn_{80}(=2)\ \ C_1 C_2 & \sigma_1\sigma_2\ \ Z_1 Z_2\ 0\ 0 & 0
\end{array}\right|$$

5 Results of Simulation

This method is applied to modelling the BOX and JENKINS system, which is a time series. This process is a gas-fired boiler with the input the gas at the inlet and the output the concentration of released CO_2.

> A time series is a sequence of observations on a variable measured at successive points in time or over successive periods of time. The measurements may be taken every hour, day, week, month, or year, or at any other regular interval. The pattern of the data is an important factor in understanding how the time series has behaved in the past. If such behaviour can be expected to continue in the future, we can use the past pattern to guide us in selecting an appropriate forecasting method.
> To identify the underlying pattern in the data, a useful first step is to construct a time series plot. A time series plot is a graphical presentation of the relationship between time and the time series variable; time is on the horizontal axis and the time series values are shown on the vertical axis. Let us review some of the common types of data patterns that can be identified when examining a time series plot [2].

The data of BOX and JENKINS consist of 296 measurements of input and output [10, 25].

The RBF neural network has two inputs, one output and a hidden layer. The number of neurons in the hidden layer Nn is determined by the NSGA II as well as the centers and the widths of the Gaussian functions, and the weights of connection between the hidden layer and the output layer.

The NSGA II optimizes simultaneously Nn and the quadratic cumulated error e_c given by:

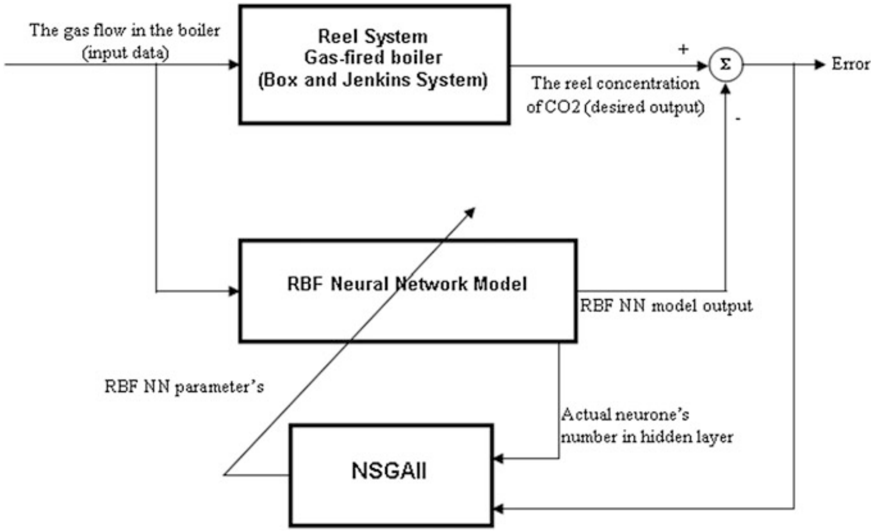


Fig. 3 Modelling schema of Box and Jenkins system using RBF NN

$$e_c = \sum_{i=1}^{N} e^2(i); \quad \text{with } e(i) = \sum_{i=1}^{N} (Y_d(i) - Y_r(i))$$

where:

e_c	cumulative error. e(i): instantaneous error. N: length of the simulation sequence (the number of data N = 296)
Yd	the desired out put,
Yr	real output (output of the RBF NN model)

The modelling schema is presented in the Fig. 3,

The neural network learning is performed on 100 data of model of Box and Jenkins and validation is performed on the remaining data (196 data).

The results obtained which represent the first Pareto front (which containing the best individuals or non dominated individuals) of the last generation is given in Table 1.

By analyzing this table, we can notice that the Pareto front contains five non dominated individuals. There is an important difference between the global error of the individual 5 and the other individuals. It was selected as the best model. The structure of this RBF neural network model is composed of seven neurons in the hidden layer and an overall error 0.5259.

Figure 4 shows the gas flow in the boiler, which represent the global real input data.

Figure 5 shows the real input training data.

Figure 6 represents the real input validation data.

Table 1 Pareto front

No. of individual	Number of neurons in the hidden layer Nn	Training error (100 data) e_t	Validation error (196 data) e_V	Global error (296 data) e.g.
1	2	5.4241	11.0611	16.4852
2	3	4.7137	8.7448	13.4585
3	4	3.0042	6.1185	9.1227
4	6	1.0218	3.5033	4.5251
5	7	0.2043	0.3216	0.5259

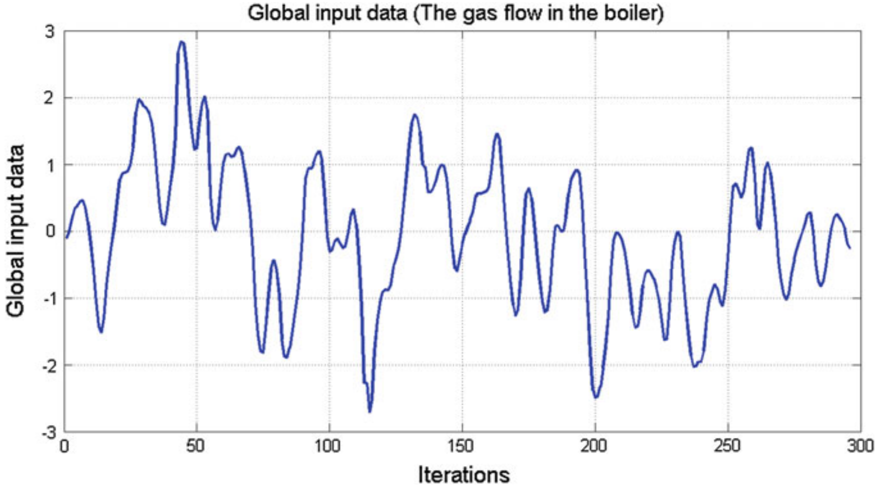


Fig. 4 Global input data, Gaz flow in the boiler

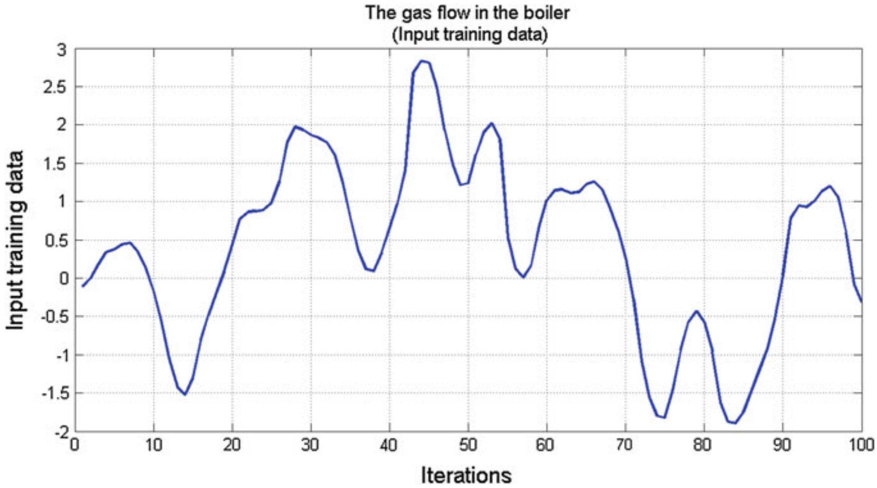


Fig. 5 Input training data

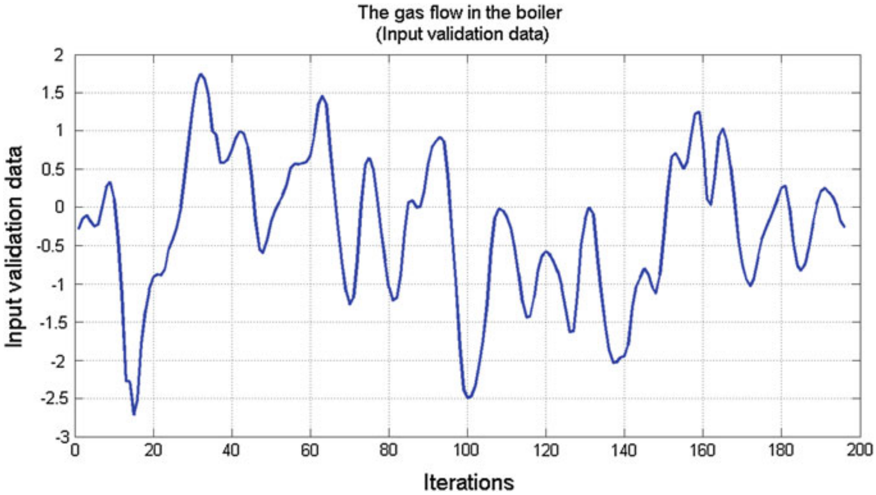


Fig. 6 Input validation data

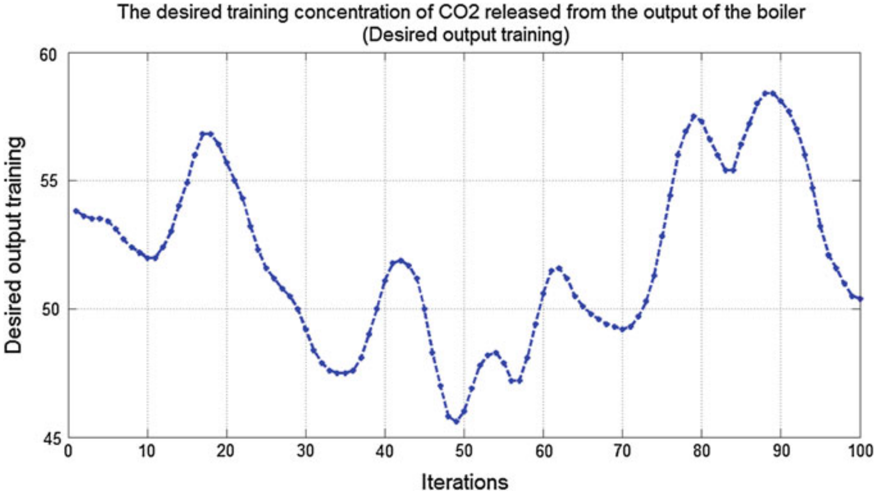


Fig. 7 Desired output training

Figure 7 represents the desired output training, which represent the desired training concentration of CO_2 released from the output of the boiler.

Figure 8 shows the RBF neural network model output during training phase.

Figure 9 represents the desired output and RBF neural network model output during training phase, and the Training error is shown in the Fig. 9.

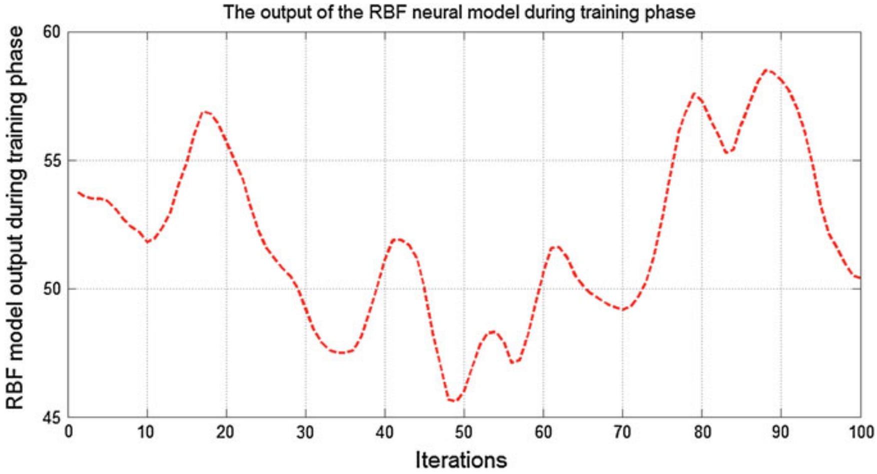


Fig. 8 RBF neural network model output during training phase

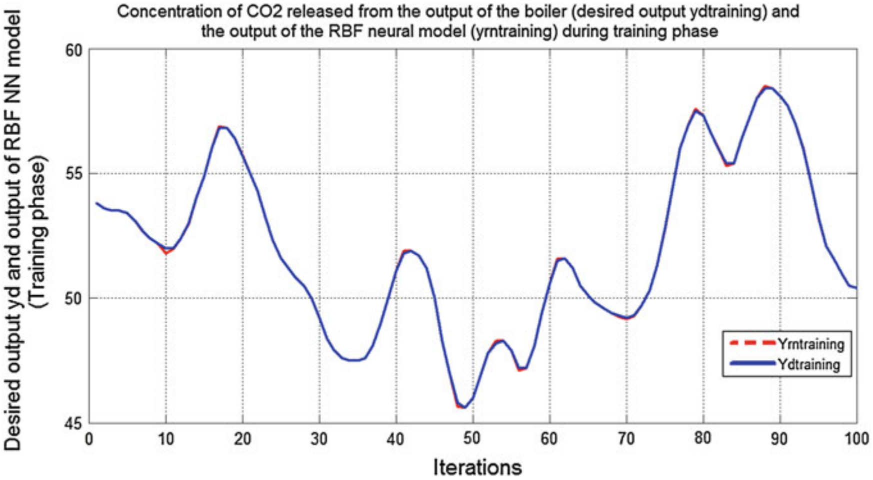


Fig. 9 Desired output and RBF neural network model output during training phase

By analyzing these figures, we can observe that the output of RBF neural network model has perfectly followed the desired output during the training phase with a training error of 0.2043.

In the training error figure (Fig. 10), the most significant peak appears at 10th iteration.

Figure 11, represents the desired validation concentration of CO_2 released from the output of the boiler (the desired output validation).

Figure 12, shows the RBF neural network model output during validation phase.

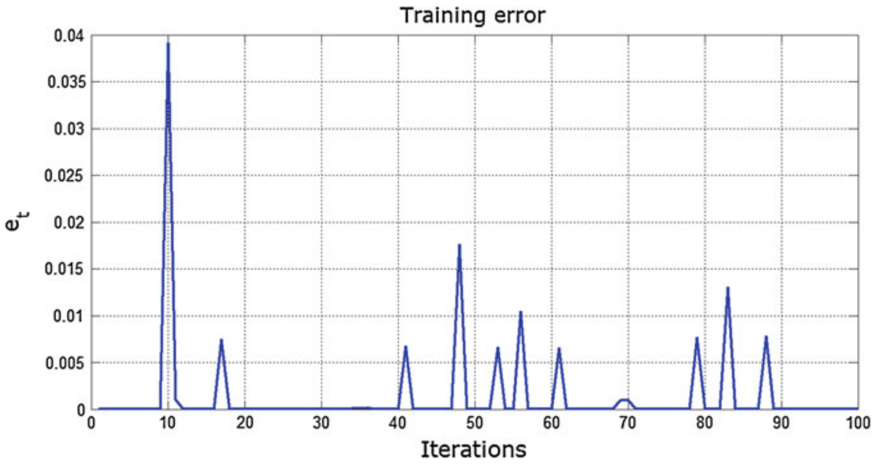


Fig. 10 Training error

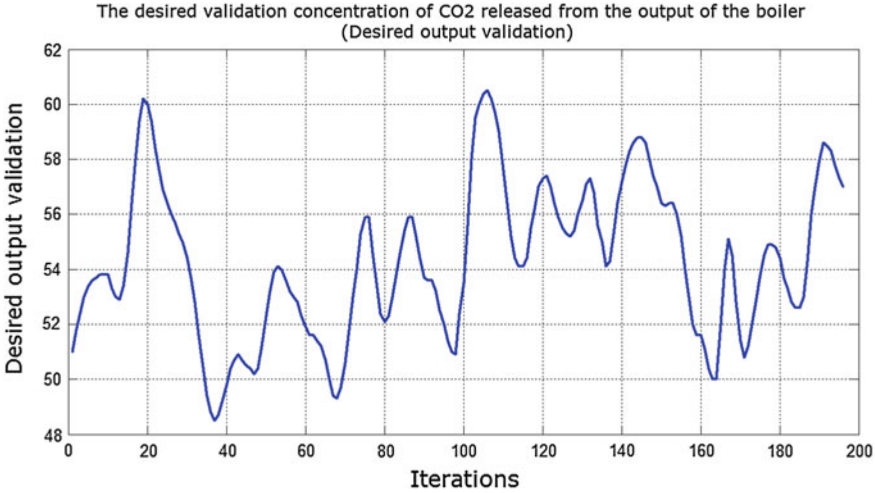


Fig. 11 Desired output validation

Figure 13, represents the desired output and the RBF neural network model output during validation phase, and the validation error is shown in the Fig. 14.

In these figures we can also observe that the output of RBF neural network model has followed the desired output during the validation phase with a training error of 0.3216.

This error is larger than the training error; this can be justified of the fact that the training data number chosen is lower than validation data number.

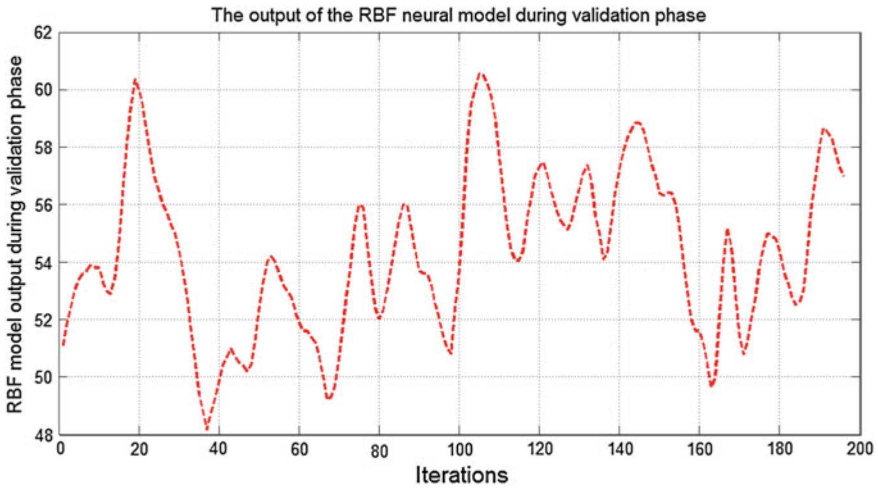


Fig. 12 RBF neural network model output during validation phase

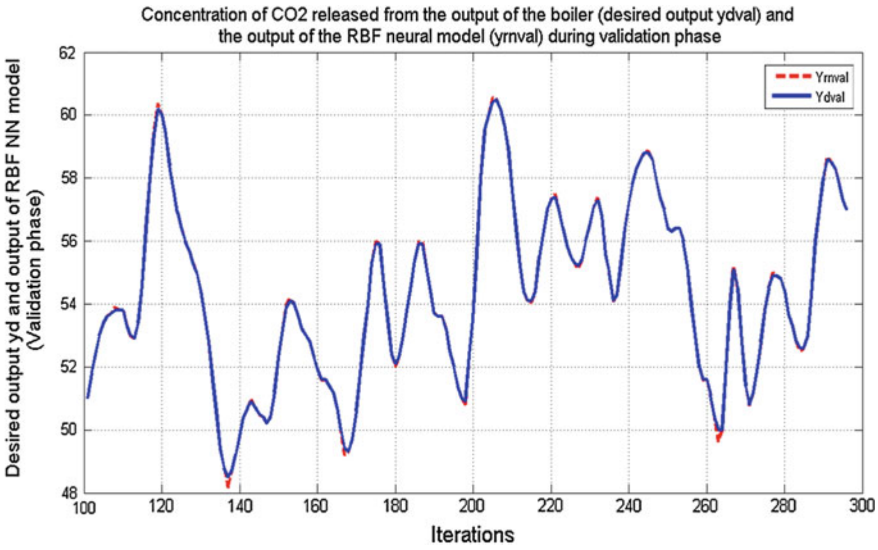


Fig. 13 Desired output and RBF neural network model output during validation phase

In the validation error figure (Fig. 14), the most significant peaks are appeared at 37th and 163th iterations.

The global desired output which is the global desired concentration of CO_2 released from the output of the boiler is shown in the Fig. 15.

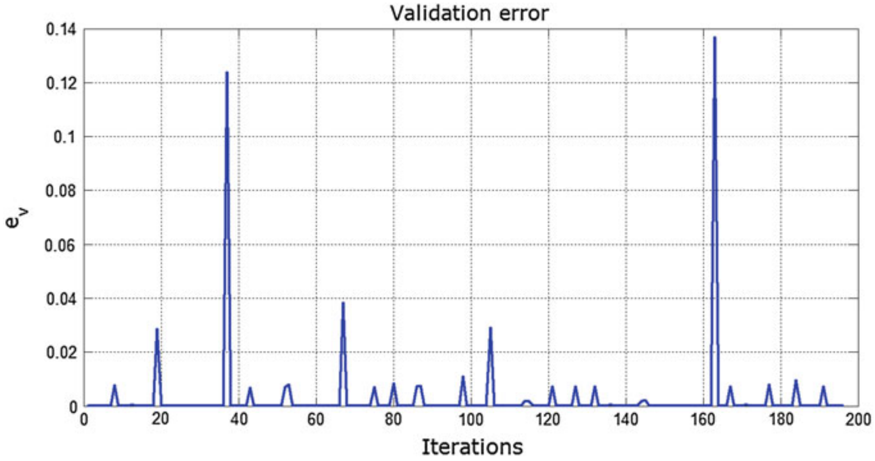


Fig. 14 Validation error

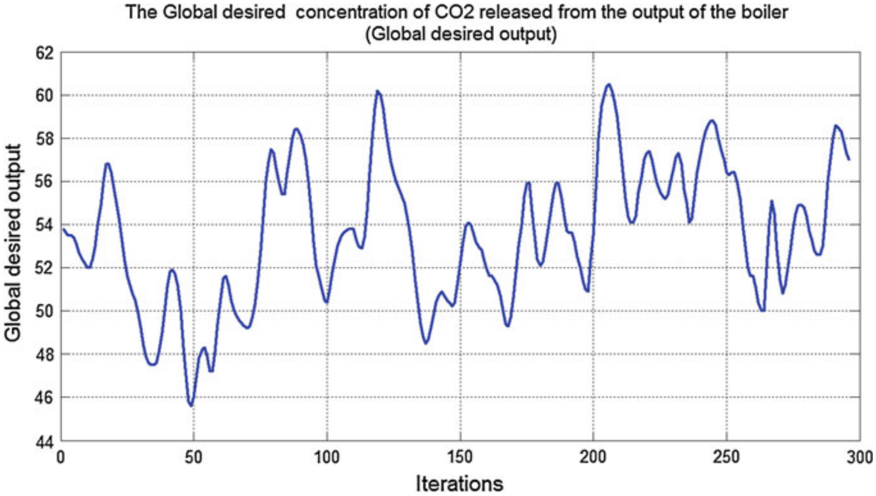


Fig. 15 Global desired output

Figure 16 represents the global RBF neural network model and the Fig. 17 shows the concentration of CO_2 released from the output of the boiler (desired output yd) and the output of the RBF neural model (y_r).

Figure 18 represents the global error, and finally, the Pareto front “global cumulated error function of the number of neurons” is shown in the Fig. 19.

Based on these results, we can conclude that the multi-objective genetic algorithm NSGAII gave a good structure of radial basis function neural network model, with a good number of neurons in the hidden layer and good connection weights

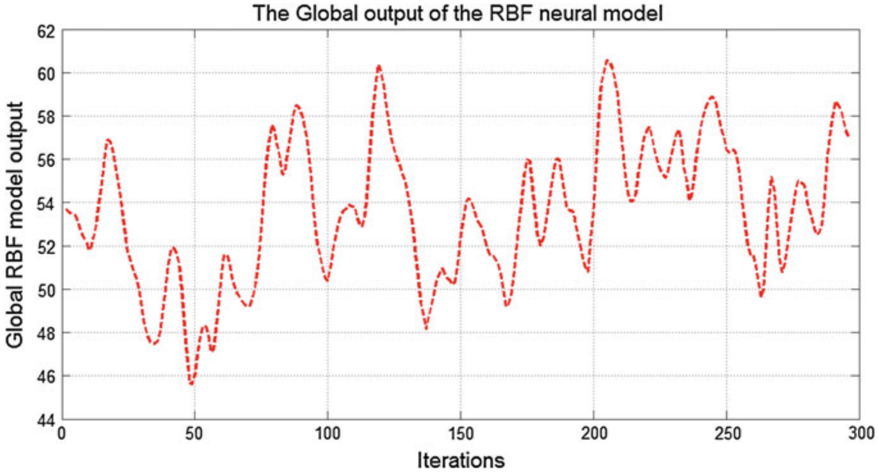


Fig. 16 Global RBF neural network model

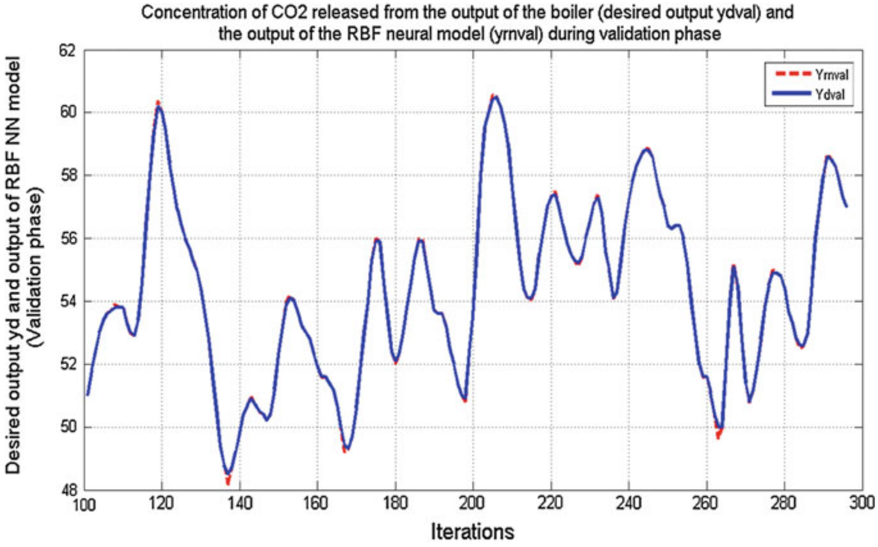


Fig. 17 The concentration of CO_2 released from the output of the boiler (desired output yd) and the output of the RBF neural model (year)

between the hidden layer and the output layer, and it also found the best parameters of the radial function of hidden layer neurons, because we see that the radial basis function neural network model output is very close to that of the desired output, with training error equal to 0.2043 and validation error equal to 0.3216 and a global error equal to 0.5259.

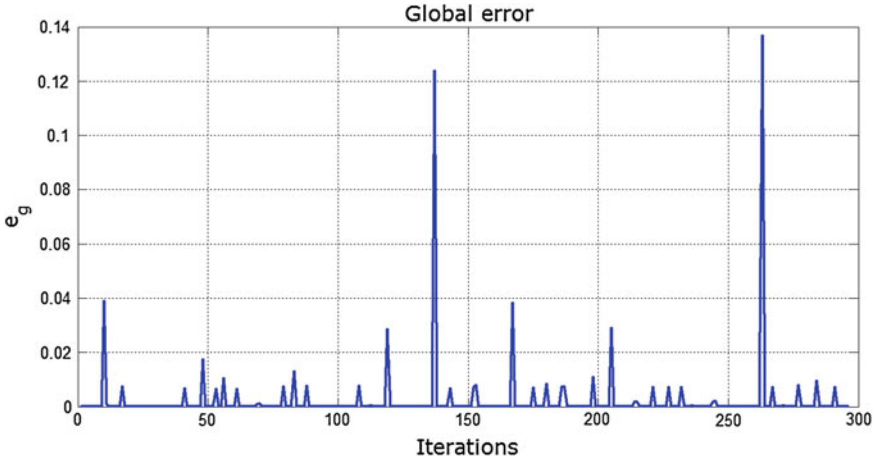


Fig. 18 The global error

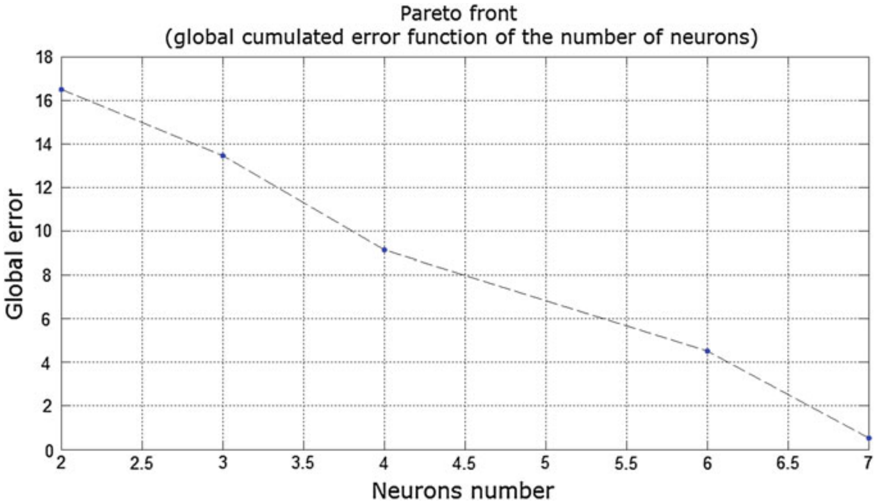


Fig. 19 Pareto front “global cumulated error function of the number of neurons”

We are currently working on the improvement of this technique; however we try to improve a new optimization method for Radial Basis Function and Multi Layer Perceptron neural networks. To do so, we did modelling by RBF and MLP neural networks with the optimization of three objectives.

The structure of neural network is hereby amended. In this technique, the input variables and the number of input neurons are not fixed and they will be included in the multi-objective optimization process carried out by the NSGAII algorithm. We have to minimize the following three functions:

- The number of neuron of the hidden layer of the radial basis function neural network.
- The quadratic error which is the difference between the desired input of the RBF neural network and its output.
- The regressor's number at the input of the RBF neural network.

6 Conclusion

Neural networks are increasingly used and applied in various fields, mainly in the problems of modelling of complex systems. This is due to their simplicity and their universal approximation properties and the ability of information parallel treatment. These properties make that these networks are well used for modeling and controlling linear and nonlinear dynamics systems, where conventional methods fail.

The most difficult problem to solve for neural networks is to obtain the best and right architecture. The networks established in most of practical applications are built with an experimental way.

This difficulty can be highlighted by a number of issues, such as the number of hidden layers to be used in a multilayer network, the optimal number of neurons in each layer and the initial values of connection weights during the learning phase … etc. A bad choice can lead to poor performances of the corresponding network.

In this work we present a technique to solve the problems mentioned above. This technique is to treat these problems as a multi-criteria optimization problem. We considered in this work the designing of RBF neural network using the multi-objective genetic algorithms type NSGAII, by optimizing simultaneously two objectives functions: the first function is the quadratic cumulatively error, which is the difference between the desired signal and the RBF neural network model output signal, and the second is the number of neurons in the hidden layer, thereby the NSGA II chromosome contains the number of neurons in the hidden layer, Gaussian functions centers and widths of the hidden layer neurons, and the weights of connections between the hidden layer and the output layer. At the end of the evolution of this algorithm off-line, we have a set of RBF models which forming the final Pareto front, and includes all allowed results, and ensuring the predefined criteria.

This optimization technique is applied to modelling a nonlinear system which is the BOX and JENKINS process. It's a gas-fired boiler with the input the gas at the inlet and the output the concentration of released CO_2. The results show that using NSGAII to optimize the RBF neural network provides a good model, these results are very satisfying.

At the end of the NSGAII algorithm evolution, it converges to a set of solutions (Pareto front), it is a set of solutions respecting the optimization criteria, which is generally difficult to choose one solution from the set, to resolve this problem we proposed in the future works, the uses of selection method such as the multi-criteria decision analysis approach.

In our future work, we are working on the improvement of this technique; nevertheless we try to improve a new optimization method for RBF and MLP neural network by the multi objectives genetic algorithms NSGAII and others and also using Particle Swarm Optimization. These techniques are applied in different areas, such as modelling and control of complex nonlinear systems, modelling of transistors, modelling and control in the field of renewable energies.

References

1. Adham, A.M., Mohd-Ghazali, N., Ahmad, R.: Optimization of an ammonia-cooled rectangular microchannel heat sink using multi-objective non-dominated sorting genetic algorithm (NSGA2). Heat Mass Transf. **48**(10), 1723–1733 (2012). doi:10.1007/s00231-012-1016-8
2. Anderson, D., Sweeney, D., Williams, T., Camm, J., Martin, R.: An introduction to management science: quantitative approaches to decision making, revised. Cengage Learning (2011)
3. Angeline, P.J., Saunders, G.M., Pollack, J.B.: An evolutionary algorithm that constructs recurrent neural networks. IEEE Trans. Neural Netw. **5**(1), 54–65 (1994). doi:10.1109/72.265960
4. Azar, A.T.: Adaptive neuro-fuzzy system as a novel approach for predicting post-dialysis urea rebound. Int. J. Intell. Syst. Technol. Appl. **10**(3), 302–330 (2011). doi:10.1504/IJISTA.2011.040352
5. Azar, A.T.: Fast neural network learning algorithms for medical applications. Neural Comput. Appl. **23**(3–4), 1019–1034 (2013). doi:10.1007/s00521-012-1026-y
6. Azar, A.T., Yashiro, M., Schneditz, D., Roa, L.M.: Double pool urea kinetic modeling. In: Azar, A.T. (ed.) Modelling and Control of Dialysis Systems. Studies in Computational Intelligence, vol. 404, pp. 627–687. Springer, Berlin (2013)
7. Badkar, D.S., Pandey, K.S., Buvanashekaran, G.: Development of RSM- and ANN-based models to predict and analyze the effects of process parameters of laser-hardened commercially pure titanium on heat input and tensile strength. Int. J. Adv. Manuf. Technol. **65**(9–12), 1319–1338 (2013). doi:10.1007/s00170-012-4259-0
8. Bermejo, D.M.A.V.: A mathematical model to predict δ- ferrite content in austenitic stainless steel weld metals. Weld. World **56**(9–10), 48–68 (2012). doi:10.1007/BF03321381
9. Binder, M.D., Hirokawa, N., Windhorst, U. (eds.) Artificial neural networks. In: Encyclopedia of Neuroscience, pp. 185–185. Springer, Berlin (2009)
10. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control. Wiley, New York (2013)
11. Cantley, K.D., Subramaniam, A., Stiegler, H.J., Chapman, R.A., Vogel, E.M.: Hebbian learning in spiking neural networks with nanocrystalline silicon TFTs and memristive synapses. IEEE Trans. Nanotechnol. **10**(5), 1066–1073 (2011)
12. Carrasco, R., Sanchez, E.N., Carlos-Hernandezy, S.: Neural network identification for biomass gasification kinetic model. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1888–1893. Available at http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6033454 (2011). Accessed: 2 May 2014
13. Chai, W., Qiao, J.: Non-linear system identification and fault detection method using RBF neural networks with set membership estimation. Int. J. Model. Ident. Control **20**(2), 114 (2013). doi:10.1504/IJMIC.2013.056183
14. Chen, H., Gong, Y., Hong, X.: Online Modeling With Tunable RBF Network. IEEE Trans. Cybern. **43**(3), 935–947 (2013). doi:10.1109/TSMCB.2012.2218804

15. Chen, T., Chen, H.: Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks. IEEE Trans. Neural Netw. **6**(4), 904–910 (1995). doi:10.1109/72.392252
16. Cherkassky, V., Friedman, J.H., Wechsler, H.: From statistics to neural networks: theory and pattern recognition applications. Springer Publishing Company, Incorporated (2012)
17. Cook, D.F., Ragsdale, C.T., Major, R.L.: Combining a neural network with a genetic algorithm for process parameter optimization. Eng. Appl. Artif. Intell. **13**(4), 391–396 (2000). doi:10.1016/S0952-1976(00)00021-X
18. Dahl, G.E., Sainath, T.N. and Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8609–8613. IEEE (2013)
19. Deb, K., Goel, T.: Controlled Elitist non-dominated sorting genetic algorithms for better convergence. In: Zitzler, E., Thiele, L., Deb, K., Coello, C.A.C., Corne, D. (eds.) Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science, pp. 67–81. Springer, Berlin (2001)
20. Deichmueller, M., Denkena, B., de Payrebrune, K.M., Kröger, M., Wiedemann, S., Schröder, A., Carstensen, C. (2013) Modeling of process machine interactions in tool grinding. In: Process Machine Interactions, pp. 143–176. Springer, Berlin
21. Deutschmann, O.: Modeling and Simulation of Heterogeneous Catalytic Reactions: From the Molecular Process to the Technical System. Wiley, New York (2013)
22. Domínguez, J., Montiel-Ross, O., Sepúlveda, R.: High-performance architecture for the modified NSGA-II. In: Melin, P., Castillo, O. (eds.) Soft Computing Applications in Optimization, Control, and Recognition. Studies in Fuzziness and Soft Computing, vol. 294, pp. 321–341. Springer, Berlin Heidelberg (2013)
23. Urbani, D., Marcos, S., Thiria, S.: (1995) Statistical methods for selecting neural architectures: application to the design of models of dynamic processes. PhD thesis of University Pierre and Marie Curie
24. Furtuna, R., Curteanu, S., Leon, F.: Multi-objective optimization of a stacked neural network using an evolutionary hyper-heuristic. Appl. Soft Comput. **12**(1), 133–144 (2012). doi:10.1016/j.asoc.2011.09.001
25. Box, G.E.P., Jenkins, G.M.: Time series analysis: forecasting and control, p. 575. Holden-Day, San Francisco (1976)
26. Giles, C.L., Chen, D., Sun, G.-Z., Chen, H.-H., Lee, Y.-C., Goudreau, M.W.: Constructive learning of recurrent neural networks: limitations of recurrent cascade correlation and a simple solution. IEEE Trans. Neural Netw. **6**(4), 829–836 (1995). doi:10.1109/72.392247
27. Giles, C.L., Miller, C.B., Chen, D., Chen, H.H., Sun, G.Z., Lee, Y.C.: Learning and extracting finite state automata with second-order recurrent neural networks. Neural Comput. **4**(3), 393–405 (1992). doi:10.1162/neco.1992.4.3.393
28. Gilson, M., Py, J.S., Brault, J.-J., Sawan, M.: Training recurrent pulsed networks by genetic and Taboo methods. In: Canadian Conference on Electrical and Computer Engineering, 2003, IEEE CCECE 2003, vol. 3, pp. 1857–1860 (2003). doi:10.1109/CCECE.2003.1226273
29. Girosi, F., Poggio, T.: Networks and the best approximation property. Biol. Cybern. **63**(3), 169–176 (1990). doi:10.1007/BF00195855
30. Gossard, D., Lartigue, B., Thellier, F.: Multi-objective optimization of a building envelope for thermal performance using genetic algorithms and artificial neural network. Energy Build. **67**, 253–260 (2013). doi:10.1016/j.enbuild.2013.08.026
31. Grasso, F., Luchetta, A., Manetti, S., Piccirilli, M.C.: System identification and modelling based on a double modified multi-valued neural network. Analog Integr. Circ. Sig. Process **78** (1), 165–176 (2014). doi:10.1007/s10470-013-0211-y
32. Guerra, F.A., dos Coelho, L.S.: Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis neural network with learning by clustering and particle swarm optimization. Chaos, Solitons Fractals **35**(5), 967–979 (2008). doi:10.1016/j.chaos.2006.05.077

33. Gupta, M.M., Rao, D.H.: Neuro-control systems: theory and applications. IEEE, New York (1993)
34. Gutiérrez, P.A., Hervas-Martinez, C., Martínez-Estudillo, F.J.: Logistic regression by means of evolutionary radial basis function neural networks. IEEE Trans. Neural Netw. **22**(2), 246–263 (2011). doi:10.1109/TNN.2010.2093537
35. Hashmi, K., Alhosban, A., Najmi, E., Malik, Z., Rezgui (2013) Automated Web service quality component negotiation using NSGA-2. In: 2013 ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1–6. doi:10.1109/AICCSA.2013.6616502
36. Haykin, S., Widrow, B. (2003) Least-Mean-Square Adaptive Filters. Wiley, New York (2003)
37. Jacek M.Z.: Introduction to Artificial Neural Systems. Jaico Publishing House, Mumbai (1992)
38. Jafarnejadsani, H., Pieper, J., Ehlers, J.: Adaptive control of a variable-speed variable-pitch wind turbine using radial-basis function neural network. IEEE Trans. Control Syst. Technol. **21**(6), 2264–2272 (2013). doi:10.1109/TCST.2012.2237518
39. Lamamra, K., Belarbi, K., Bosche, J., Hajjaji, A.E.L.: A neural network controller optimised with multi objective genetic algorithms for a laboratory anti-lock braking system. Sci. Technol. J. Constantine 1 Univ **35** (2012)
40. Kasabov, N., Dhoble, K., Nuntalid, N., Indiveri, G.: Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. Neural Networks **41**, 188–201 (2013)
41. Levine, D.S., Aparicio I.V.M.: Neural networks for knowledge representation and inference. Psychology Press, Rouledge (2013)
42. Mallot, H.A.: Artificial neural networks. In: Computational Neuroscience, Springer Series in Bio-/Neuroinformatics, vol. 2, pp. 83–112. Springer International Publishing, Berlin (2013)
43. Min, B.H., Park, C., Jang, I.S., Lee, H.Y., Chung, S.H., Kang, J.M.: Multi-objective history matching allowing for scale-difference and the interwell complication. doi:10.3997/2214-4609.20130172
44. BG, Mirta: Dynamics of complex systems and applications to SHS: models, concepts, methods. Leibniz-IMAG Laboratory, Grenoble (2004)
45. Morse, J.N.: Reducing the size of the nondominated set: pruning by clustering. Comput. Oper. Res. **7**(1–2), 55–66 (1980). doi:10.1016/0305-0548(80)90014-3
46. Mukhopadhyay, S., Panigrahi, P.K., Mitra, A., Bhattacharya, P., Sarkar, M., Das, P.: Optimized DHT-RBF model as replacement of ARMA-RBF model for wind power forecasting. In: 2013 International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), pp. 415–419. doi:10.1109/ICE-CCN.2013.6528534 (2013)
47. Nikdel, N., Nikdel, P., Badamchizadeh, M.A., Hassanzadeh, I.: Using neural network model predictive control for controlling shape memory alloy-based manipulator. IEEE Trans. Industr. Electron. **61**(3), 1394–1401 (2014). doi:10.1109/TIE.2013.2258292
48. Pendharkar, P.C.: A hybrid radial basis function and data envelopment analysis neural network for classification. Comput. Oper. Res. **38**(1), 256–266 (2011). doi:10.1016/j.cor.2010.05.001. (Project Management and Scheduling)
49. Poggio, T., Girosi, F.: Networks for approximation and learning. Proc. IEEE **78**(9), 1481–1497 (1990). doi:10.1109/5.58326
50. Prasad, K.V.R.B., Singru, P.M.: Optimum design of turbo-alternator using modified NSGA-II algorithm. In: Bansal, J.C., Singh, P., Deep, K., Pant, M., Nagar, A. (eds.) Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Advances in Intelligent Systems and Computing, vol. 202, pp. 253–264. Springer, India (2013)
51. Roberto, B., Ubaldo, C., Stefano, M., Roberto, I., Elisa, S., Paolo, M.: Graybox and adaptative dynamic neural network identification models to infer the steady state efficiency of solar thermal collectors starting from the transient condition. Sol. Energy **84**(6), 1027–1046 (2010)

52. Dos Santos Coelho, L., Ferreira da Cruz, L., Zanetti Freire, R.: Swim velocity profile identification by using a modified differential evolution method associated with RBF neural network. In: 2013 Third International Conference on Innovative Computing Technology (INTECH), pp. 389–395. doi:10.1109/INTECH.2013.6653721 (2013)
53. Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.-P. (eds.) (2000) A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. Parallel Problem Solving from Nature PPSN VI. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin (2000)
54. Sheikhan, M., Shahnazi, R., Garoucy, S.: Hyperchaos synchronization using PSO-optimized RBF-based controllers to improve security of communication systems. Neural Comput. Appl. **22**(5), 835–846 (2013). doi:10.1007/s00521-011-0774-4
55. Sheikhan, M., Shahnazi, R., Hemmati, E.: Adaptive active queue management controller for TCP communication networks using PSO-RBF models. Neural Comput. Appl. **22**(5), 933–945 (2013). doi:10.1007/s00521-011-0786-0
56. Syed, A.A., Pittner, A., Rethmeier, M., De, A.: Modeling of gas metal arc welding process using an analytically determined volumetric heat source. ISIJ Int. **53**(4), 698–703 (2013)
57. Tang, Y., Wong, W.K.: Distributed synchronization of coupled neural networks via randomly occurring control. IEEE Trans. Neural Netw. Learn. Syst. **24**(3), 435–447 (2013)
58. Teixidor, D., Grzenda, M., Bustillo, A., Ciurana, J.: Modeling pulsed laser micromachining of micro geometries using machine-learning techniques. J. Intell. Manuf. 1–14. doi:10.1007/s10845-013-0835-x (2013)
59. Whitley, D., Starkweather, T., Bogart, C.: Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Comput. **14**(3), 347–361 (1990). doi:10.1016/0167-8191(90)90086-O
60. Xia, D., Wang, L., Chai, T.: Neural-network-friction compensation-based energy swing-up control of pendubot. IEEE Trans. Industr. Electron. **61**(3), 1411–1423 (2014). doi:10.1109/TIE.2013.2262747
61. Xiao, Z., Liang, S., Wang, J., Chen, P., Yin, X., Zhang, L., Song, J.: Use of general regression neural networks for generating the GLASS leaf area index product from time-series MODIS surface reflectance. IEEE Trans. Geosci. Remote Sens. **52**(1), 209–223 (2014). doi:10.1109/TGRS.2013.2237780
62. Xu, Y., Zheng, J.: Identification of network traffic based on radial basis function neural network. In: Chen, R. (ed.) Intelligent Computing and Information Science. Communications in Computer and Information Science, vol. 134, pp. 173–179. Springer, Berlin (2011)
63. Yu, H., Xie, T., Paszczynski, S., Wilamowski, B.M.: Advantages of radial basis function networks for dynamic system design. IEEE Trans. Industr. Electron. **58**(12), 5438–5450 (2011). doi:10.1109/TIE.2011.2164773
64. Yuan, J., Yu, S.: Privacy preserving back-propagation neural network learning made practical with cloud computing. IEEE Trans. Parallel Distrib. Syst. **25**(1), 212–221 (2014). doi:10.1109/TPDS.2013.18
65. Zhang, H., Yang, F., Liu, X., Zhang, Q.: Stability analysis for neural networks with time-varying delay based on quadratic convex combination. IEEE Trans. Neural Netw. Learn. Syst. **24**(4), 513–521 (2013)
66. Zhi, C., Guo, L.H., Zhang, M.Y., Shi, Y.: Research on dynamic subspace divided BP neural network identification method of color space transform model. Adv. Mater. Res. **174**, 97–100 (2011)

Back-Propagation Neural Network for Gender Determination in Forensic Anthropology

Iis Afrianty, Dewi Nasien, Mohammed R.A. Kadir
and Habibollah Haron

Abstract Determination of gender is the foremost and important step of forensic anthropology in determining a positive identification from unidentified skeletal remains. Gender determination is the classification of an individual into one of two groups, male or female. The classification technique most used by anthropologists or researchers is traditional gender determination with applied linear approach, such as Discriminant Function Analysis (DFA). This paper proposed non-linear approach specific Back-Propagation Neural Network (BPNN) to determine gender from sacrum bone. Sacrum bone is one part of the body that is usually regarded as the most reliable indicator of sex. The data used in the experiment were taken from previous research, a total of 91 sacrum bones consisting of 34 females and 57 males. Method of measurement used is metric method which is measured based on six variables; real height, anterior length, anterior superior breadth, mid-ventral breadth, anterior posterior diameter of the base, and max-transverse diameter of the base. The objective of this paper is to examine and compare the degree of accuracy between previous research (DFA) and BPNN. There are two architectures of BPNN built for this case, namely [6; 6; 2] and [6; 12; 2]. The best average accuracy obtained by BPNN is model [6; 12; 2] with accuracy 99.030 % for training and 97.379 % for testing on experiment *lr* = 0.5 and *mc* = 0.9, then obtained Mean Squared Error (MSE) training is 0.01 and MSE testing is 1.660. Previous research

---

I. Afrianty (✉)
Faculty of Science and Technology, UIN Suska Riau, Pekanbaru 28124, Indonesia
e-mail: afrianty_iis@yahoo.com

D. Nasien · H. Haron
Faculty of Computing, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
e-mail: dewinasien@utm.my

H. Haron
e-mail: habib@utm.my

M.R.A. Kadir
Faculty of Bioscience and Medical Engineering, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
e-mail: afiq@fkm.utm.my

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_11

using DFA only obtained accuracy as high as 87 %. Hence, it can be concluded that BPNN provide classification accuracy higher than DFA for gender determination in forensic anthropology.

Keywords Forensic anthropology · Gender determination · Sacrum bones · Back-propagation neural network

1 Introduction

In the past few years, many issues have been raised concerning the many bone fragments or skeletal remains found but not identified with various conditions such as burns, dismemberment, dry bone, or simply not intact [2, 40]. These conditions cannot provide further information as to whether the fragments are derived from human or non-human remains. Usually the cases related to research into human skeletons from ancient times. However, lately the findings of human skeletons is in the criminal context. Several criminal cases have included the discovery of bodies and human skeletons without viable means of identification. These problems are usually handled by forensic anthropologist or researchers in forensic anthropology field. Forensic anthropologists often bemoan the major issues in forensic anthropology namely the process of identifying the bones or skeletal remains of the unknown. These cases have always been a challenge for forensic anthropologist in order to recognize and identify the skeletal remains. The skeleton or skeletal remains that have been found should be identified and established the cause and manner of death.

Identification of skeletal remains (like Fig. 1) is a mainstay in forensic anthropology. The ability of the forensic anthropologist to undertake analysis is fundamentally determined by the preservation of the skeletal remains to identify and its biological profile uncovered [44]. Forensic anthropology is one of the fastest growing medico-legal disciplines whose main goal is to identify the biological profile of unknown skeletal remains, including the cause and time of death [19]. As a discipline, forensic anthropology must see its responsibilities through from the scene to the courtroom. Forensic anthropology is a relatively young subfield within biological anthropology with a focus on biological profile identification leaving the task of building detailed biological profiles of human skeletal remains vacant [6].

The most important component of identification of biological profile of an individual is gender determination. However, gender determination is the first essential step in the positive identification of skeletal remains. Knowledge of the gender of an unknown set of remains is essential to make a more accurate estimation of age [24]. Without an accurate determination of gender, there can be no accurate determination or estimation of age at death. Thus, gender determination is a necessary to process of identification which also includes the Big Fours parameters.

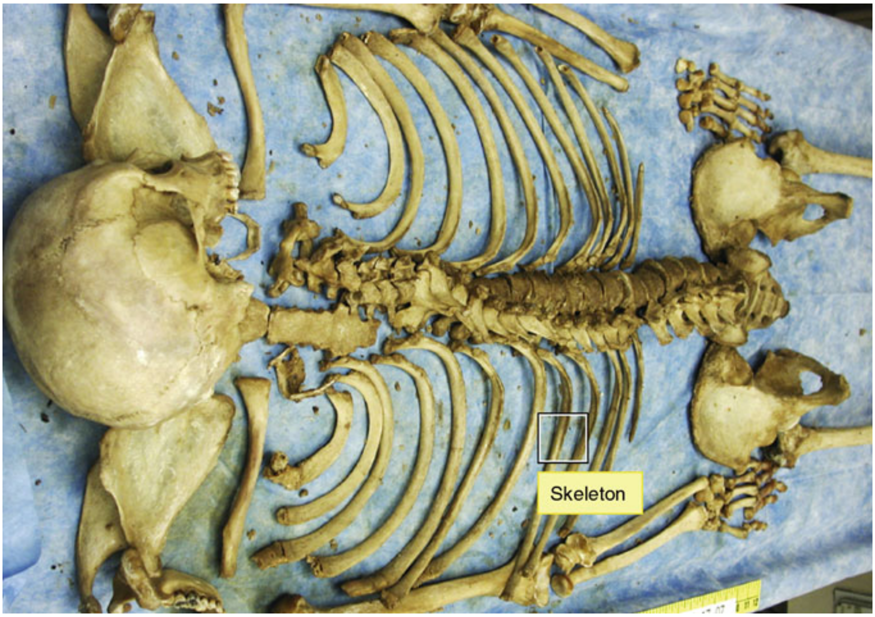


Fig. 1 The sample of male skeletal remains [26]

To assist determine gender tools or classification techniques are used that can provide more accurate information about the biological profile of an individual. The condition of skeleton is complete and available while identification will obtain a more accurate result in gender determination [10, 45]. From previous studies, many researchers have used linear approach such as Discriminant Function Analysis (DFA). DFA or DA is a method used to find a set of axes that possess the greatest possible ability to discriminate between two or more groups [15]. In previous studies, researchers have been using many parts of the skeleton for identification and one of the skeletal elements used as an indicator of gender is the pelvic bones. Using the same data from previous studies, this paper will use another classification technique from non-linear approach that apply Artificial Neural Network model, namely Back-Propagation Neural Network (BPNN).

BPNN is a classical domain-dependent technique for supervised training [4]. The purpose of this paper is to determine gender using BPNN from a part of pelvic bones namely sacrum bones, and to compare classification accuracy of the results obtained by BPNN with previous techniques (DFA). This paper is structured as follows: Sect. 2 will review and discuss the related work; Sect. 3 gives an overview of proposed technique to determine gender from Neural Network model, namely BPNN technique; Sect. 4 described the research methodology which will be developed and explain key functions of the proposed technique; Sect. 5 is discussion that explain data acquisition in gender determination process and showed the result obtained; and finally, a conclusion of this paper is provided in Sect. 6.

2 Related Work

Anthropology is the study about the human biological, cultural and linguistic conditions. Anthropology is divided into two main branches, namely cultural and physical [26]. Forensic anthropology is closely related with physical or biological anthropology that work through identification of skeletal remains. Forensic anthropology is disciplines of physical or biological anthropology that the fastest growing [19, 26]. The main objective of forensic anthropology is to identify skeletal remains and thus generate a biological profile of the individual. Following the biological profile, anthropologists or researchers will endeavor to provide a personal identification of the remains based on evidence, any distinguishing characteristics the individual may display, and determine whether remains derived are human or non-human. The biological profile includes gender, age, race (ancestry), and stature, also known as the "Big Four" parameters of forensic anthropology [29, 35]. The first step for positive identification when dismembered or decomposed bodies are recovered is gender determination [28, 50]. Identification then proceeds toward the determination of age, race, and stature. In other words, gender determination is necessary to identify age, ancestry, and stature.

From previous researches, identification of skeletal remains might be done by fingerprint, anthropological, dental, DNA analysis at laboratory or radiological examinations [55]. However, the most popular method for identification of gender is DNA analysis. In some cases, where the bones are burned, dismembered, or very dry, DNA analysis has failed because suitable DNA cannot be extracted under the conditions mentioned and not recoverable from remains in all circumstances [6, 49]. Thus, protein analysis or the study of the microscopic structure of the fragment may be useful, and, at times, the only applicable method [6]. DNA analysis has been developed to provide accurate gender determination. For gender determination cases, DNA analysis cannot replace the anthropological analysis because it cannot provide data on some of the important parameters of the biological profile [7]. Moreover a thorough anthropological analysis is conducted to obtain a more reliable characterization of the individual, providing more data to confirm identity [7]. Therefore, due to the drawbacks of DNA analysis, forensic anthropology has been developed in order to improve previous identification methods of profiling unknown remains, particularly in gender determination. Forensic anthropology assists in creating a biological profile including determination of gender, age, race, and stature, also known as the "Big Four" parameters of forensic anthropology [29]. The contribution of forensic anthropology is often important during the investigation and the interpretation of decomposing human remains [26].

Gender builds based on biological sex. Gender is the very process of creating a dichotomy by effacing similarity and elaborating on difference, which gender is related to biology. In general, gender determination is an important part of the forensic process and gender determination will more reliable if the skeleton is complete and in good condition. The purpose of gender determination is to identify human skeletal remains in order to know the difference between male and female

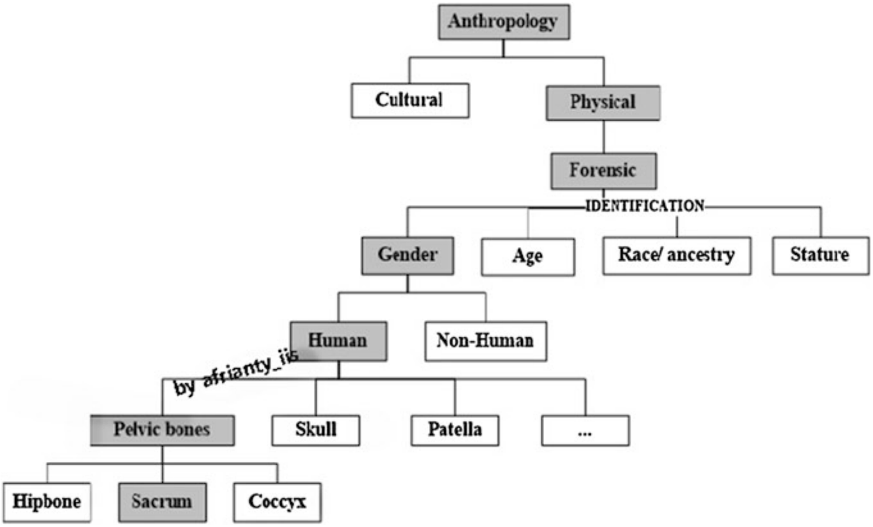


Fig. 2 The hierarchy of above summaries of forensic anthropology

based on a series of parameters or variables. In other words, gender determination is the classification of individuals into two groups, male or female. In previous studies, researchers have used many parts of the skeleton for identification, such as skull [17, 45], patella [3, 22, 36], long bones [39, 41, 43]. However, one of the skeletal elements used as sex indicator recently with near 100 % accuracy is the use of pelvic bones, specifically sacrum bones. From the above explanation, a hierarchy can be developed as seen in Fig. 2.

2.1 Historical Development of Forensic Anthropology

Development of forensic anthropology has evolution that always developed. The evolution of forensic anthropology extends back to nineteenth century when anatomists and physical anthropologist were occasionally asked to provide their assistance in human identification. Traditionally, the forensic anthropology analyzes human remains in full or semi-skeletonized [26]. In many instances, the researchers is asked to decide whether the remains of the skeleton were found, identified whether it comes from a human or animal, to assess the biological characteristics, and to help in a positive identification. In addition, the post-mortem interval, ante-mortem and pathology of trauma, and post-mortem artifact linked to the scene determined [26]. For about 60 years anthropologists have participated in forensic research and assisted law enforcement personnel in solving crime [11].

The development of forensic anthropology has begun on 18th century [16]. Development of forensic anthropology happened in many state, like Europe and

United State. Review of paper's [16] distinguished three periods about the development of history of forensic anthropology: pre-1939, 1939–1972, and post-1972. While [53] distinguished four periods of forensic anthropology development which is based on milestone in the publication document and the research:

1. Early 18th to last quarter of the 19th century
2. In 1878 until 1939 year
3. In post-World War II until the last quarter of the 20th century
4. In 1972 until present-day because many the researchers still research about it (post-1972).

Their paper discussed about identification from skeletal remains and researched parts of bones were found, such as research involving estimates of gender, age, ancestry, and stature. Based on previous reviews, can be summarized that period of forensic anthropology development may be distinguished in three periods: on 18th century, 19th century, and 20th century.

Before the late 18th century, skeletal analysis within a forensic context was mostly an applied area of anatomy so that anatomists and physicians could use general knowledge, the few techniques that existed in textbooks, and their experience [54]. In 19th century, the anthropologists were increasing demanded by medico-legal establishments thus can render aid in the case of skeletonized (i.e. for identification human remains in World War II the Korean War) [35]. Before 1939, the principal contributors to the methodology of human skeletal variation, using a collection of bodies known age, heredity, gender, and morbidity [16]. In the late 19th century and began to enter the 20th century, forensic anthropology has developed into modern period, namely the establishing of the "Physical Anthropology Section of the American Academy of Forensic Sciences" in 1971 and anthropologists have applied modern techniques in solving of the cases [13, 35].

Recent developments place forensic anthropology in a wider criminal investigation context, researchers are even asked to aid in the identification of living individuals [26]. In forensic anthropology covers a variety of topics and issues. Anthropologists no longer limited to research involving the estimation of determine gender, age, ancestry, and stature. Forensic anthropology worldwide, there seem to be considerable differences in many aspects of education, training, professional status, research activities and job opportunities globally [26]. However, research is still continuously improved and developed with the involvement of a variety of techniques to assist research [16].

2.2 *Identification of Biological Profile in Forensic Anthropology*

Identification analysis in forensic anthropology is based on two foremost goals, namely establishing a profile of individual represented that will assist in positive identification and recognition and interpretation of evidence of foul play [54]. The biological profile includes gender, age, race (ancestry), and stature, also known as the "Big Four" parameters of forensic anthropology [29, 35].

1. Gender determination
   Gender determination is the classification of an individual, whether as male or female [12]. Gender determination is based on skeletal features plays a crucial role in legal medicine and forensic anthropology [18]. Many researchers have done research to gender determination using several of parts the body such as [14, 23, 34].
2. Age determination
   Knowledge of the gender of an unknown set of remains is essential to make a more accurate estimation of age [24]. The determination of age can be estimated of progression of the skeletal maturity.
3. Race (ancestry) determination
   Ancestry determination is important component of a skeletal profile that very difficult to estimate [35]. Ancestry estimates are based on two features, namely the shape or morphology and the metric analysis various elements of the skeleton.
4. Stature determination
   Determination of stature is an important aspect in establishing identity. The determination of stature standards are based on two major methods namely the anatomical and mathematical (regression) method [1, 35]. The anatomy method requires the presence of a complete or near complete skeleton thus provide the best approximation of stature. But, it has main drawbacks which it requires a complete skeleton and the addition of correction factors to compensate for soft tissues [1]. Whereas, the mathematical method is used for analyzing an incomplete skeleton such as single or body part, which the lengths of target bones are regressed upon stature [35]. That method was using regression equation or multiplication factors based on the correlation of individual measurements of bones to living statures. It has disadvantage that its predictive ability is less accurate because of wide variability in population body proportions [1, 35].

2.3 Materials

Pelvic bones consist of a pair of hipbones, sacrum, and coccyx. The pelvic is another element of the skeleton that exhibits sexual dimorphism [35]. It is considered the most sexually dimorphic skeletal element because the female pelvic must accommodate the relatively large head of an infant during childbirth. Hence, the female pelvic is typically wider in every dimension than the male pelvic [51]. Sacrum bones are a part of the pelvic bones that are related to reproduction and

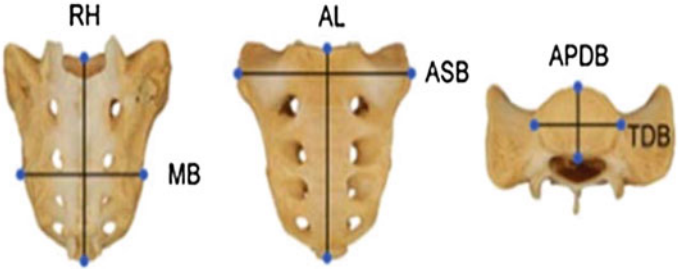


Fig. 3 The six variables for sacrum bones by metric measurement [14]

Table 1 The variables for measurement of sacrum bones [14]

Description of Variables	Variables
Real height	RH
Anterior length	AL
Anterior superior breadth	ASB
Mid-ventral breadth	MB
Anterior posterior diameter of the base	APDB
Max-transverse diameter of the base	TDB

fertility. The sacrum could well be thought to share significant qualities with the reproductive organs, and even to transport material from the brain to those organs. Hence, sacrum bones can be up to 100 % accurate as an indicator of sex if all of the required data is complete. Some of researchers have conducted a study of the pelvic bone in gender determination, such as [14, 15, 56].

The data used in this paper included as many as 91 sacrum bones consisting of 34 females and 57 males derived from analysis of previous researches, namely [14]. The collected data was then measured using metric measurement (can be seen in Fig. 3).

There are six measurements for sacrum bones that are used as indicators or variables in determining gender. They are real height, anterior length, anterior superior breadth, mid-ventral breadth, anterior posterior diameter of the base, and max-transverse diameter of the base. The variables for measurement of sacrum bones with its code respectively, can be seen in the following Table 1.

2.4 Previous Technique

From the review of previous studies about forensic anthropology, it can be summarized that the processes of gender determination are traditionally divided into two processes. The first is measurement of data collected and the second process is selection of classification technique used to determine gender. Previously, using classification techniques, the data collected must be measured to facilitate data analysis. Traditionally, evidence was collected and then identified by measuring data based on defined variables. The measurement of the data can be divided into two categories, metric and morphology measurement [8]. Metric measurement is concerned with size, weight, and proportion of the human body and skeleton, such as pubic angle, pubic length, and so on. The science that deals with metrics in anthropology is known as Anthropometry [8, 21]. Morphology measurement is concerned with observations of visual criteria, such as sub-pubic angle, and so on. The science that studies morphology is called Anthroposcopy [21, 47].

The second process is concerned with selection technique for gender determination. Many anthropologists used tools and techniques for facilitating the process of determining gender. To help determine gender, classification techniques are used that can provide more accurate information about the biological profile of the individual. Many researchers haves done research into techniques for gender determination. The most popular techniques used by previous researchers using linear approach. The one of techniques included in linear approach and the most popular is used in gender determination is Discriminant Function Analysis (DFA) [8, 14, 15, 17, 18, 23, 34, 39, 45, 46, 50].

DFA or DA is one of the linear approach that a statistical technique used to find a set of axes that possess the greatest possible ability to discriminate between two or more groups and it does not necessarily "understand" biological differences [15, 46]. DFA tends to magnify the differences between predefined groups is used as classifier in gender determination [17]. The objective of DFA is easily applicable method without the need for specific skill sets, other than knowledge of the definitions of traditional anthropological measurements [17]. In addition to objective of DFA are very easy to use technique, an economic, robust, easy-to-use modeling technique [9]. The main advantage of DFA reduces subjective judgment as well as the level of expertise and experience needed for the determination of gender [38]. Although DFA is very easy to use but it is also quite constraining. Disadvantages of DFA are implied in paper [9] which they have done comparison of technique to obtain more accuracy rate. Based on previous reviews, application of DFA has disadvantage because requires meeting three main assumptions which sometimes make it is difficult to meet. They are the explanatory variables within each group must follow a multivariate normal distribution [9]; the variance–covariance matrices of the groups must be equal that is to say the variance of each variable must be similar in each group; and the correlation between explanatory variables must be as low as possible. The assumptions described are sometimes difficult to meet. The assumption of linearity between function output and the input variables does not always apply. The groups being considered are often non-linearly separable.

To improve the lack of DFA described above, then by using the same data this paper will use development of another classification technique from non-linear approach, namely Artificial Neural Network (ANN) specifically Back-Propagation

Neural Network (BPNN). Unlike DFA, ANN does not require distributional assumptions of the variables and is able to model all types of non-linear functions between input and output of a model [9].

3 Proposed Technique

In this paper, proposed Artificial Neural Network (ANN) technique specific BPNN for gender determination. ANN is a characteristic in biological Neural Network (NN) in which it contains an information processing system modeled on the structure of the dynamic process that are composed of simple elements operating in parallel [37]. ANN consists of a number of neurons that are linked with the neurons in the human brain. ANN can be classified into feed forward and recurrent, depend on their connectivity. The ability of an ANN to predict outcomes accurately depend on the selection of proper weights during the training. Training or learning is the relationship between inputs and target. The rule of the learning defined as a procedure of a network aims to adjust weights and biases [5]. Its learning rule uses the most rapid descent to continuously adjust the weights and thresholds of neural network through back propagation, so that the sum of squared error of network is minimum [20]. Three learning of neural network methods are supervised, unsupervised learning, and reinforced learning [30]. In supervised learning, the network is provided with inputs and desired outputs or target values. In unsupervised learning, on the other hand, the weights and biases are modified in response to network inputs only. The performance of the models is measured using Mean Squared Error (MSE). MSE is the average of the squares of the difference between each output and the desired output, given by equation (Eq. 1) below:

$$\mathrm{MSE} = \frac{1}{n}\sum_{\mathrm{i}=1}^{n}\sum_{j=1}^{yd}(y_j^i - \hat{y}_j^i)^2 \tag{1}$$

ANN is learned using the back propagation algorithm in which the errors for the units of the hidden layer are determined by back propagating (BP) the errors of the units of the output layer [25]. There are two processes involved in a BPNN namely input signal and error signal. Input signal presented to input layer and continued till produced the output layer. On the contrary, error signal is caused the different from the desired output then error is calculated after that propagated backwards from the output layer to input layer. Back Propagation Neural Network (BPNN) is currently the most widely used algorithm for supervised learning with multilayer feedforward networks [37]. It works by measuring the output error, calculating the gradient of this error, and adjusting the ANN weights (and biases) in the descending gradient direction. Hence, BPNN is a gradient-descent local search procedure (expected to stagnate in local optima in complex landscapes) [4]. BPNN due to its good robustness and fault tolerance is widely used in optimization and function

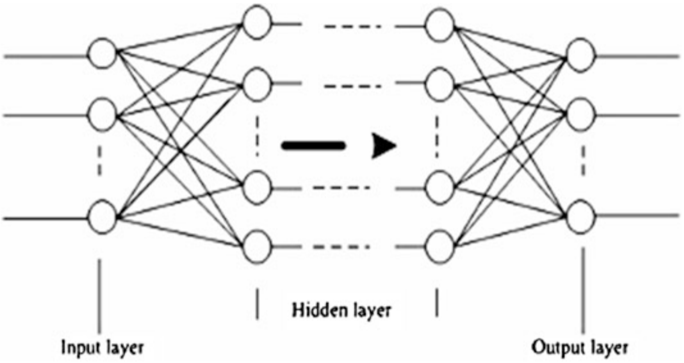


Fig. 4 The structure of BPNN [32]

approximation [52]. BPNN are adjusted or trained to establish a required path/trend, so that a particular input leads to a specific target output [37].

BPNN consists of [31]:

1. An input (I) layer with nodes representing input variables to the problem.
2. An output (O) layer with nodes representing the dependent variables, and
3. One or more hidden (H) layers containing nodes/neurons to help capture the nonlinearity in the data [33]. The neurons between layers can be fully or partially interconnected between layers with weight (*w*). The procedure of the BPNN repeatedly adjusts the weights of the connections in the network so as to minimize the measure of the difference between the actual output vector of the net and the desired output vector.

The structure of BPNN can be seen in Fig. 4.

The number of inputs to the network is limited by the problem, and the number of neurons in the output layer is also limited by the amount of output required by the problem. The number of hidden layers and layer sizes are designed in accordance with specific problems. However, the number of neurons in the hidden layer varied initially and to determine the optimum combination depends on the period of training and performance errors [42]. Each neuron has an internal state known as activation function. Activation function is applied to the input of a neuron, determines the output of that neuron. Many activation functions have been tested, but only a few have found practical applications including step function, sign function, sigmoid function, and linear function. Sigmoid function usually is used in the process of Back-Propagation Neural Networks.

BPNN algorithms are [48]:

1. Normalize the inputs and outputs with respect to their maximum values (between 0 and 1). For each training pair, assume the input is "l" input given by I_I and n outputs O_o in a normalized form.
2. Assume the number of neurons in the hidden layer to lie between $l < m < 2\ l$. Where l is input layer and m is hidden layer.

3. [V] Represents the weights of synapses connecting input neurons and hidden neurons and [W] represents weights of synapses connecting hidden neurons and output neurons. Initialize the weight to small random values usually −1–1. For general problems, λ can be assumed as 1 and the threshold values can be taken as 0.

$$\begin{aligned} [\mathrm{V}]^0 &= [\text{random weight}] \\ [W]^0 &= [\text{random weight}] \\ [\Delta V]^0 &= [\Delta W]^0 = [O] \end{aligned} \tag{2}$$

A common initial weight is to set the range between −0.5 and +0.5 for large database and −0.35 and +0.35 for small database.

4. Present one set of inputs and outputs for the training data. Present the pattern to the input layer I_I as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\begin{array}{ccc} \mathrm{O_I} & = & \mathrm{I_I} \\ l \times 1 & & l \times 1 \end{array} \tag{3}$$

5. Compute the inputs to the hidden layer by multiplying corresponding weights of synapses as

$$\begin{array}{ccc} \mathrm{I_H} & = [\mathrm{V}]^{\mathrm{T}} & \mathrm{O_I} \\ \mathrm{m} \times 1 & \mathrm{m} \times 1 & 1 \times 1 \end{array} \tag{4}$$

6. Let the hidden layer units evaluate the output using sigmoidal function as

$$\mathrm{O_H} = \underset{\mathrm{m} \times 1}{\left\{ \begin{array}{c} \cdot \\ \cdot \\ \frac{1}{(1+e^{-I_{Hi}}} \\ \cdot \\ \cdot \end{array} \right\}} \tag{5}$$

7. Compute the inputs to the output layer by multiplying corresponding weights of synapses as

$$\begin{array}{ccc} I_O & = [W]^T & O_H \\ \mathrm{n} \times 1 & \mathrm{n} \times \mathrm{m} & \mathrm{m} \times 1 \end{array} \tag{6}$$

8. Let the output layer units evaluate the output using sigmoidal function as

$$\mathrm{O_H} = \left\{ \begin{array}{c} \cdot \\ \cdot \\ \frac{1}{(1+\mathrm{e}^{-I_{oj}}} \\ \cdot \\ \cdot \end{array} \right\} \tag{7}$$

9. Calculate the error and difference between the network output and desired output as for *i*th training set as

$$E^p = \frac{\sqrt{\sum (T_j - O_{oj})^2}}{n} \tag{8}$$

10. Find d as

$$\mathrm{d} = \left\{ \begin{array}{c} \cdot \\ \cdot \\ (Tk - Ook)Ook(1 - Ook) \\ \cdot \\ \cdot \end{array} \right\} \tag{9}$$

11. Find $[Y]$ matrix as

$$\begin{array}{ccc} [Y] = & O_H & \langle d \rangle \\ \mathrm{m} \times \mathrm{n} & \mathrm{m} \times 1 & 1 \times \mathrm{n} \end{array} \tag{10}$$

12. Find

$$\begin{array}{ccc} [\Delta W]^{\mathrm{t+1}} = & \alpha\, [\Delta W]^{\mathrm{t}} + & \eta\, [Y] \\ \mathrm{m} \times \mathrm{n} & \mathrm{m} \times \mathrm{n} & \mathrm{m} \times \mathrm{n} \end{array} \tag{11}$$

13. Find e

$$\begin{array}{ccc} e = & [W] & d \\ \mathrm{m} \times 1 & \mathrm{m} \times \mathrm{n} & 1 \times 1 \end{array} \tag{12}$$

$$\begin{array}{c} \mathrm{d}^* = \left\{ \begin{array}{c} . \\ . \\ e_i \langle O_{Hi} \rangle \langle 1 - O_{Hi} \rangle \\ . \\ . \end{array} \right\} \\ \mathrm{m} \times 1 \quad \mathrm{m} \times 1 \end{array} \tag{13}$$

14. Find $[X]$ matrix as

$$\begin{array}{c} [X] = O_I \langle d^* \rangle = \quad I_I \langle d^* \rangle \\ 1 \times \mathrm{m} \quad 1 \times 1 \quad 1 \times \mathrm{m} \quad 1 \times 1 \quad 1 \times \mathrm{m} \end{array} \tag{14}$$

15. Find $[\Delta V]^{\mathrm{t+1}}$

$$\begin{array}{c} [\Delta V]^{\mathrm{t+1}} = \alpha \, [\Delta V]^{\mathrm{t}} + \eta \, [X] \\ l \times m \quad l \times m \quad l \times m \end{array} \tag{15}$$

16. Find

$$\begin{aligned} [V]^{\mathrm{t+1}} &= [V]^{\mathrm{t}} + [\Delta V]^{\mathrm{t+1}} \\ [W]^{\mathrm{t+1}} &= [W]^{\mathrm{t}} + [\Delta W]^{\mathrm{t+1}} \end{aligned} \tag{16}$$

17. Find error rate as

$$\text{Error rate } = \frac{\sum Ep}{nset} \tag{17}$$

18. Repeat steps iv–xvi until convergence in the error is less than tolerance value.

The neural network methodology is well known by its ability for generalization, its massive parallel processing power and its high nonlinearity, making it perfect for gender estimation [36].

4 Research Methodology

This research describes the research methodology that explains the activities and the detailed phases of the process of gender determination using classification techniques. This research framework is composed of five main phases as follows:

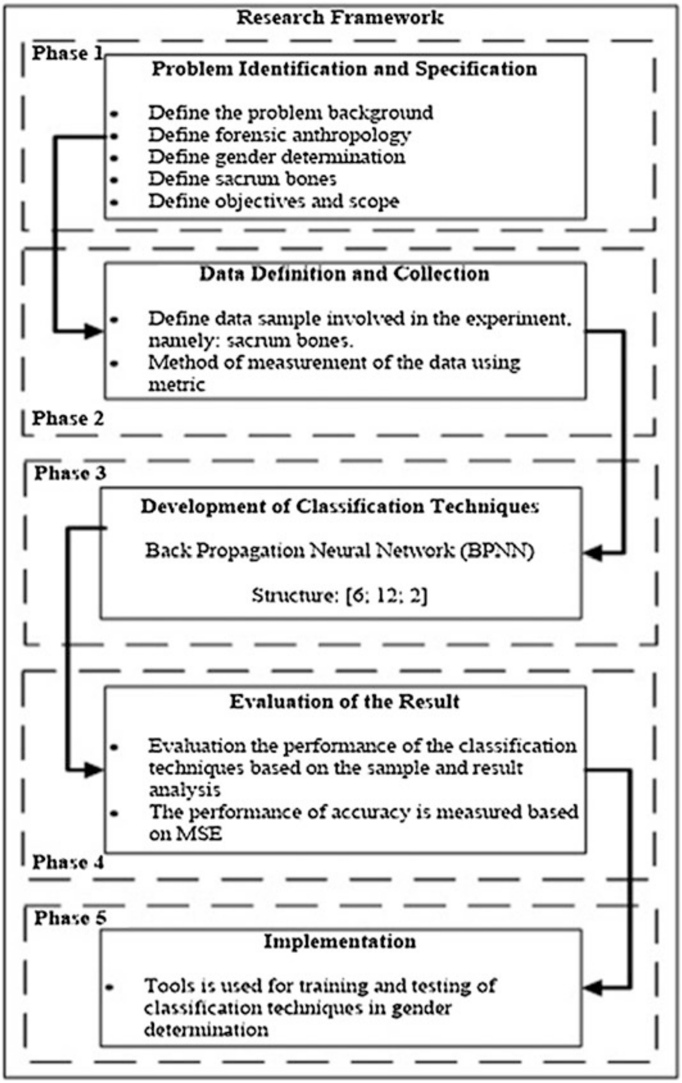

Research Framework
Phase 1
Problem Identification and Specification
Define the problem background
Define forensic anthropology
Define gender determination
Define sacrum bones
Define objectives and scope
Data Definition and Collection
Define data sample involved in the experiment, namely: sacrum bones.
Method of measurement of the data using metric
Phase 2
Phase 3
Development of Classification Techniques
Back Propagation Neural Network (BPNN)
Structure: [6: 12: 2]
Evaluation of the Result
Evaluation the performance of the classification techniques based on the sample and result analysis
The performance of accuracy is measured based on MSE
Phase 4
Phase 5
Implementation
Tools is used for training and testing of classification techniques in gender determination

Fig. 5 Research framework of the case

1. Problem identification and specification
2. Data definition and collection
3. Classification techniques
4. Evaluation of the result
5. Implementation

The process of research framework is illustrated in Fig. 5.

The framework of this paper has five phases and each phase represents the problems in this research.

1. The **first phase** is problem identification and specification

In this phase includes the problem background, forensic anthropology specifically identification of skeletal remains, gender determination, sacrum bones, objective and scope in this research.

Identification of the problem statement began with a literature review on issues related to gender determination based on sacrum bones in forensic anthropology. There are three major issues related to the problems namely Forensic Anthropology, gender determination, and development of classification technique. After the literature review, then define problem solving of this research which is examined thus the objective can be determined to answer the problem statement. The process of problem identification is done by referring to the previous literatures in published paper and journals. For constraints of work, the scope should be defined in accordance with a predetermined objective. The scope of this research is limited to explaining the process of BPNN in gender determination based on six measurements as variables of sacrum bones.

2. The **second phase** is data definition and collection.

Data definition means a process of defining the type of data used, deciding the source of data, categorizing the data for testing and training. The data is used is sample data of sacrum bones. Otherwise, data collection is having the input data built from original source and gathered into a compilation of numbers of input that relevant information toward validating and analyzing the algorithm. In data collection for the purpose of validating and analyzing the algorithm, a function is created in the Matrix Laboratory (MATLAB) program. For analyzing of the data collection is measured using metric measurement. The data type and sources that used in this paper are sacrum bones dataset in tables form. The dataset is obtained from analysis the previous research.

3. The **third phase** is explaining the classification technique

The classification technique is used in the process of gender determination, namely Back Propagation neural Network (BPNN). This phase is initiated with variables linked with numerous data in the process of gender determination. The variables will be used as nodes in input layer of BPNN. Then, BPNN will do learning appropriate to the structures and parameters which have been decided, such as determining number of neurons in hidden layer, learning rate (*lr*), momentum (*mc*), and activation function.

4. The **fourth phase** is evaluation of the result

The evaluation of the result specifically an evaluation of the performance accuracy of the classification techniques based on the sample data and result analysis produced in classification phase. The performance of the models is measured using mean squared error (MSE). The result obtained by algorithm is

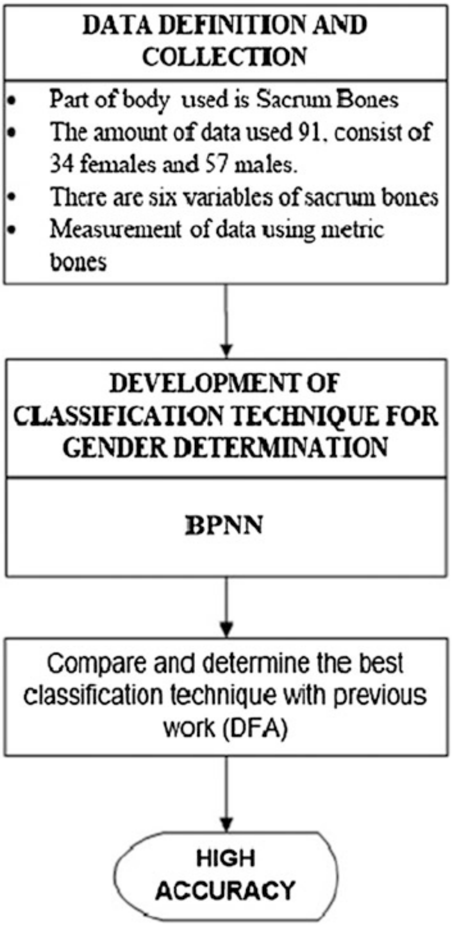

DATA DEFINITION AND COLLECTION
Part of body used is Sacrum Bones
The amount of data used 91, consist of 34 females and 57 males.
There are six variables of sacrum bones
Measurement of data using metric bones
DEVELOPMENT OF CLASSIFICATION TECHNIQUE FOR GENDER DETERMINATION
BPNN
Compare and determine the best classification technique with previous work (DFA)
HIGH ACCURACY

Fig. 6 The detail of the research framework

recorded in table form then copied into Excel software for graph plotting. Based on the table and graph, manual analysis by observation on the output of algorithm is conducted to give the conclusion on the best technique for gender determination. Based on the experiment of the classification technique, the result is recognition accuracy for each algorithm when analyzing input data. In analyzing the classification result namely accuracy is recorded manually and copied into Excel table. After each testing session, the graph is plotted using standard functions such as Plot Graph in Excel software. Manual graph analysis is done, thus it can be found the best classifier to gender determination based on the high accuracy. A detail of the research framework in this paper is described in Fig. 6.

5. The last phase, **fifth phase** is implementation.

The implementation includes discussion of the tools that required in the fourth phase. The requirements to develop the integrated system are categorized into two parts; hardware and software. The hardware specification is used in this paper is Compaq Presario CQ41 with Intel® Core™ i3 processor; its operating system is Windows 8 32-bit with 2 GB RAM memory. The software required for analyzed data is Matrix Laboratory (MATLAB) program; MATLAB 2012a that is used as platform to write the code for BPNN.

5 Discussion

The recovery process of fragmentary skeletal remains in forensic anthropology requires easy and rapid techniques for biological profiling and reconstruction of the scene history [27]. The first and most vital biological characteristic under consideration is sex since it reduces the number of possible matches in the population by 50 % [27]. Although identification of gender can be easily established when a complete skeleton is present, this is rarely the case in forensic anthropology where mostly fragmented bony parts are recovered.

The amount data used is 91 sacrum bones consisting of 34 females and 57 males. The data used derived from the result of the analysis of the mean, Standard Deviation (SD), minimum, and maximum value of previous study [14] which only provided information about mean, SD, minimum, and maximum value. From that information can be obtained the simulation of data used according mentioned. The data collection measured using metric method. Data measured and saved in table form Excel. After measurement, then calculated use BPNN was developed in MATLAB R2012a. Data must be normalized and divided into data training and data testing. The data is divided namely 70 % for training and 30 % for testing from overall data. In step of this discussion, just explain techniques BPNN. The DFA technique is not described, because it is only used as a comparison of the final results only.

As a first step, the architecture of the network has to be decided. The architecture of BPNN for case divided into two models, namely [6; 6; 2] and [6; 12; 2]. The architectures in this research are shown in Figs. 7 and 8.

Figures 7 and 8 demonstrate that the architecture of BPNN used of this case consisted of six inputs based on the variables of sacrum bones (Real Height, Anterior Length, Anterior Superior Breadth, Mid-Ventral Breadth, Anterior Posterior Diameter of the Base, and Max-Transverse Diameter of the Base). On hidden layer using six neurons, whereas on hidden layer Fig. 7 using formula $2n$ (n = *input*), thus hidden layer consisted of 12 neurons. The output layer consisted of two neurons, namely female and male. After the layers have been designed, the process of calculation of BPNN is developed by coding in MATLAB R2012a.

Before learning process, parameters to be used must be defined. In this research, learning process was stopped after 100,000 iteration epochs using log-sigmoid for activation function, momentum (*mc*) was 0.1; 0.5; 0.9 and learning rate (*lr*) was 0.1; 0.5; 0.9 (like Table 2). Computing error in the output layer was back propagated to

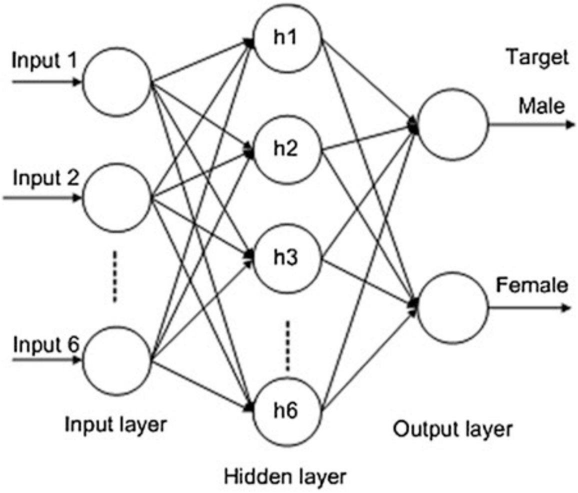


Fig. 7 The architecture of BPNN with hidden layer 6 neurons

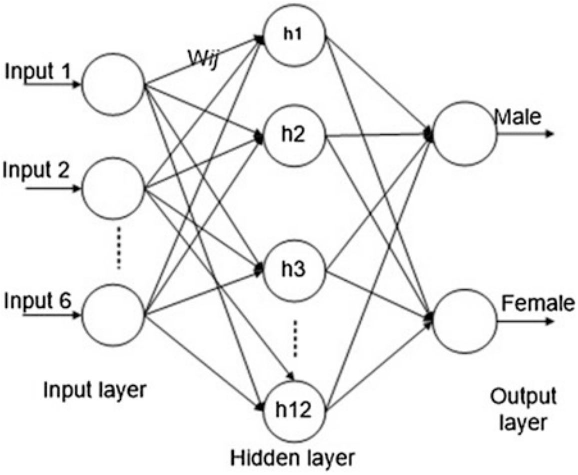


Fig. 8 The architecture of BPNN with hidden layer 12 neurons

earlier ones in order to update the current input-hidden layer weights and hidden-output layer weights. By updating these weights, the network would learn to reach the target. The target reached is 1 for female and 0 for male. In the algorithm, the error was calculated in the output and the new values of weights were computed in each layer until the error was minimized to a considerable value. The measurement of ANN performance was observed by using the MSE and total prediction accuracy of the network to the tested data. And training is best when the ANN is capable to achieve lowest MSE value.

Table 2 Parameters learning rate (*lr*) toward momentum (*mc*) in BPNN

No.	lr	mc
1	0.1	0.1
2		0.5
3		0.9
4	0.5	0.1
5		0.5
6		0.9
7	0.9	0.1
8		0.5
9		0.9

The parameters used in Matlab2012a as follow.

```
net.trainParam.show = 500; %Epochs between displays
net.trainParam.epochs = 100000;%Maximum number of epochs to
train
net.trainParam.goal = 0.01; %Performance goal
net.trainParam.max_fail = 5; % Maximum validation failures
net.trainParam.min_grad = 1e-10; %Minimum performance
gradient
net.trainParam.time = inf;
net.trainParam.lr = 0.1; %Learning rate (0.1; 0.5; 0.9)
net.trainParam.mc = 0.1; %momentum (0.1; 0.5; 0.9)
```

Table 3 Experimental result of BPNN for training and testing [6; 6; 2]

Performance training	lr	mc		
		0.1	0.5	0.9
Accuracy (%)	0.1	97.259	97.581	97.901
MSE		0.012	0.010	0.011
Accuracy (%)	0.5	97.260	97.582	97.901
MSE		0.012	0.011	0.010
Accuracy (%)	0.9	96.785	**97.740**	97.102
MSE		0.014	0.010	0.010
Performance testing				
Accuracy (%)	0.1	96.503	96.282	96.767
MSE		0.012	0.010	0.011
Accuracy (%)	0.5	96.370	96.297	96.733
MSE		0.012	0.011	0.010
Accuracy (%)	0.9	95.879	**96.422**	95.611
MSE		0.014	0.010	0.010

Table 4 Experimental results of BPNN for training and testing [6; 12; 2]

Performance of training	lr	mc		
		0.1	0.5	0.9
Accuracy (%)	0.1	97.900	98.870	98.500
MSE		0.012	0.010	0.011
Accuracy (%)	0.5	97.740	98.549	**99.030**
MSE		0.012	0.011	0.010
Accuracy (%)	0.9	98.390	98.549	98.390
MSE		0.014	0.010	0.010
Performance of testing				
Accuracy (%)	0.1	96.790	96.927	97.200
MSE		1.120	1.952	1.356
Accuracy (%)	0.5	96.805	97.044	**97.379**
MSE		0.954	1.515	1.660
Accuracy (%)	0.9	96.720	96.855	96.850
MSE		1.683	1.704	1.547

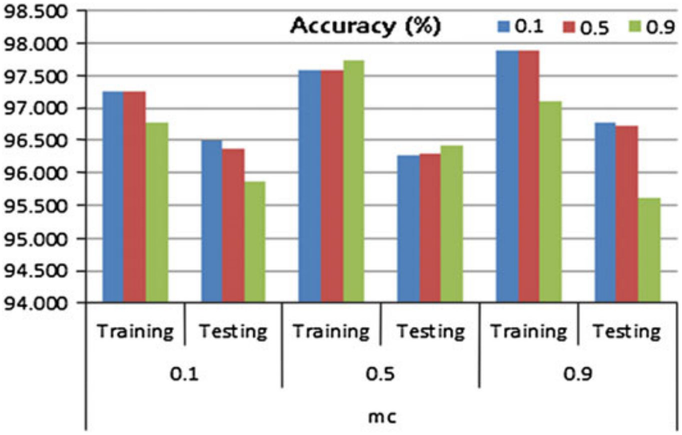


Fig. 9 The performance of structure [6; 6; 2]

Determination of average accuracy of male and female can be visualized as a confusion matrix, where each column represents the predicted instances of a character, while each row represents the actual instances of an output. A confusion matrix is a visualization tool typically used in supervised learning. Supervised learning is where definitions of classes are done by researcher and learning process is based on training data, formed by pairs of input object and desired output.

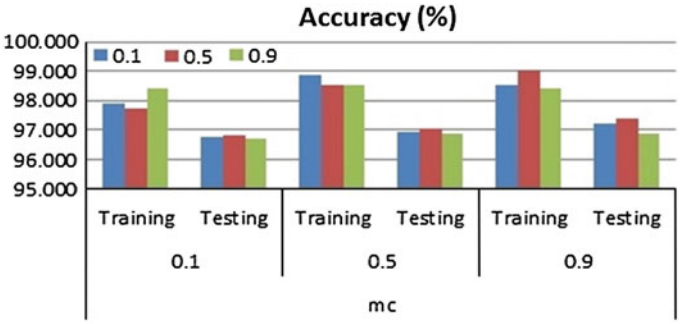


Fig. 10 The performance of structure [6; 12; 2]

Table 5 The comparison advantages and disadvantages between DFA and ANN

Classification technique	Advantages	Disadvantages
Discriminant function analysis (DFA)	1. DFA reduces subjective judgment as well as the level of expertise and experience needed for the determination of sex	1. DFA gave to slightly better results of classification (in the case of sex determination from upper femur by [9]
	2. DFA is techniques that are simple, quick, and accurate for gender determination which is always population specific	2. Regarding this research, the previous study obtained accuracy 87 %, which is lower than BPNN
Artificial neural network (ANN)	1. The neural network is a powerful classification technique and may improve the accuracy rate of gender determination models	1. The neural network have architecture is different from the architecture of microprocessors
	2. This applies to problems where the relationships may be quite dynamic or non-linear such as DFA	2. Requires high processing time for large neural networks
	3. The neural network using many variables gives the best overall results achieving the highest rate of correctly classified individuals	3. The neural network needs training to operate
	4. The neural network using variables correctly classified 92.1 % of male femurs and 94.7 % of female femurs [9]	
	5. Regarding this research, BPNN model provide result accuracy 99.030 % for training and 97.379 % for testing. Thus, indicate that BPNN give accuracy higher than DFA	

In the learning process of BPNN the experiment was repeated 10 times and the results are outlined in Tables 3 and 4.

Table 3 described that experimental result of 10 times experiment obtained best training and testing is on *lr* = 0.9 and *mc* = 0.5. The average accuracy obtained is 97.740 % training and 96.422 % testing. The performance of accuracy obtained by [6; 6; 2] structure of BPNN described in Plot Graph in Fig. 9

Whereas on model of structure [6; 12; 2] can be seen on Table 4.

Table 4 indicate that the performance of each *lr* and *mc* yeild different results in both training and testing. The experiment was repeated 10 times. The highest accuracy was found to be experimental *lr* = 0.5 and *mc* = 0.9, namely 99.030 and 97.379 % training and testing classification rates, respectively. The performance of accuracy obtained by [6; 12; 2] of BPNN described in Plot Graph in Fig. 10.

6 Conclusion

This paper presents a complete classification framework for gender determination in forensic anthropology. A proposed classification technique to determine gender using BPNN is presented. Based on the discussion of previous sections, it can be concluded that gender determination from measurements of the sacrum bones using BPNN is a legitimate alternative in cases of badly fragmented bones, when other classic measurements are not available (especially the bi-condylar width). ANN can improve the result in process gender determination with provide high accuracy result when compared to other classification techniques such as DFA. Sample data used is 91 sacrum bones obtained correct classification rate of 99.030 % for training and 97.379 % for testing was obtained using [6; 12; 2] structure of ANN as a way of discriminating between males and females. Structure of BPNN [6; 6; 2] provide average accuracy lower than [6; 12; 2] structure, namely 97.740 % training and 96.422 % testing. But if compared with DFA, it is still better than DFA which only provide accuracy 87 %. From the result of DFA and ANN can be concluded advantages and disadvantages of both techniques. The comparison advantages and disadvantages of DFA and ANN are shown in Table 5. Hence, the ANN achieved better results than other linear approach.

In the future work, in gender determination can use combination of classification techniques thus provide high accuracy and better than previous techniques. Although, in process data collection, can apply technique for feature extraction or feature selection to get best features will be processed in classification techniques.

Acknowledgment This research has been supported by Malaysian Ministry of Education under Fundamental Research Grant Scheme with UTM Vote Number RJ130000.7828.4F115.

References

1. Ahmed, A.A.: Estimation of stature from the upper limb measurements of Sudanese adults. Forensic Sci. Int. **228**, 178.e171–e178 (2013). doi:10.1016/j.forsciint.2013.03.008
2. Akhlaghi, M., Sheikhazadi, A., Ebrahimnia, A., Hedayati, M., Nazparvar, B., Saberi Anary, S. H.: The value of radius bone in prediction of sex and height in the Iranian population. J. Forensic Leg. Med. **19**(4), 219–222 (2012). doi:10.1016/j.jflm.2011.12.030
3. Akhlaghi, M., Sheikhazadi, A., Naghsh, A., Dorvashi, G.: Identification of sex in Iranian population using patella dimensions (Research Support, Non-U.S. Gov't). J. Forensic Leg. Med. **17**(3), 150–155 (2010). doi:10.1016/j.jflm.2009.11.005
4. Alba, E., Chicano, J.F.: Training neural networks with GA hybrid algorithms. Genetic and Evolutionary Computation—GECCO 2004, pp. 852–863. Springer, Heidelberg (2004)
5. Beale, M.H., Hagan, M.T., Demuth, H.B.: Neural Network Toolbox™.. User's Guide, Mar 2012
6. Cattaneo, C.: Forensic anthropology: developments of a classical discipline in the new millennium (review). Forensic Sci. Int. **165**(2–3), 185–193 (2007). doi:10.1016/j.forsciint.2006.05.018
7. Cunha, E., Pinheiro, J., Nuno Vieira, D.: Identification in forensic anthropology: its relation to genetics. Int. Congr. Ser. **1288**, 807–809 (2006). doi:10.1016/j.ics.2005.12.068
8. Dixit, S.G., Kakar, S., Agarwal, S., Choudhry, R.: Sexing of human hip bones of Indian origin by discriminant function analysis. J. Forensic Leg. Med. **14**(7), 429–435 (2007). doi:10.1016/j.jflm.2007.03.009
9. du Jardin, P., Ponsaille, J., Alunni-Perret, V., Quatrehomme, G.: A comparison between neural network and other metric methods to determine sex from the upper femur in a modern French population (comparative study). Forensic Sci. Int. **192**(1–3), 127 e121–126 (2009). doi:10.1016/j.forsciint.2009.07.014
10. Duric, M., Rakocevic, Z., Donic, D.: The reliability of sex determination of skeletons from forensic context in the Balkans. Forensic Sci. Int. **147**(2–3), 159–164 (2005). doi:10.1016/j.forsciint.2004.09.111
11. Editorial: Global forensic anthropology in the 21st century. Forensic Sci. Int. **117**, 1–6 (2001)
12. El Morsi, D.A., Al Hawary, A.A.: Sex determination by the length of metacarpals and phalanges: X-ray study on Egyptian population. J. Forensic Leg. Med. **20**(1), 6–13 (2013). doi:10.1016/j.jflm.2012.04.020
13. Eshak, G.A., Ahmed, H.M., Abdel Gawad, E.A.: Gender determination from hand bones length and volume using multidetector computed tomography: a study in Egyptian people. J. Forensic Leg. Med. **18**(6), 246–252 (2011). doi:10.1016/j.jflm.2011.04.005
14. Gomez-Valdes, J.A., Torres Ramirez, G., Baez Molgado, S., Herrera Sain-Leu, P., Castrejon Caballero, J.L., Sanchez-Mejorada, G.: Discriminant function analysis for sex assessment in pelvic girdle bones: sample from the contemporary Mexican population. J. Forensic Sci. **56**(2), 297–301 (2011). doi:10.1111/j.1556-4029.2010.01663.x
15. Gonzalez, P.N., Bernal, V., Perez, S.I.: Geometric morphometric approach to sex estimation of human pelvis (research support, Non-U.S. Gov't). Forensic Sci. Int. **189**(1–3), 68–74 (2009). doi:10.1016/j.forsciint.2009.04.012
16. Grisbaum, G.A., Ubelaker, D.H.: An Analysis of Forensic Anthropology Cases Submitted to the Smithsonian Institution by the Federal Bureau of Investigation from 1962 to 1994. Smithsonian Institution Press, Washington, DC (2001)
17. Guyomarc'h, P., Bruzek, J.: Accuracy and reliability in sex determination from skulls: a comparison of Fordisc(R) 3.0 and the discriminant function analysis (comparative study). Forensic Sci. Int. **208**(1–3), 180 e181–186 (2011). doi:10.1016/j.forsciint.2011.03.011

18. Hsiao, T.H., Tsai, S.M., Chou, S.T., Pan, J.Y., Tseng, Y.C., Chang, H.P., Chen, H.S.: Sex determination using discriminant function analysis in children and adolescents: a lateral cephalometric study (research support, Non-U.S. Gov't validation studies). Int J Legal Med. **124**(2), 155–160 (2010). doi:10.1007/s00414-009-0412-1
19. Iscan, M.Y., Olivera, H.E.S.: Forensic anthropology in Latin America. Forensic Sci. Int. **109**, 15–30 (2000)
20. Jianguo, Z., Gang, Q.: Application of BP neural network forecast model based on principal component analysis in railways freight forecas. In: Paper presented at the international conference on computer science and service system (2012)
21. Kanchan, T., Krishan, K.: Anthropometry of hand in sex determination of dismembered remains—a review of literature (review). J. Forensic Leg. Med. **18**(1), 14–17 (2011). doi:10.1016/j.jflm.2010.11.013
22. Kemkes-Grottenthaler, A.: Sex determination by discriminant analysis: an evaluation of the reliability of patella measurements. Forensic Sci. Int. **147**(2–3), 129–133 (2005). doi:10.1016/j.forsciint.2004.09.075
23. Kim, D.I., Kim, Y.S., Lee, U.Y., Han, S.H.: Sex determination from calcaneus in Korean using discriminant analysis. Forensic Sci. Int. **228**(1–3), 177 e171–177 (2013). doi:10.1016/j.forsciint.2013.03.012
24. Koçak, A., Özgür Aktas, E., Ertürk, S., Aktas, S., Yemisçigil, A.: Sex determination from the sternal end of the rib by osteometric analysis. Leg. Med. **5**(2), 100–104 (2003). doi:10.1016/s1344-6223(03)00045-2
25. Kottaimalai, R., Rajasekaran, M.P., Selvam, V., Kannapiran, B.: EEG signal classification using principal component analysis with neural network in brain computer interface applications. In: Paper presented at the international conference on emerging trends in computing, communication and nanotechnology (2013)
26. Kranioti, E., Paine, R.: Forensic anthropology in Europe: an assessment of current status and application. J. Anthropol. Sci. **89**, 71–92 (2011). doi:10.4436/jass.89002
27. Kranioti, E.F., Bastir, M., Sanchez-Meseguer, A., Rosas, A.: A geometric-morphometric study of the Cretan humerus for sex identification (research support, Non-U.S. Gov't). Forensic Sci. Int. **189**(1–3), 111 e111–118 (2009). doi:10.1016/j.forsciint.2009.04.013
28. Kranioti, E.F., Michalodimitrakis, M.: Sexual dimorphism of the humerus in contemporary Cretans–a population-specific study and a review of the literature* (review). J. Forensic Sci. **54**(5), 996–1000 (2009). doi:10.1111/j.1556-4029.2009.01103.x
29. Krishan, K., Sharma, A.: Estimation of stature from dimensions of hands and feet in a North Indian population. J. Forensic Leg. Med. **14**, 327–332 (2007). doi:10.1016/j.jcfm.2006.10.008
30. Kumaravel, G., Kumar, C.: A Novel Bats Echolocation System Based Back Propagation Algorithm for Feed Forward Neural Network. Springer, Heidelberg (2012)
31. Lee, T.-L.: Back-propagation neural network for the prediction of the short-term storm surge in Taichung harbor, Taiwan. Eng. Appl. Artif. Intell. **21**(1), 63–72 (2008). doi:10.1016/j.engappai.2007.03.002
32. Li, X.-f., Zhang, P.: A research on value of individual human capital of high-tech enterprises based on the bp neural network algorithm. In: The 19th International Conference on Industrial Engineering and Engineering Management, pp. 71–78. Springer, Berlin (2013)
33. Liang, L., Wu, D.: An application of pattern recognition on scoring Chinese corporations financial conditions based on backpropagation neural network. J. Comput. Oper. Res. **32**, 1115–1129 (2005)
34. Lin, C., Jiao, B., Liu, S., Guan, F., Chung, N.E., Han, S.H., Lee, U.Y.: Sex determination from the mandibular ramus flexure of Koreans by discrimination function analysis using three-dimensional mandible models (research support, Non-U.S. Gov't). Forensic Sci. Int. **236**, 191 e191–e196 (2014). doi:10.1016/j.forsciint.2013.12.015
35. Love, J. C., Hamilton, M.D.: Introduction to Forensic Anthropology, pp. 509–537 (2011). doi:10.1007/978-1-60761-872-0_19

36. Mahfouz, M., Badawi, A., Merkl, B., Fatah, E.E., Pritchard, E., Kesler, K., Jantz, L.: Patella sex determination by 3D statistical shape models and nonlinear classifiers. Forensic Sci. Int. **173**(2–3), 161–170 (2007). doi:10.1016/j.forsciint.2007.02.024
37. Mandal, S.R., Raju, D.H.: Ocean wave parameters estimation using Backpropagation Neural Networks. Mar. Struct. **18**, 301–318 (2005). doi:10.1016/j.marstruc.2005.09.002
38. Mastrangelo, P., De Luca, S., Aleman, I., Botella, M.C.: Sex assessment from the carpals bones: discriminant function analysis in a 20th century Spanish sample (historical article). Forensic Sci. Int. 206(1–3), 216 e211–210 (2011). doi:10.1016/j.forsciint.2011.01.007
39. Mastrangelo, P., De Luca, S., Sanchez-Mejorada, G.: Sex assessment from carpals bones: discriminant function analysis in a contemporary Mexican sample. Forensic Sci. Int. **209**(1–3), 196 e191–115 (2011). doi:10.1016/j.forsciint.2011.04.019
40. Mostafa, E.M., El-Elemi, A.H., El-Beblawy, M.A., Dawood, A.E.-W.A.: Adult sex identification using digital radiographs of the proximal epiphysis of the femur at Suez Canal University Hospital in Ismailia, Egypt. Egypt. J. Forensic Sci. **2**(3), 81–88 (2012). doi:10.1016/j.ejfs.2012.03.001
41. Mountrakis, C., Eliopoulos, C., Koilias, C.G., Manolis, S.K.: Sex determination using metatarsal osteometrics from the Athens collection (research support, Non-U.S. Gov't). Forensic Sci. Int. **200**(1–3), 178 e171–177 (2010). doi:10.1016/j.forsciint.2010.03.041
42. Nagalakshmi, S., Kamaraj, N.: On-line evaluation of loadability limit for pool model with TCSC using back propagation neural network. Int. J. Electr. Power Energy Syst. **47**, 52–60 (2013). doi:10.1016/j.ijepes.2012.10.051
43. Nagaoka, T., Hirata, K.: Reliability of metric determination of sex based on long-bone circumferences: perspectives from Yuigahama-minami, Japan (research support, Non-U.S. Gov't). Anat. Sci. Int. **84**(1–2), 7–16 (2009). doi:10.1007/s12565-008-0003-0
44. Nicholas, G., Hollowell, J.: World archaeological Congress research handbooks in archaeology. In: Blau, S., Ubelaker, D.H. (eds.) Handbook of Forensic Anthropology and Archaeology. Left Coast Press Inc, California (2009)
45. Ogawa, Y., Imaizumi, K., Miyasaka, S., Yoshino, M.: Discriminant functions for sex estimation of modern Japanese skulls. J. Forensic Leg. Med. **20**(4), 234–238 (2012). doi:10.1016/j.jflm.2012.09.023
46. Papaioannou, V.A., Kranioti, E.F., Joveneaux, P., Nathena, D., Michalodimitrakis, M.: Sexual dimorphism of the scapula and the clavicle in a contemporary Greek population: applications in forensic identification. Forensic Sci. Int. **217**(1–3), 231 e231–237 (2012). doi:10.1016/j.forsciint.2011.11.010
47. Raghavendra Babu, Y.P., Kanchan, T., Attiku, Y., Dixit, P.N., Kotian, M.S.: Sex estimation from foramen magnum dimensions in an Indian population. J. Forensic Leg. Med. **19**(3), 162–167 (2012). doi:10.1016/j.jflm.2011.12.019
48. Rajasekaran, S., Vijayalakshmi, G.A.: Neural Networks, Fuzzy Logic, Genetic Algorithms, Synthesis and Applications. Prentice-Hall of India, New Delhi (2007)
49. Ramsthaler, F., Kreutz, K., Verhoff, M.A.: Accuracy of metric sex analysis of skeletal remains using Fordisc based on a recent skull collection (comparative study). Int. J. Legal Med. **121**(6), 477–482 (2007). doi:10.1007/s00414-007-0199-x
50. Slaus, M., Bedic, Z., Strinovic, D., Petrovecki, V.: Sex determination by discriminant function analysis of the tibia for contemporary Croats (research support, Non-U.S. Gov't). Forensic Sci. Int. **226**(1–3), 302 e301–304 (2013). doi:10.1016/j.forsciint.2013.01.025
51. Steadman, D., Andersen, S.A.: Personal identification: theory and applications the case study approach, pp. 12–15 (2008)
52. Tan, M., He, G., Nie, F., Zhang, L., Hu, L.: Optimization of ultrafiltration membrane fabrication using backpropagation neural network and genetic algorithm. J. Taiwan Inst. Chem. Eng. (2013). doi:10.1016/j.jtice.2013.04.004

53. Thompson, T.J.U.: Recent advances in the study of burned bone and their implications for forensic anthropology. Forensic Sci. Int. **146**, S203–S205 (2004). doi:10.1016/j.forsciint.2004.09.063
54. Ubelaker, D.H.: Chap. 1: Forensic anthropology. Humana Press Inc, Totowa (1989)
55. Uthman, A.T., Al-Rawi, N.H., Al-Naaimi, A.S., Tawfeeq, A.S., Suhail, E.H.: Evaluation of frontal sinus and skull measurements using spiral CT scanning: an aid in unknown person identification. Forensic Sci. Int. **197**(1–3), 124 e121–127 (2010). doi:10.1016/j.forsciint.2009.12.064
56. Zech, W.D., Hatch, G., Siegenthaler, L., Thali, M.J., Losch, S.: Sex determination from os sacrum by postmortem CT. Forensic Sci. Int. **221**(1–3), 39–43 (2012). doi:10.1016/j.forsciint.2012.03.022

Neural Network Approach to Fault Location for High Speed Protective Relaying of Transmission Lines

Moez Ben Hessine, Houda Jouini and Souad Chebbi

Abstract Fault location and distance protection in transmission lines are essential smart grid technologies ensuring reliability of the power system and achieve the continuity of service. The objective of this chapter is to presents an accurate algorithm for estimating fault location in Extra High Voltage (EHV) transmission lines using Artificial Neural Networks (ANNs) for high speed protection. The development of this algorithm is based on disturbed transmission line models. The proposed fault protection (fault detection/classification and location) uses only the three phase currents signals at the one end of the line. The proposed technique uses five ANNs networks and consists of two steps, including fault detection/classification and fault location. For fault detection/classification, one ANN network is used in order to identify the fault type; the fault detection/classification procedure uses the fundamental components of pre-fault and post-fault sequence samples of three phase currents and zero sequence current. For fault location, four ANNs networks are used in order to estimate the exact fault location in transmission line. Magnitudes of pre-fault and post-fault of three phase currents are used. The ANNs are trained with data under a wide variety of fault conditions and used for the fault classification and fault location on the transmission line. The proposed fault detection/classification and location approaches are tested under different fault conditions such as different fault locations, different fault resistances and different fault inception angles via digital simulation using MATLAB software in order to verify the performances of the proposed methods. The ANN-based fault classifier and locator gives high accuracy for all tests under different fault conditions.

M.B. Hessine (✉) · H. Jouini · S. Chebbi
Laboratory of Technologies of Information and Communication and Electrical Engineering (LaTICE), High Engineering School of Tunis, ENSIT, University of Tunis, 5 Avenue Taha Hussein-Montfleury, BP 56 Bab Mnara, 1008 Tunis, Tunisia
e-mail: benhessinemoez@yahoo.fr

H. Jouini
e-mail: houda.jouini@gmail.com

S. Chebbi
e-mail: souadchebbi@yahoo.com

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_12

The simulations results show that the proposed scheme based on ANNs can be used for on-line fault protection in transmission line.

Keywords Transmission line protection · Extra high voltage · Fault detection/classification · Fault location algorithm · Artificial neural network · Modular artificial neural networks

1 Introduction

Electrical transport network is an important mean of delivering to consumers, the powers produced by the electrical energy production stations. A disturbance occurring on transport elements evacuating this production can have severe consequences. Sometimes a national or regional blackout can occur due to a fault on one of the components of the transport network. Among recent blackout and the affected population we cite as an example the incident of 30 June 2002 in Tunisia: 10 million people, the incident of 3 February 2003 in Algeria: 25 million people, the incident of 14 August 2003 in united states of America and Canada: 50 million people, the incident of 23 September 2003 in Sweden and Denmark: 5 million people, the incident of 28 September 2003 in Italy: 57 million people. Thus we understand the particular attention with which establishes the protections of Transport networks and the means of locating faults. Fast fault location is one of the major concerns of electricity companies to ensure the continuity of service and to reduce the duration of interruptions.

Transmission line is one of the main important components of the electric power system and its protection is necessary for ensuring reliability of the power system, continuity of service, stability of system and economic operation of power system. Transmission line is exposed to the environment and the possibility of experiencing faults on the transmission line is generally higher than that on other components. On a transmission lines the protective relaying system is integrated in order to detect the faults and to isolate the faulted part from the rest of the power system.

The electromechanical and the electronic relays were first used for the protection of transmission lines. These relays informed only about the zone of the line where the fault occurs of which the location requires a visual inspection, requiring the intervention of qualified personnel for a long period.

In recent years the digital protection relays began to replace the electromechanical and electronic protection relays. These relays have the advantage of integrating a fault locator which remains an expensive solution if it is generalized to the transmission lines.

Transmission line protection consists of three major tasks fault detection, classification and fault location. A fast detection of a fault occurred in transmission line allows fast isolation of the faulty line from service and thus offering protection from adverse effects of the fault. Once we know that a fault has occurred on an electrical transmission line, the next step is to identify the fault type into the different categories based on the phases that are faulted. Then, the third step is designed to estimate the distance of the fault in the transmission line.

Accurate fault location is highly required by operators and utility staffs to expedite service restoration, fast reparation and restoration of the faulty line in order to improve reliability and the service restoration reduce outage time, operating costs and customer complains. Fault location is still the subject of rapid further developments. Research efforts are focused on developing efficient fault location algorithms intended for application to more and more complex networks.

Fault location is a process enables to locate of the fault in transmission line with the highest accuracy possible. Fault locators algorithms present generally the supplementary protection equipment, which apply the fault location algorithms for estimate the exact fault location a distance to the fault. When the transmission line consisting of more than one section (multi-terminal line), initially a faulted section has to be identified and then a fault on this section has to be located.

Fault location algorithm can be implemented in:

- Microprocessor-based relays;
- digital fault recorders (DFRs);
- Stand-alone fault locators;
- Post-fault analysis programs.

Transmission fault location techniques can be classified into three main categories: techniques based on traveling waves [5, 22, 24, 27, 54, 55, 59, 67], techniques utilizing the higher frequency components (harmonics) of currents and voltages [19, 43, 65] and techniques utilizing the fundamental frequency voltages and currents measured at the terminals of a line [25, 31, 51]. The techniques in these categories can be further classified into two subcategories: these techniques which use measurements from one terminal of the transmission line and techniques which use measurements taken from both terminals line [21, 41] are generally more accurate than the ones using data only from one terminal. However, in many transmission lines, a communication channel between the line terminals is not available, thus make it necessary to use data from the one terminal line only.

Fault location algorithms using one terminal line data (voltages and currents) need to make some simplifying assumptions for the fast calculation of the exact fault location. However, the fault detection/classification and location techniques using one terminal data could be more attractive for researchers. Various techniques of fault detection/classification and location have been developed in the literature. Transmission lines protection is based on the estimation of the fundamental power frequency components. Barros and Drake [17], Girgis and Brown [30] used kalman filter, discrete Fourier transformation [23], walsh function [33], etc. for estimate the phasor quantities. Nevertheless, these techniques didn't have the ability to adapt dynamically to the system operating conditions, and require a long computation time.

There is a need to develop algorithms that have the ability to adapt dynamically to the system operating conditions such as changes in the system configuration, source impedances and faults conditions (fault resistance, fault inception angle, fault position).

In this context, various fault detection, classifications and location approaches for transmission lines have been developed. These approaches are based on intelligent artificial tools such as Fuzzy Logic [20, 28, 46, 50, 67], Neuro-fuzzy [37, 38, 54, 64], Fuzzy Logic-Wavelet based systems [47, 69] and Artificial Neural Networks ([4, 6, 31, 35 41, 46, 49, 62]; Yilmaz et al. 2012).

The goal of this chapter is to develop and integrate a new and accurate fault detection/classification and location based on ANN for high speed protection relays in EHV transmission lines compared to conventional methods. A single end fault detection/classification and location algorithms are proposed for on-line application using artificial neural networks (ANNs) for all the ten types of faults in transmission lines. Throughout the study a 400 kV transmission line of 100 km length has been chosen as a representative system. Pre-fault and post-fault samples of three phase currents and zero sequence currents are used to train the ANNs in order to classify and accurately locate the faults on transmission line.

The remainder of the chapter is organized as follows: Sect. 2 describes the reviews for existing fault detection/classification and location in transmission lines. The power system under study used for training and testing the proposed ANNs based on fault detection/classification and location is given in Sect. 3. Description for the artificial neural networks and the learning algorithm used in this work is presented in Sect. 4. Section 5 describes the proposed algorithms for fault detection/classification and location using single ANN approach and modular ANNs approach respectively. Tests performances for the proposed fault protection scheme are given in Sect. 6. Finally, Sect. 7 presents a comparative study between the proposed scheme and the related works.

2 Reviews for Existing Fault Location

Since the 1960s, several significant researches for fault location and protection of energy transmission line subjects had developed. It was motivated by the fact that more than half of the faults occur on the airlines. Thus (Rockefeller 1969) proposed a protection scheme for the electrical network based on digital relays. In recent years, new systems of digital relays were developed [58, 59] and field-tested by electric companies. Many reasons favoring the fast development of digital relays are presented as follows:

- Reduced price of digital equipments;
- High reliability made possible by monitoring the power grid and self -diagnostic system relay;
- Best performance, due to practicality, to implement the various functions of relays and desired to form complex features operational;

- Best coordination of system operation, control and protection through effective organization of data transmission channels.

Therefore microprocessors are used to make an algorithm for fault location of the distance relay in order to calculate the distance of a fault on transmission line. With the possibilities offered by the microprocessors, the corresponding complex calculations can be performed quickly and with good accuracy. Several fault location algorithms have been proposed and applied to determine the point of faults on transmission lines. The algorithms developed in the literature can be classified into two main categories: algorithms based on the fundamental power frequency components, and algorithms based on the high frequency component of the fault signals. Each of these two groups can be further divided into two subcategories: to use measurements from one terminal of the transmission line or to use measurements taken from both terminals line.

In the rest of this section, some fault location algorithms in transmission lines are reviewed and the object of this chapter is described

2.1 Double-End Measurement

2.1.1 Synchronized Measurement

The fault location algorithms could be more accurate if more information about the transmission line were available. Thus, if the communication channels are available in transmission line, the techniques use the measurements at the two ends are used for locating the exact fault position. These techniques are more precise than the distance relaying protection algorithms which are affected by the insufficient transmission line modeling and the parameter uncertainty due to the aging of lines.

In the 80s the techniques use the synchronization measurement technology appeared as a promising prospect in the realization of real time protection in transmission lines. With global positioning system (GPS), digital measurement at different line terminals can be performed synchronously [1, 8, 18, 55]. Phasor Measurements Units (PMU) are the most frequently used synchronized measurement devices for system protection, whose measurements are synchronized relative to a GPS clock, for it the fault locators algorithm used the PMU are more accurate than the method based on unsynchronized measurements [40, 43, 45, 60, 63].

Moreover, these techniques require the presence of a GPS where measurements are synchronized compared to a GPS clock. Nevertheless, the synchronized measurement technology presents many drawbacks as the high cost and the presence of a communication channel between the line terminals which is not available in the majority of lines. Therefore, the fault diagnosis techniques using one terminal data could be more attractive for researchers (Fig. 1).

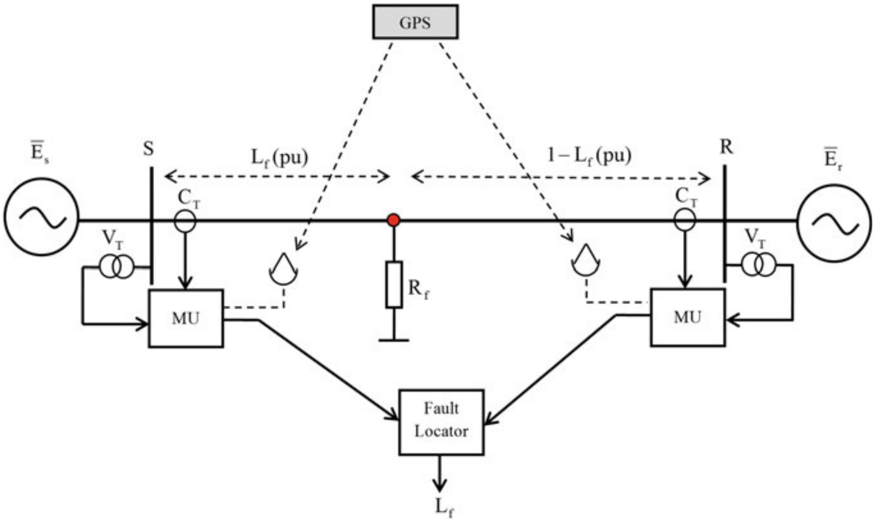


Fig. 1 Schematic diagram of two-end synchronized fault location using GPS synchronization

2.1.2 Unsynchronized Measurement

The fault location algorithms based on unsynchronized measurements are cheaper, because in this approach there is no need to use the GPS. In the Phasor Measurements Units (PMU) the digital measurements are realized. In this case, it is Considered That the Analog-Digital (A/D) converters installed in the PMU are not supplied with the GPS signals. Accordingly, the fault location techniques used the unsynchronized measurement are not affected by errors due to various sampling rates established by the various recording devices and transducers. Thus, the inputs signals (currents and voltage) for fault locator do not have common time reference; i.e. the measurements are unsynchronized.

Various fault location algorithm used this technique have been developed and presented by Izykowski et al. [36], in this works the percentage errors of fault location are negligible if phasor and transmission line parameters are accurate. Although the use of GPS, Phasor Measurement Units (PMU), digital communication technologies, high precision signal transducers have facilitated accurate protection of power system over a wide area, they are subjected to software insecurity and communications latency (Fig. 2).

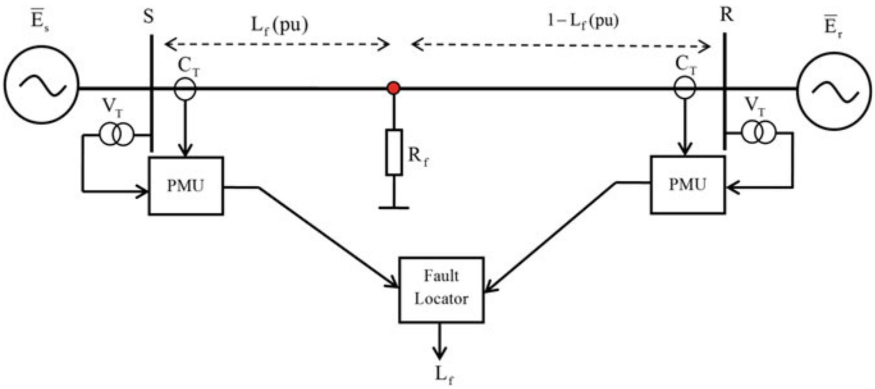


Fig. 2 Schematic diagram of two-end unsynchronized fault location

2.2 Single-End Measurement

Fault location techniques use measurements taken from both terminals line, is expensive since it requires the synchronization equipment and telecommunication. This approach is not generalized and is only used on high-voltage lines and direct current lines. This technique utilizes measurements of three phase current and voltage from one terminal line (Fig. 3). This technique has a major advantage such as no communication means are needed and simplicity for implementation. Fault location based on data from only one end is the most commonly used.

Single-end algorithms-based fault location estimate a distance to fault with the use of fundamental components of three phase voltages and currents acquired at a particular end of the transmission line by the protection relay through C_T and V_T.

Different fault location techniques using one terminal line data are developed in the literature, these techniques based on the estimation of the fundamental power frequency components using kalman filter, discrete Fourier transformation, walsh function, etc. Nevertheless, these techniques didn't have the ability to adapt dynamically to the system operating conditions, and require a long computation time.

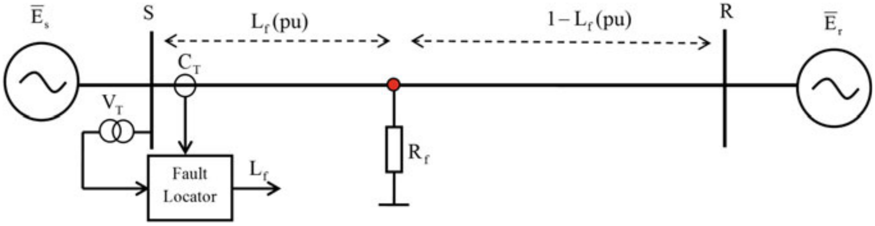


Fig. 3 Schematic diagram of one-end fault location

There is a need to develop algorithms that have the ability to adapt dynamically to the system operating conditions such as changes in the system configuration, source impedances and faults conditions (fault resistance, fault inception angle, fault position).

Today, Fuzzy Logic and Artificial Neural Networks represent an area of intensive research in different applications of system identification, control systems, biomedical application, signal processing and fault diagnosis. Fuzzy logic, which is a mathematical tool based on fuzzy sets theory [10, 11, 14, 51] has rapidly become one of the most successful technologies for developing sophisticated control systems today.

Recently, the combination of fuzzy logic and neural networks has been studied for applying in real applications [9, 12, 13, 15, 37, 38]. With this combination, neuro-fuzzy systems use the advantages of both approaches, namely, fuzzy logic and neural networks. These advantages make neuro-fuzzy systems a powerful tool which can be applied in different disciplines as system identification, control systems, signal process, load forecasting in power systems, and protection system.

The principal objective of this chapter is to explore the capacity of artificial neural network for identify the fault types and to estimate the faults in the transmission lines for high speed protective relaying system.

The ANN based on protection relays have been developed as an alternative to conventional methods, since they present very promising results with regard to precision and operating time. Different techniques have been published describing the applications of neural networks in fault classification and location.

Application of ANN to fault location in transmission line is proposed by Tahar [63]. This approach consists of two parts. In the first part the fault detection is determined, and in the second part the fault position is calculated. But the fault type and the response time for this approach is not indicate. RBF (radial basis function) neural networks based fault classification and location algorithms are proposed by Joorabian et al. [41]. The maximum error of the fault location algorithm is 0.5 %. Nevertheless, the fault detection and the response time for this approach are not indicated. Banu and Suja [16] developed a new fault location scheme for a transmission line is proposed. The scheme uses a single ANN based on the Levenberg Marquardt optimization technique. But the fault detection and the fault type are not indicated; also the error of the algorithm is kept below 0.65 %. Wavelet and neural network fault classification and location are developed by Aritra et al. [7]. The reported method is suitable for classifies all ten types of faults as well as estimates the location of faults simultaneously with maximum fault-location error is 3.25 % and a response time is not indicated. Gaganpreet et al. [29], Hassan and Zuyi [33] proposed a neural network approaches for fault detection and fault location in transmission lines. Nevertheless, these approaches detect only the faults appeared in the first zone of the line, namely, 80 % of the transmission line length. A neural network approach for fault classification is presented by Arita et al. [7].This approach can be classified the faults line-to-ground (L-G), line-to-line (L-L), and three-line (L-L-L) faults for a particular distance of fault location, although an line-to-line-to-ground (L-L-G) fault is not considered. An alternative approach to fault

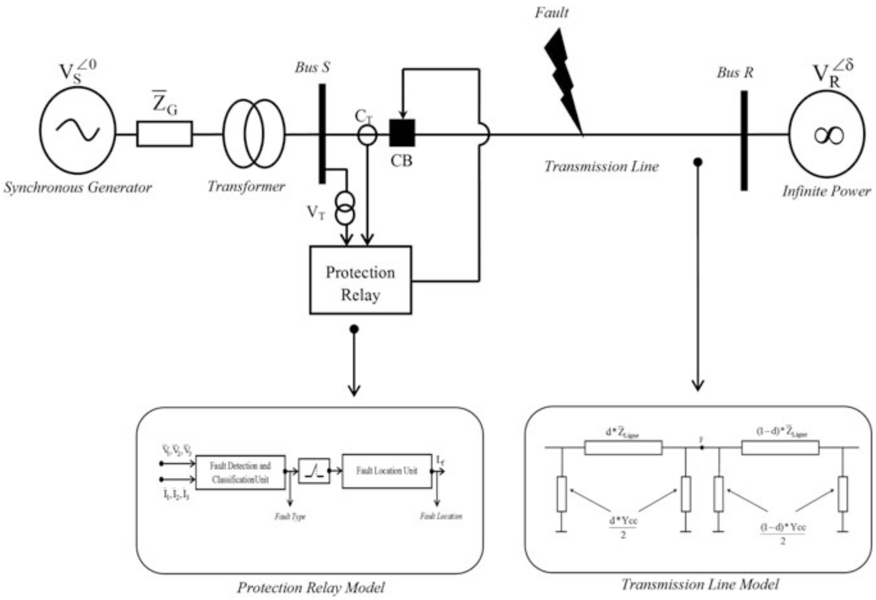


Fig. 4 Power system under study

classification and location are presented by Yilmaz [66]. The error of the algorithm is kept below 3 %. In Ref. [4] fault distance and direction estimation based on ANN for protection of doubly fed transmission lines is proposed, but the fault type is not indicated. The operating time of this approach is 1.5 cycles. Jiang et al. uses a fault-location module for fault diagnosis, which incorporates a two-stage adaptive structure neural network, the fault detection, classification and location algorithms are presented with averaged fault location error equal to 0.5 %.The results clearly show that this approach leads to a reliable location for all types of faults with times equal to 1.28 cycle after fault occurrence.

3 Power System under Consideration

The performance validities of the developed fault detection/classification and location algorithms are evaluated by the simulation model illustrated by Fig. 4. The model is composed of an electrical energy production station connected to an infinite power network via 100 km transmission line. The transmission line is represented by disturbed parameters model. Transmission line parameters are given in Table 1.

The model of the synchronous generator may be found in [2, 3, 50–52].

Table 1 Transmission line sequence impedances

Component	Parameter	
Transmission line	Length (km)	100
	Voltage (kV)	400
	Positive sequence impedance (Ω/km)	(0.0275 + j0.422)
	Zero sequence impedance (Ω/km)	(0.275 + j1.169)
	Positive sequence capacitance (nF/km)	9.483
	Zero sequence capacitance (nF/km)	6.711 nF/km

4 Configuration of Proposed Fault Location

In this section the architecture of the proposed fault location algorithm is developed and presented.

The proposed technique consists of two modules: fault detection/classification and fault location. This strategy can analyze faults occurring between two busses. Specifically, the first step of the proposed scheme is to detect and classify the fault type in the transmission line in real time. If no fault is detected, the remaining portion of the modules will not be activated. On the other hand, if the fault detection/classification module captures the feature of a fault, it will activate fault location module.

The inputs of the protective relay are principally three phase voltages and currents at the recorded location of the protective relay. These signals (currents and voltages) at the end of line (relay site in the transmission line) will be acquired by the relay via current transformer C_T and voltage transformer V_T.

Pre-processing of three phase currents and voltages signal measured at one end only of the transmission line can significantly reduce the size and the training time of the Neural Network.

The ANN-based fault protection (ANN-based fault detector/classifier and ANN-based fault locator) uses the fundamental magnitudes of pre-fault and post-fault samples of three phase currents.

Most digital protection relays use the fundamental frequency of a signal sampled, for that the Fast Fourier Transform is the most common method to estimate the magnitude and the phase of the fundamental frequency for each signal (current and voltage). The schematic diagram for the proposed fault classifier and locator depicted by Fig. 5.

- Anti-Aliasing Filter: The anti-aliasing filter removes the unwanted frequencies from a sampled waveform. A simple second order low-pass Butterworth filter with cut-off frequency of 400 Hz is integrated.
- Sampling Rate: The three phase currents and voltages signals are sampled at a sampling frequency of 1 kHz; this sampling rate is compatible with the sampling rates currently used by the digital relays

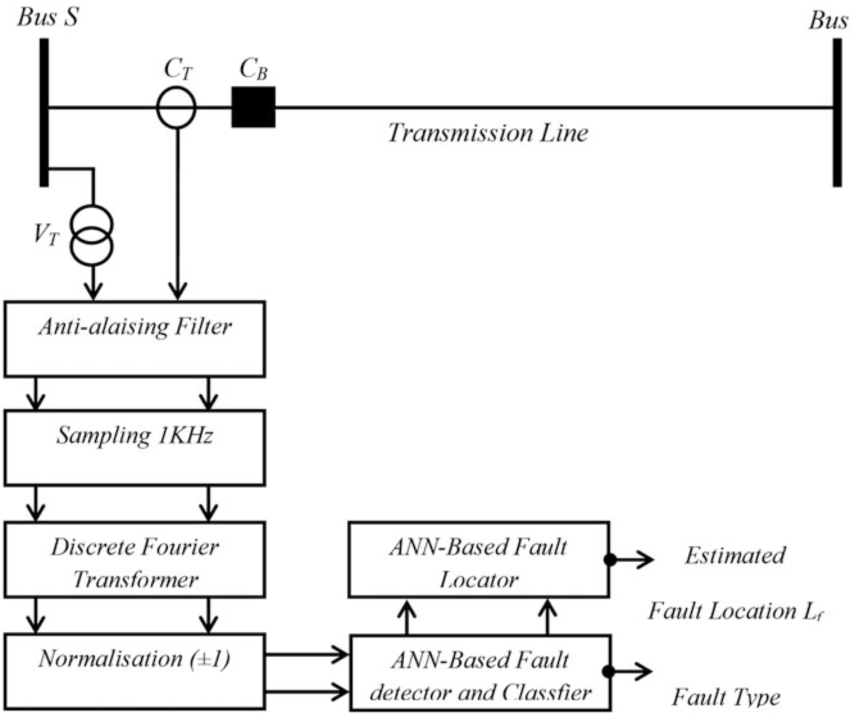


Fig. 5 Complete Fault Location algorithm

- Discrete Fourier Transformer: One full cycle Discrete Fourier Transform (DFT) is used to calculate the magnitudes of fundamental components of three phase currents and voltages after the fault appearance
- Normalization (±1): The input signals samples have to be normalized in order to reach the input level (±1).

This work presents a new scheme for fault protection in transmission line. The proposed scheme employs Artificial Neural Networks for fault detection/classification and fault location on transmission lines. In this respect, the main goal of the next section is to develop the principal function of the ANN used in this work.

4.1 Feed- Forward Neural Networks

4.1.1 Presentation

In this paper, a multi-layer neural networks (FFNNs) was used and trained with a supervised learning algorithm called back-propagation. The multi-layer neural

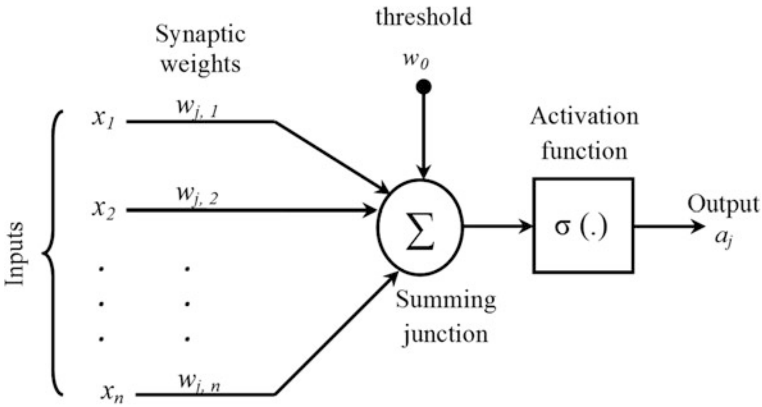


Fig. 6 Neural network model

network consists of three layers: an input layer, an output layer, and one or more hidden layer. Each layer consists of a predefined number of neurons. We recall that the neural network is a collection of cells of neurons interconnected by synaptic weights and biases. The inputs are connected to the first hidden layer. Each hidden layer is connected to the next hidden layer, and the last hidden layer is connected to the output layer (Fig. 6).

A neuron mathematical model has a much simpler structure comparing with a biological neuron [37]. However, a neuron j can be described mathematically with the following equation.

$$a_j = \sigma(w_0 + \sum_{i=1}^{P} w_{ij} x_i) \tag{1}$$

Where:

σ	represent the transfer function (activation function) of neuron j
$\{x_i\}$	i = 1…n: represents the inputs signals of neuron j
$\{w_{ij}\}$	represents the weight coefficients of the connection between inputs and neuron j
w_0	is the bias of neuron j

An feed-forward NN consist of input, hidden and output layers is considered with P, Q and R neurons for each layer respectively. In this structure, $[x] = [x_1, x_2, \ldots, x_i, \ldots, x_p]$ representing the inputs are applied to the first layer, then, the inputs vector are transferred to the hidden layer using the connection weight between the input and the hidden layer. The output vector $[a] = [a_1, a_2, \ldots, a_j, \ldots, a_Q]$ of the hidden layer is then obtained. The neuron output a_j is determined as follows:

$$a_j = \sigma_{hidden}\left(w_0 + \sum_{i=1}^{P} w_{ij}^{hidden} x_i\right) \quad (2)$$

Where:

w_{ij}^{hidden}	represent the connection weight between the neuron j in the hidden layer and the ième neuron of the input layer
w_0	represent the bias of neuron j
σ_{hidden}	represent the activation function of the hidden layer

The values of the vector [a] of the hidden layer are transferred to the output layer using the connection weight between the hidden layers and the output layer. However, the output vector [b] = (b_1, b_2, b_k, …, b_R) of the output layer is determined. The output a_k of the neuron K (on the output layer) is obtained as follows:

$$a_k = \sigma_{out}\left(w_0 + \sum_{j=1}^{R} w_{jk}^{out} x_i\right) \quad (3)$$

w_{jk}^{out}	represent the connection weight between the neuron K in the output layer and of the jth neuron of the hidden layer
σ_{out}	is the activation function of the output layer

The error in the output layer between the output a_k and its desired value $a_{k\text{-desired}}$ ($a_k - a_{k\text{-desired}}$) is minimized by the mean square error at the output layer, defined as follows:

$$\text{Error} = \frac{1}{2}\sum_{k=1}^{Q} (a_{k-\text{desired}} - a_k)^2 \quad (4)$$

The training data set of an ANN should contain the necessary information to generalize the problem. In this work, different combinations of various fault conditions were considered and training patterns were generated by simulating different fault situation on the power system study. Fault conditions such as fault resistance, fault location, and fault inception angle were changed to obtain training patterns covering a wide range of different power system conditions.

4.1.2 Design Process

The design process of the Artificial Neural Networks (ANNs) used on fault detector/classifier $F_{Classifier}$ and fault locator $F_{Locator}$ in transmission line detailed by the following steps:

Step 1. Preparation a data base from all simulation.
Step 2. Assemble and pre-process the training data for ANNs.
Step 3. Training Process.
Step 4. Test of performances.
Step 5. Select the best ANN gives the best performance and stored the trained network.
Step 6. Application.

5 Proposed Fault Classifier and Locator Using ANN's Tool

Our contribution consists in presenting a new approach for fault detection/classification and location in Extra High Voltage transmission line EHV. The suggested methodologies uses the single ANN approach for fault detection/classification in order to identify the presence or not of a fault and indicate the fault type, in the second part a new modular ANNs approach is used for estimate the exact fault location in EHV transmission line. In this respect, the objective of this section is to conceive, develop, test and implement a complete fault protection strategy.

Firstly, the proposed fault classifier in this work based on the integration of one Artificial Neural Network (ANN) used to classify the different fault type in transmission line. The ANN takes in consideration the pre-fault and post-fault samples of the fundamental components of the three phase currents and the zero sequence currents. These signals (three phase currents and zero sequence currents) are sampled at a frequency of 1 kHz. The inputs data for ANN-based-fault classifier are four pre-fault and four post-fault for each phase currents and four sample for the zero sequence current.

The ($^{I_i(k)}/_{I_{i-PF}(k)}$, with $i = \{R, S, T\}$) are the per-unit values calculated by the division of the samples currents in fault time $I_i(k)$ (post-fault) to the pre-fault samples current $I_{i-PF}(k)$ in related phase. Consequently, the selected input numbers for the fault classification algorithm is equal to 16: four current samples for each phase (R, S and T) and four samples for zero sequence current. The input vector as shown in the following equation:

$$X_{FC1} = \left[\frac{I_R(k)}{I_{R-PF}(k)}, \ldots, \frac{I_R(k+3)}{I_{R-PF}(k-3)}; \frac{I_S(k)}{I_{S-PF}(k)}, \ldots, \frac{I_S(k+3)}{I_{S-PF}(k-3)}; \frac{I_T(k)}{I_{T-PF}(k)}, \ldots, \frac{I_T(k+3)}{I_{T-PF}(k-3)}; I_0(k), \ldots, I_0(k+3)\right] \tag{5}$$

The outputs have been termed as R, S, T and G, which represent the three phases and ground. Any one of the outputs R, S, G approaching 1 indicates a fault in that phase, and if G is taken 1 indicates a fault related to ground (Table 2).

Once the fault is detected and classified, the relevant ANNs for fault location are activated. The inputs for these networks are the magnitude of three phase current and the output is the normalized distance of the fault point from the sending end of the transmission line.

Table 2 Logic output of the ANN fault classifier

Fault type	*R*	*S*	*T*	*G*
No-fault	0	0	0	0
'R'-phase earth fault	1	0	0	1
'R'-'S'-phase fault	1	1	0	0
'R'-'S'-phase earth fault	1	1	0	1
Three phase fault	1	1	1	0

The proposed neural locator for our study case is designed to indicate the fault location in transmission lines. The fault locator is activated when a fault is detected and classified by the fault detector and fault classifier respectively. The exact location of such a fault is given by identifying directly the power system state starting from the instantaneous current and voltage data.

The overall algorithm of proposed ANN-based fault locator is detailed in Fig. 7. The single ANN approach based fault locator present many disadvantage such as the wide training sets, long training time, complexity architecture which affect the accuracy of the fault location algorithms [4, 7, 63]. Thus, it was decided to develop a new algorithm based on modular ANN approach present many advantage (simplicity, less training sets, less training time and more accuracy) compared on single ANN approach. The proposed fault locator algorithm consists of four independent ANNs, one for each fault type ($\mathrm{ANN_{LG}}$, $\mathrm{ANN_{LLG}}$, $\mathrm{ANN_{LL}}$ and $\mathrm{ANN_{LLL}}$). In this case each fault type trained by one neural network. Finally, the outputs of the ANNs are used to realise the fault location task.

The principal factor in the determination of the adequate size and architecture for Artificial Neural Network is the input and output numbers which it must have. However, the sufficient input data to characterize the problem must be assured. The recorded signals at one terminal line are used for the fault location task

Various works [4, 7, 63, 66] treated at the same time the magnitudes of the fundamental components (50 Hz) of three phase currents and voltages measured where the protection relay is installed in order to estimate the exact fault location, which leads to establishment of a complex ANNs architecture dedicated to this task and a long training time and slow learning capability.

As a perspective to reduce the ANN sizes used for fault location and to allot additive performances to this task, we are based on the fact that only magnitudes of the fundamental components of three phase currents $\mathrm{I_R}$, $\mathrm{I_S}$ and $\mathrm{I_T}$ used. This makes it possible to solve the quoted problems with reduced ANN architectures with high accuracy and a fast training time. For that, in our study case, we thought of integrate a neural fault locator which treats only the magnitude of three phase currents.

The inputs of ANNs are the magnitudes of the fundamental components (50 Hz) of three phase currents. The output of modular ANN-based fault locator is a real number indicates the fault distance location in km.

Before the currents signals penetrate in the neural network, a scaling technique will have a great importance in order to reduce the computing execution time.

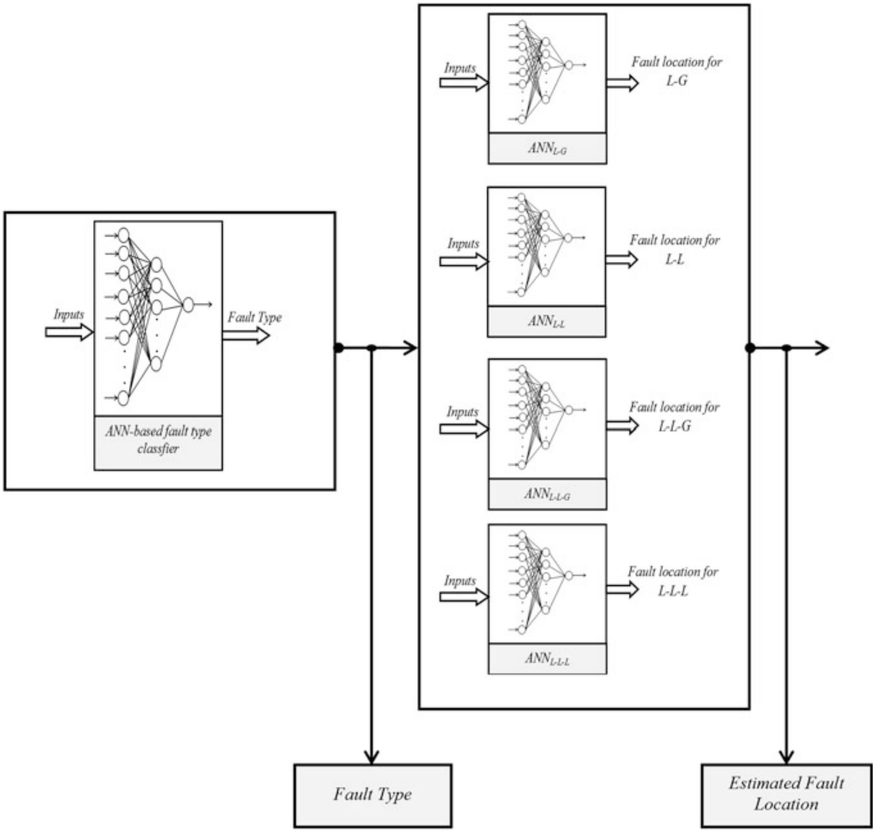


Fig. 7 The proposed ANN structure for fault location

For this purpose, we thought to adopt a scaling technique expressed by the dividing the magnitudes of the fundamentals components of three phase currents $I_i(k)$ during the fault time (post-fault) to the pre-fault fundamental components in related phase $(I_{i-PF}(k))$ with $i = \{R, S, T\}$. Thus we indicated $X_{F-locator}$ the inputs vector taken by the fault neural locator and by $Y_{F-locator}$ the output of the proposed fault locator.

$$\begin{aligned} X_{F-locator} &= \left[\frac{I_R(k)}{I_{R-PF}(k)}, \frac{I_S(k)}{I_{S-PF}(k)}, \frac{I_T(k)}{I_{T-PF}(k)} \right] \\ Y_{F-locator} &= L_F \end{aligned} \tag{6}$$

It is, extremely important to subject ANNs a good training and to test them correctly, we used the error back-propagation training algorithm (BPNN). ANNs undergo training with various patterns corresponding to different types and different fault conditions such as various fault location L_f, various fault resistances R_f and various fault inception angle FIA. At the training time, different structures (many

Table 3 Parameter settings for generating training and test patterns

Parameter	Training	Testing
Fault location L_f (km)	01, 10, 20, 30, 40, 50, 60, 70, 80, and 90	08, 11, 12, 24, 32, 48, 53,62, 77, 86, 88 and 95
Fault inseption angle FIA (°)	0, 180 and 360	5,10, 25, 40, 125, 135, 145, 215 and 275
Fault resistance R_f (Ω)	0, 75 and 100	1, 2.6, 12, 13, 22, 26, 35, 44, 65, 75, 95

neurons in the hidden layer) with various parameters such as the training rate and the transfer functions are evaluated to determine the optimal structure of the network making it possible to produce a good training and to have the best results. In order to obtain a wide training process for an effective performance of the suggested fault locator, each of the ten fault types was simulated at various location of the considered transmission line; various fault conditions such as fault resistance and fault inception angle are also modified to include several fault scenarios possible in real time.

Table 3 contains the parameter values used to generate data training sets and test patterns for the ANNs of the fault classifier and locator.

Each fault type at various fault conditions such as different fault locations L_f, different fault resistances R_f and different fault inception angle FIA have been simulated as shown below in Table 1. The total number of fault simulated are 10 (fault locations) × 3 (fault resistance) × 10 (fault type) × 3 (fault inception angles) = 900 for fault classification and for fault location.

The fault classifier and locator structure corresponds to the layer numbers and the neurons numbers in the hidden layers, input and output. After a test series and the ANN architecture modifications, the best performance is obtained by using a neurons network with three layers, (Fig. 3). The neurons number in the input layer corresponds to the ANN input variable numbers. The neurons number in the hidden layer was given after a test series.

The ANN fault classifier consists of 16 input neurons (four samples of each signal): I_R, I_S, I_T, I_0), 30 neurons in the hidden layer selected after a series of trials and four output neurons dedicated to indicate the fault type in the transmission line. Consequently, the ANN structure of the adopted fault classifier is (16-30-4).

Also architectures of ANN based fault distance locator are shown in Table 4. The number is epochs required for training varies from 188 to 300 to reduce the mean square error below 3.88e−5 (Figs. 8 and 9).

The final determination of the neural network requires the relevant transfer functions in the hidden and output layers. After analyzing the various possible combinations of transfer functions usually used such as “logsig”, “tansig” and “purelin” functions. The hyperbolic tangent sigmoid function “tansig” has been used in the hidden layer and the purely linear transfer function “purelin” has been used in the output layer.

Table 4 Architecture of Modular ANN-based Fault Distance Locator

Modular ANN-based fault locator	Architecture	Mean square error (MSE)	Number of epochs
Phase to ground	3-16-1	5.023e−04	211
Phase to phase	3-18-1	3.89e−04	188
Double phase to ground	3-14-1	5.08e−05	300
Three phase	3-7-1	3.88e−04	227

6 Performance Results

The effectiveness of the new fault detection/classification and location scheme was tested for various fault conditions such as different fault locations L_f, different fault resistances R_f and different fault inception angles FIA for each fault type. The training and testing process were generated using the single line diagram of a 100 km, 400 kV transmission line shown in Fig. 4. This system has been simulated using Matlab Software Program and the obtained data of three phase currents and zero sequence current for pre-fault and post-fault are obtained. The obtained results are used for training and testing of neural detector/classifier and neural locator using "Matlab/neural network toolbox".

6.1 Fault Type Classification

Different fault types (R-G) 'R'-phase-ground fault, (S-G) 'S'-phase-ground fault, (T-G) 'T'-phase-ground fault, (R-T) 'R'-'T'-phases fault, (R-S-G) 'R'-'S'-phases-ground fault, (R-S-T) 'R'-'S'-'T'-phases fault simulated with various fault condition such as fault location L_f, fault resistance R_f and fault inception angle FIA which were not presented to the ANN during the training process. Table 5 shows the test results for the proposed fault classification algorithm using single ANN approach.

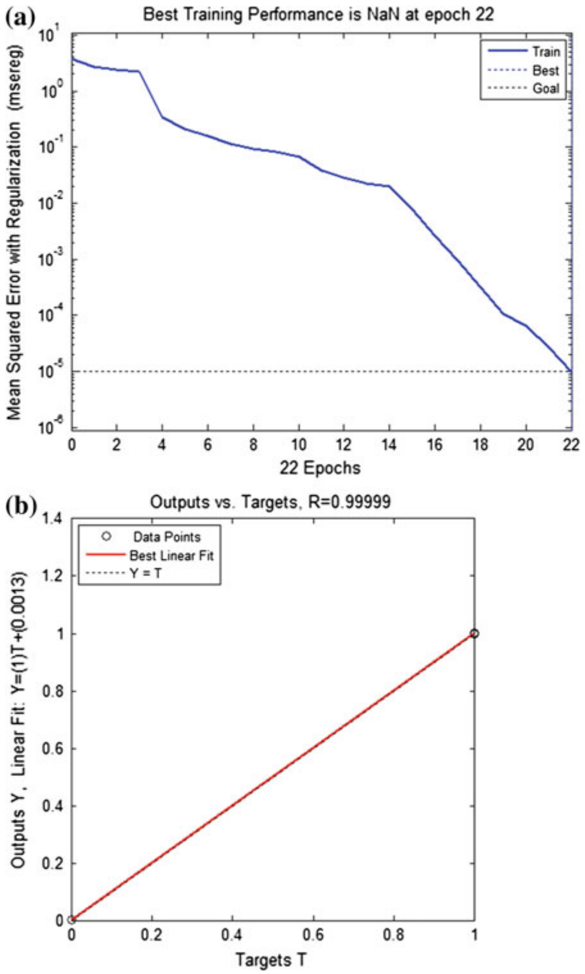
For each case it can be seen that the ANN outputs converge to the desired values, and are either very close to zero or to one. It is evident from the results that the ANN can accurately identify and classify the different faults in transmission lines.

6.2 Accurate Fault Location Using Modular ANNs Approach

6.2.1 Testing of the Fault Locator

Once the ANNs training procedure is entirely carried out, all networks of the fault locator are tested under different fault scenarios using different fault conditions which are not presented during the training process. All fault types with different

Fig. 8 **a** Mean-square error performance of the network. **b** Targets for the network



fault resistances R_f, different inception angles FIA and different fault location L_f in the transmission line are simulated in order to evaluate the performances of the proposed fault location scheme.

The criterion for evaluating the performance of the proposed neural fault locator is based on the following equation.

$$Absolute\ Error\ (\%) = \frac{|Estimated\ Distance\ -\ Actual\ Distance|}{Length\ of\ line} \cdot 100\,\% \qquad (7)$$

Some test results of single phase to ground and double phase to ground under different fault conditions presented in Tables 6 and 7. Also, the simulation

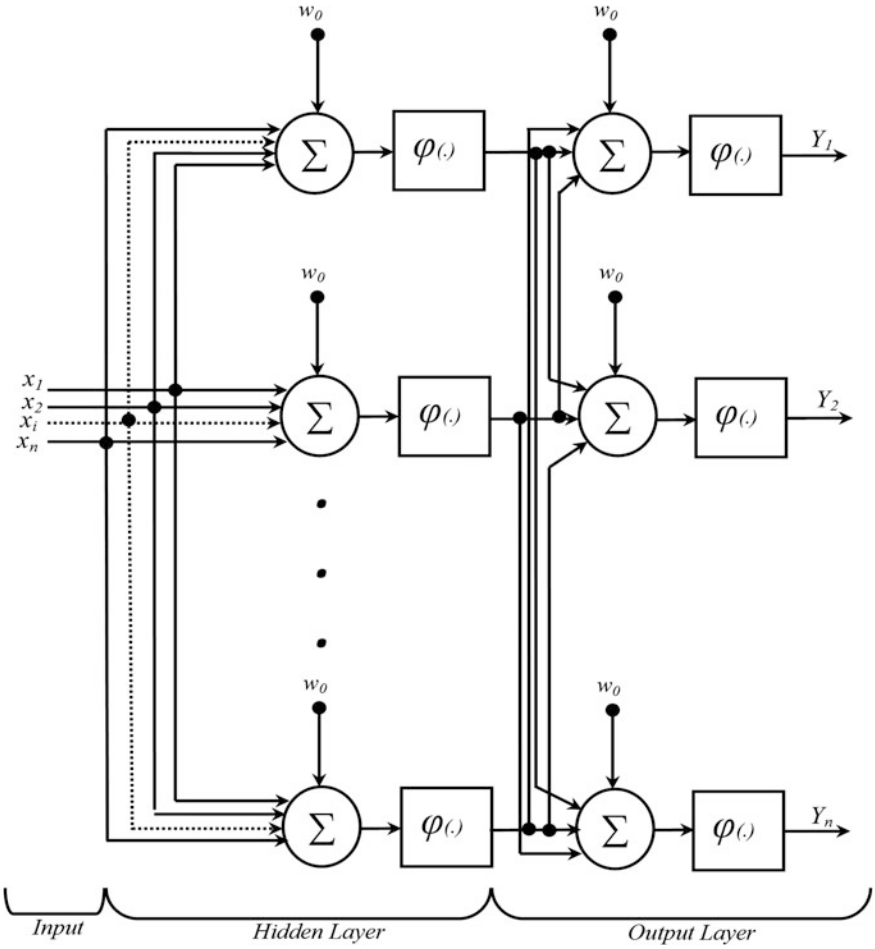


Fig. 9 A feed-forward multilayer for the fault classification and location

Table 5 Tests results for fault classification

Fault type	*R*	*S*	*T*	*G*
–	0	0	0	0
R-G	1.1082	0.1002	−0.0039	0.9910
S-G	−0.0059	0.9801	0.0941	1.0884
T-G	0.0129	−0.1103	1.1009	0.9739
R-T	0.9846	0.0023	1.0922	−0.0194
R-S-G	1.0291	1.1009	−0.1283	0.9956
R-S-T	1.0029	0.9998	0.9976	−0.0059

Table 6 Fault condition and percentage error for L-G, L-L-G faults

Fault conditions			Testing results of L-G		Testing results of L-L-G	
Fault location (km)	Fault inception angle (°)	Fault resistance (Ω)	Output of ANN based fault locator	Percentage error of ANN based fault locator	Output of ANN based fault locator	Percentage error of ANN based fault locator
08	325	26	08.0137	0.0137	08.0911	0.0911
12	125	130	12.0522	0.0522	12.1091	0.1091
24	145	2.6	23.8829	0.1171	23.8009	0.1991
32	275	100	32.2281	0.2281	32.1990	0.1990
48	05	65	48.1094	0.1094	48.2119	0.2119
53	25	95	53.0998	0.0998	52.6918	0.3082
62	215	22	62.3787	0.2787	62.2790	0.2210
77	10	13	77.1210	0.1210	77.3117	0.3117
88	40	44	87.8917	0.1083	88.3071	0.3071
95	135	35	95.3321	0.3321	95.4566	0.4566

Table 7 Fault condition and percentage error for L-L, L-L-L faults

Fault conditions			Testing results of L-L		Testing results of L-L-L	
Fault location (km)	Fault inception angle (°)	Fault resistance (Ω)	Output of ANN based fault locator	Percentage error of ANN based fault locator	Output of ANN based fault locator	Percentage error of ANN based fault locator
08	325	26	07.9751	0.0249	08.0983	0.0983
12	125	130	12.0908	0.0908	11.8299	0.1701
24	145	2.6	24.2079	0.2079	24.2229	0.2229
32	275	100	32.3009	0.3009	32.2007	0.2007
48	05	65	48.3291	0.3291	48.1399	0.1399
53	25	95	53.1812	0.1812	53.1678	0.1678
62	215	22	62.4097	0.4903	61.6911	0.3089
77	10	13	77.3088	0.3088	77.4018	0.4018
88	40	44	87.7191	0.2809	88.5091	0.5091
95	135	35	95.2791	0.2971	94.7117	0.2883

conditions and the percentage error for double phase and three phase fault are presented in Tables 6, 7 and 8.

The percentage error for 10 patterns for the transmission line for each fault type (phase to ground fault L-G, double phase to ground fault L-L-G, double phase fault and three phase fault L-L-L) are indicated in Figs. 10, 11, 12 and 13.

The minimum, maximum and average error percentages of the proposed fault locator are illustrated in Table 8. The average error value of this algorithm for the single phase ground fault was 0.1460 and 0.2512 % for the two phases fault and 0.2415 % for the two-phases to ground and 0.2512 % for the three-phase fault.

Table 8 Results error of fault locator

	% Error			
Fault type	L-G	L-L-G	L-L	L-L-L
Min	0.0137	0.0911	0.0249	0.0983
Max	0.3321	0.4566	0.4903	0.5091
Aver	0.1460	0.2415	0.2512	0.2508

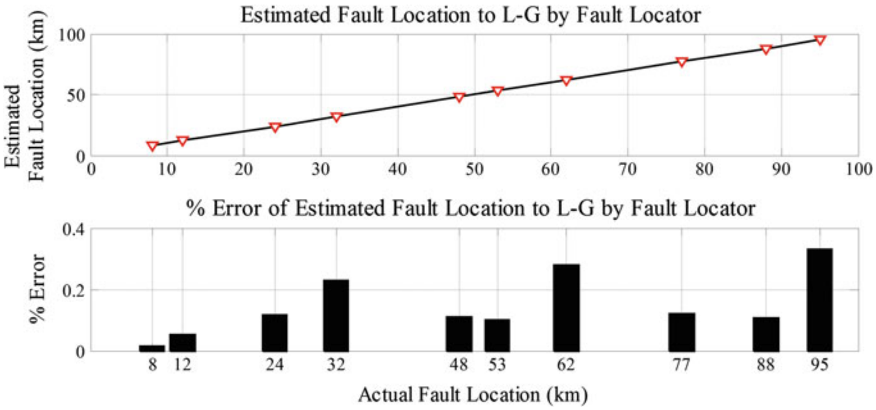


Fig. 10 Estimated fault location and percentage error during testing of L-G faults

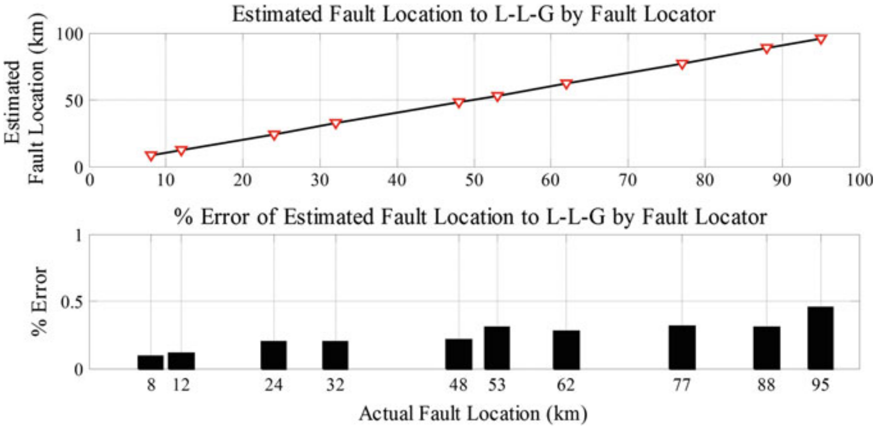


Fig. 11 Estimated fault location and percentage error during testing of L-L-G faults

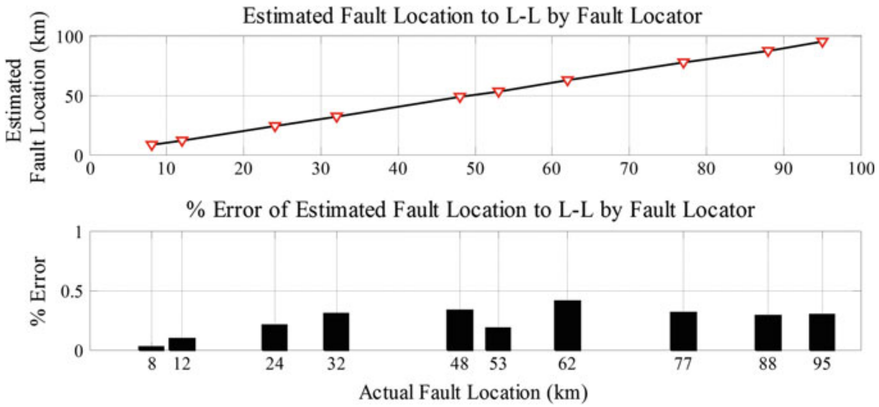


Fig. 12 Estimated fault location and percentage error during testing of L-L faults

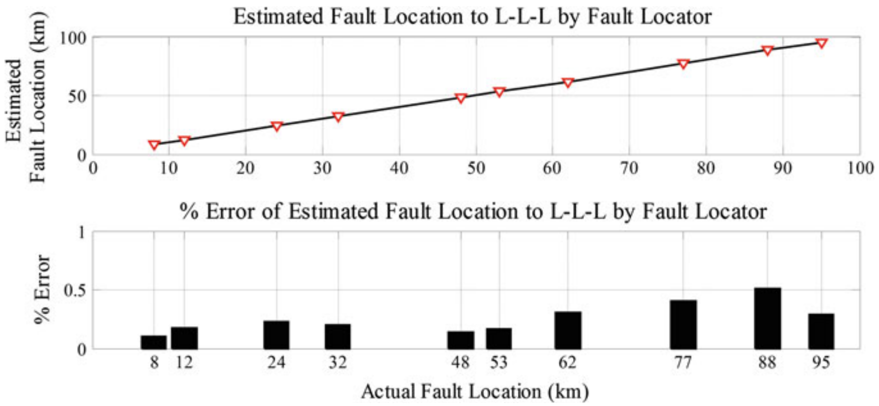


Fig. 13 Estimated fault location and percentage error during testing of L-L-L faults

The simulation results in these figures and tables prove the capacity of the fault locator to produce a correct answer in all simulations test. Moreover, the ANNs output stability under normal study state and under fault situations and the fast convergence of the output variables to the desired values in the presence of fault is well checked that confirms that the proposed fault locator algorithm is effective.

6.2.2 Performances Characteristics

In order to evaluate the performance of the ANN based on fault locator it is necessary to verify the following performance characteristics:

- Stability of ANN output based on fault locator under normal situation and under fault situation.
- Minimal response time of the proposed fault location based on modular ANN approach with the response time T_r equal the difference between the occurrence fault time T_f and the time where the ANN output is stabilized by indicating to the exact fault location T_e:

$$T_r = T_e - T_f \tag{8}$$

- Generalization capabilities.

A best ANN-based fault locator is selected when the response time is minimal. The only means of validating the performance of the neural network is to perform extensive testing. After training process, the ANN based fault locator is then extensively tested using different fault scenarios never used in training process.

In our study case, the ANN-based fault locator is trained to show the output as 110 km for no fault situation or for fault outside the segment on the line. For faults occurred on the segment of the transmission line the ANN is trained to show the estimated fault location as output.

In order to evaluate the response time of the proposed neural fault locator we have simulated various fault scenarios with different fault conditions.

First Scenario: Single Phase to Ground Fault with High Fault Conditions

To study the effect of high fault conditions such as fault resistance R_f, fault inception angle FIA and fault location L_f a single phase to ground fault (R-G) has been simulated with $R_f = 92\ \Omega$, $L_f = 86$ km and FIA = 135° corresponding to the fault appearance time at 71 s. Figure 14.

The response time of the ANN-based fault locator is simulated and depicted in Fig. 15. it can be seen that the ANN output is 85.8962 km what implies a precision of 0.1038 %. Thus, the fault is occurred at time T_f equal 71 s and will be located by the adopted neural locator at time T_e equal 71.02 s. What gives a response time T_r equal to 20 ms.

Second Scenario: Three Phase Fault Occurs Near to the Source

In other hand, we have study the case when a fault occurs near to the source end (side S) where the relays are installed. A three phase fault is simulated at 11 km

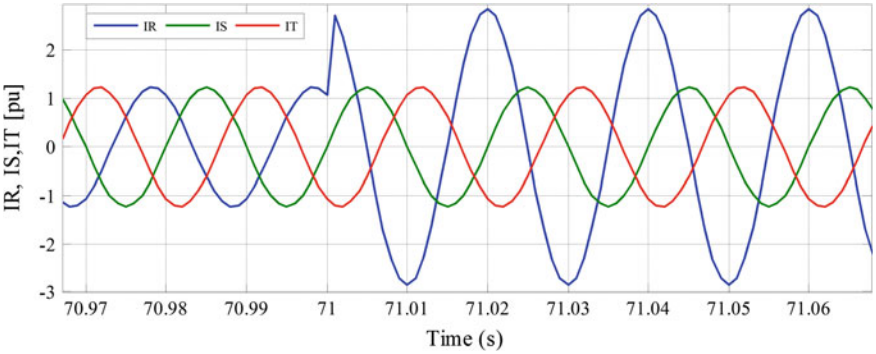


Fig. 14 Three phase currents during phase R to ground fault

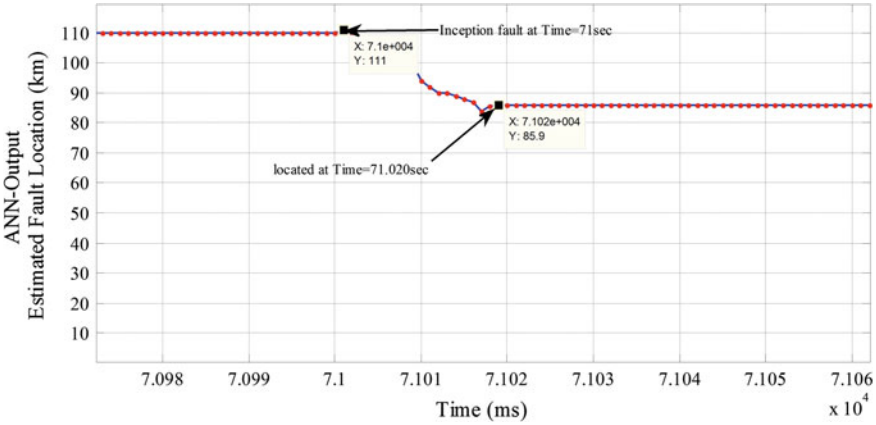


Fig. 15 Test result of fault locator during phase R to ground fault

from source (S) end. Test conditions where "R-S-T" fault with $L_f = 11$ km, $R_f = 0\ \Omega$ and occurred at time $T_f = 71$ s.

Test results of the proposed ANN-based fault locator under this condition are shown in Fig. 16. From the Figure, it can be seen that after one cycle from the inception of fault (71 s), that is at 71.022 s, is $L_f = 11.0967$ as against 11 km actual fault distance what implies a fast response time about $T_r = 22$ ms and a precision of 0.0967 %.

The ANN output is almost constant around the real fault location. Thus it is clear that the proposed ANN-based fault locator can precisely estimate the exact fault location in Extra High Voltage transmission line (EHV). Further, the operating time of the proposed algorithm is about one cycle time from the inception of fault (Fig. 17).

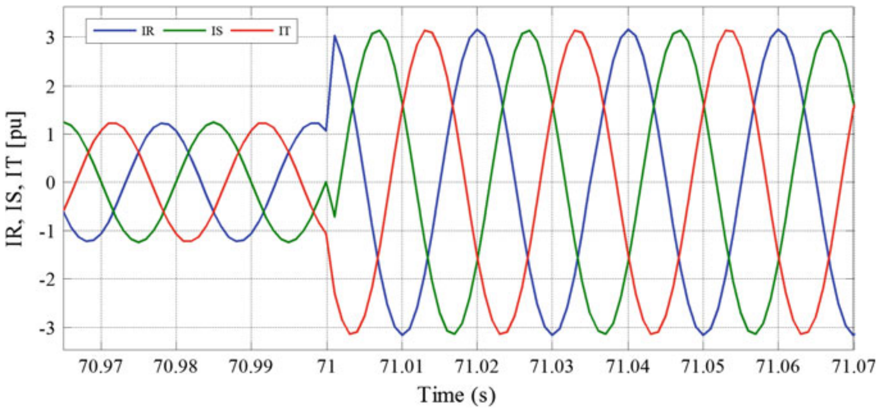


Fig. 16 Three phase currents during R-S-T fault

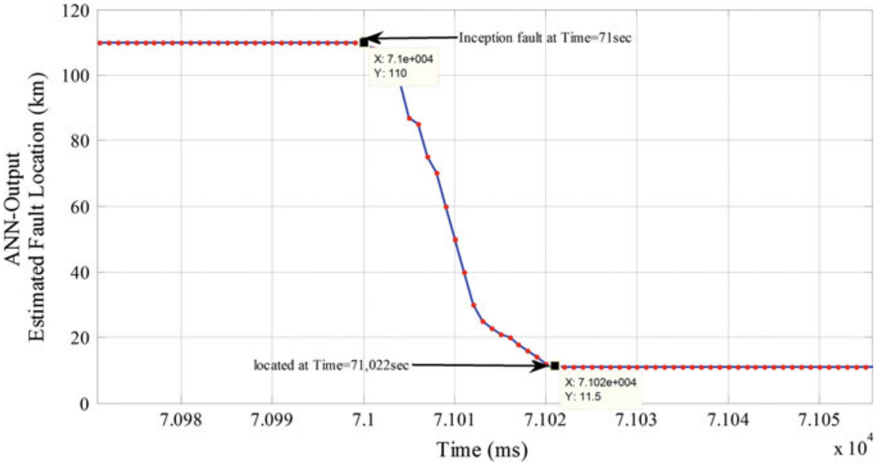


Fig. 17 Fault Locator test result during R-S-T fault

7 Discussion and Comparative Study

The salient features of some existing artificial neural network based fault detection/classification and location schemes and those of the proposed algorithms are presented in this section. The proposed fault protection scheme: fault detection/classification and fault location in Extra High Voltage EHV transmission lines is evaluated and compared with some former works. This adopted fault protection scheme has several advantages:

Table 9 Fault location schemes comparison

Schemes suggested	Algorithm used	Fault locator inputs	R_f range (Ω)	L_f range (km) (%)	FIA (°)	% Error
Mohammad and Javad [47]	Hybrid wavelet-prony	Voltages samples	100	0–90	0–180	0.72
Majid et al. [48]	Wavelet transform	Currents and voltages samples	100	0–90	0–180	10.5
Ekici et al. [26]	Support vector machine	Currents and voltages samples	50	0–90	0–180	7.9
Proposed scheme	Modular ANNs	Currents samples	100	0–95	0–360	0.2512

Table 10 Comparative results of fault location schemes using ANNs

Schemes suggested	Algorithm used	Fault locator inputs	R_f range (Ω)	L_f range (km) (%)	FIA (°)	% Error
Tahar [63]	Single ANN	Currents and voltages samples	Not indicate	0–80	Not indicate	0.64
Anamika and Thoke [4]	Single ANN	Currents and voltages samples	Not indicate	0–95	Not indicate	1.57
Jiang et al. [39]	Modular ANNs	Currents and voltages samples	Not indicate	0–90	0–180	0.5
Yilmaz [66]	Single ANN	Currents and voltages samples	50	0–80	Not indicate	3
Proposed scheme	Modular ANNs	Currents samples	100	0–95	0–360	0.2512

- Only current inputs are required for fault detection/classification and location.
- Wider range of different fault conditions such as fault resistance R_f, fault location L_f and fault inception angle FIA.
- Fast response better than, the existing schemes.

Table 9 compares the proposed fault location algorithm with some recent published methods used others tools such as the hybrid Wavelet-Prony and Wavelet Transform. Using only single ended current measurement is one of the salient advantages of the proposed method. The proposed algorithm exhibits better performance compared to the algorithms presented by Mohammad and Javad [47] and Majid et al. [48] which use only the single-ended current as well as voltage measurements. Indeed, the proposed fault location algorithm in this chapter has led to more accurate fault locating.

The proposed neural fault detector/classifier and locator are compared also with some published algorithms used the ANNs for identify and estimate the fault location in transmission lines, Table 10. In this context, the proposed algorithm and the algorithms proposed by Tahar [63], Jiang et al. [39], Anamika and Thoke [4] and Yilmaz [66] are similar in the sense that both are Neural Network based

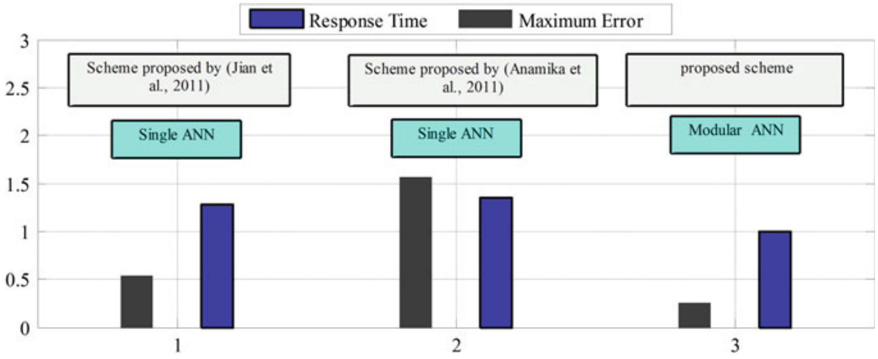


Fig. 18 Comparative study between ANN-based fault location

schemes which require the consideration of the measurement signals at one terminal of transmission line at the relay location. The proposed algorithm is applicable for a wider variation for fault conditions compared to the other methods. Whereas the latter is valid for fault resistance variation R_f up to 50 Ω, fault inception angle FIA up to 270° and fault location up to to 90 % of line length. By against, the proposed algorithm is valid for fault resistance variation R_f up to 100 Ω, fault inception variation angle FIA up to 360° and fault location variation L_f up to 95 % of line length.

Further, the proposed scheme for fault location is simple compared to other scheme because it requires the computation of ratios from the pre-fault and post-fault currents samples for identify the fault type. Furthermore, a filtering processor (one-cycle DFT) for extracting the magnitude of fundamental frequency components (50 Hz) necessary for estimate the fault location in transmission line.

We notice also another comparison of the proposed algorithm with the other works and this compared to the response time as shown in Fig. 18. Thus, the response time of the proposed scheme for detection/classification and location is about one cycle from the inception of fault which is comparable to the conventional distance relay.

8 Conclusion

An accurate fault location based on artificial neural network for fast protection in Extra High Voltage (EHV) transmission lines. The algorithm consists of two stages, including fault detection/classification and fault location. For fault detection/classification the single ANN approach is used, this approach uses the pre-fault and post-fault samples of three phase currents and zero sequence current. For fault location the modular ANN approach is used, this approach uses the magnitudes of fundamental components of three phase currents. Simulations studies carried out considering wide variations in fault conditions such as fault locations, fault inception angles and fault resistances for different fault types have proved the validity of the proposed scheme. The average error value of the proposed fault location algorithm for single phase to ground fault was 0.1460 and 0.2512 % for two phases fault and 0.2415 % for two phases to ground and 0.2512 % for three phase fault.

The response time of the proposed fault protection scheme is about one-cycle from the inception fault.

Consequently, the ANNs can be used for on-line fault classification and location in transmission lines.

References

1. Abdelaziza, A.Y., Mekhamera, S.F., Ezzat, M.: Fault location of uncompensated/series-compensated lines using two-end synchronized measurements. Electr. Power Compon. Syst. **41**(7), 693–715 (2013)
2. Abbassi, R., Chebbi, S.: Energy Management strategy for a grid–connected wind-solar hybrid system with battery storage: policy for optimizing conventional energy generation. Int. Rev. Electr. Eng. **7**(2), 3979–3990 (2012)
3. Abbasi, R., Marrouchi, S., Moez, B.H., Chebbi, S., Houda, J.: Voltage control strategy of an electrical network by the integration of the UPFC compensator. Int. Rev. Model. Simul. **5**(1), 380–384 (2012)
4. Anamika, Y., Thoke, A.S.: Transmission line fault distance and direction estimation using artificial neural network. Int. J. Eng. Sci. Technol. **3**(8), 110–121 (2011)
5. Ancell, G.B., Pahalawaththa, N.C.: Maximum likelihood estimation of fault location on transmission lines using travelling waves. IEEE Trans. Power Delivery **9**(2), 680–688 (1994)
6. Al-Shaher, M., Saleh, A.S., Sabry, M.M.: Estimation of fault location and fault resistance for single line-to-ground faults in multi-ring distribution network using artificial neural network. Electr. Power Compon. Syst. **37**(7), 697–713 (2009)
7. Aritra, D., Sudipta, N., Arabinda, D.: Transmission line fault classification and location using wavelet entropy and neural network. Electr. Power Compon. Syst. **40**(15), 1676–1689 (2012)
8. Apostolopoulos, C.A., Korres, G.N.: Accurate fault location algorithm for double-circuit series compensated lines using a limited number of two-end synchronized measurements. Int. J. Electr. Power Energy Syst. **42**(1), 495–507 (2012)
9. Azar, A.T.: A novel ANFIS Application for Prediction of Post-Dialysis Blood Urea Concentration. Int. J. Intell. Syst. Technol. Appl. **12**(2), 87–110 (2013)
10. Azar, A.T.: Fast neural network learning algorithms for medical applications. Neural Comput. Appl. **23**(3–4), 1019–1034 (2013)
11. Azar, A.T.: Overview of Type-2 Fuzzy logic systems. Int. J. Fuzzy Syst. Appl. **2**(4), 1–28 (2012)
12. Azar, A.T.: Neuro-fuzzy system for cardiac signals classification. Int. J. Model. Ident. Control **13**(12), 108–116 (2011)
13. Azar, A.T.: Adaptive neuro fuzzy system as a novel approach for predicting post-dialysis urea rebound. Int. J. Intel. Syst. Technol. Appl. **10**(3), 302–330 (2011)
14. Azar, A.T.: Fuzzy systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
15. Azar, A.T.: Adaptive neuro-fuzzy systems. In: Azar, A.T. (ed.) fuzzy systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
16. Banu, G., Suja, S.: ANN based fault location technique using one end data for UHV lines. Eur. J. Sci. Res. **77**(4), 549–559 (2012)

17. Barros, J., Drake, J.M.: Real time fault detection and classification in power systems using microprocessors. IEE Proc. Gener. Transm. Distrib. **141**(3), 315–322 (1994)
18. Bo, Z.Q., Weller, G., Lomas, T., Redfern, M.A.: Positional protection of transmission systems using global positioning system. IEEE Trans. Power Delivery **15**(4), 1163–1167 (2000)
19. Borghetti, A., Bosetti, M., Silvestro, D.M., Nucci, C.A., Paolone, M.: Continuous-wavelet transform for fault location in distribution power networks: definition of mother wavelets inferred from fault originated transients. IEEE Trans. Power Syst. **23**(2), 380–388 (2008)
20. Carlo, C., Kaveh, R.: Fuzzy-logic-based high accurate fault classification of single and double-circuit power transmission lines. In: The 2012 IEEE Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 20–22 June 2011, pp. 883–889. Sorrento (2012). doi:10.1109/SPEEDAM.2012.6264636
21. Chun, W., Qing, Q.J., Xin, B.L., Chun, X.D.: Fault location using synchronized sequence measurements. Electr. Power Energy Syst. **30**(2), 134–139 (2008)
22. Desikachar, K.V., Singh, L.P.: Digital travelling–wave protection of transmission lines. Electr. Power Syst. Res. **7**(1), 19–28 (1984)
23. D’Amore, D., Ferrero, A.: A simplified algorithm for digital distance protection based on fourier techniques. IEEE Trans. Power Delivery **4**(1), 157–164 (1989)
24. Dong, X., Kong, W., Cui, T.: Fault classification and faulted-phase selection based on the initial current travelling wave. IEEE Trans. Power Delivery **24**(2), 552–559 (2009)
25. Eriksson, L., Saha, M.M., Rockefeller, G.D.: An accurate fault locator with compensation for apparent reactance in the fault resistance resulting from remote-end infeed. IEEE Trans. Power Apparatus Syst. **104**(2), 424–435 (1985)
26. Ekici, J.S.: Support vector machines for classification and locating faults on transmission lines. Appl. Softw. Comput. **12**(6), 1650–1658 (2012)
27. Ernesto, V.M.: A travelling wave distance protection using principle component analysis. Int. J. Electr. Power Energy Syst. **25**(6), 471–479 (2003)
28. Ferrero, S., Sangiovanni., Zapitteli, E.: Fuzzy-set approach to type-faut identification in digital relaying. IEEE Trans Power Delivery. **10**(1), 169–175 (1995)
29. Gaganpreet, C.M., Sachdev, S., Ramakrishna, G.: Artificial neural network applications for power system protection. In: The 2005 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 1–4 May 2005, pp. 1954-1957. Saskatoon (2005). doi:10.1109/CCECE.2005.1557365
30. Girgis, A.A., Brown, R.G.: Adaptive Kalman filtering in computer relaying: fault classification using voltage models. IEEE Power Eng. Rev. **5**(5), 44–45 (1985)
31. Gracia, J., Mazón, A.J., Zamora, I.: Best ANN structures for fault location in single and double-circuit transmission lines. IEEE Trans. Power Delivery **20**(4), 2389–2395 (2005)
32. Gohokar, V.N., Khedkar, M.K.: Faults locations in automated distribution system. Electr. Power Syst. Res. **75**(1), 51–55 (2005)
33. Hassan, K.Z., Zuyi, L.: An ANN based approach to improve the distance relaying algorithm. Turkish J. Electr. Eng. Comput. Sci. **14**(2), 345–354 (2006)
34. Héctor, J.A.F., Ismael, D.V., Ernesto, V.M.: Fourier and Walsh digital filtering algorithms for digital distance protection. IEEE Tran. Power Syst. **11**(1), 457–462 (1996)
35. Huseyin, E.: Fault diagnosis system for series compensated transmission line based on wavelet transform and adaptive neuro-fuzzy inference system. Measurement **46**(1), 393–401 (2013)
36. Izykowski, J., Rosolowski, E., Balcerek, P., Fulczyk, M., Saha, M.: Fault location on double-circuit series-compensated lines using two-end unsynchronized measurements. IEEE Trans. Power Delivery **26**(4), 2072–2080 (2011)
37. Jang, J.S.R.: ANFIS: adaptive- network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–684 (1993)
38. Javad, S., Hamid, A.: A new and accurate fault location algorithm for combined transmission lines using adaptive network-based fuzzy inference system. Electr. Power Syst. Res. **79**(11), 1538–1545 (2009)

39. Jiang, J.A., Chuang, C.L., Wang, Y.C., Hung, C.H., Wang, J.Y., Lee, C.H., et al.: A hybrid framework for fault detection, classification, and location—Part I: concept, structure, and methodology. IEEE Trans. Power Delivery **26**(3), 1988–1998 (2011)
40. Jiang, J.A., Yang, J.Z., Lin, Y.H., Liu, C.W., Ma, J.C.: An adaptive PMU based fault detection/location technique for transmission lines Part-I; theory and algorithms. IEEE Trans. Power Delivery **15**(2), 486–493 (2000)
41. Joorabian, S.M.A., Taleghani, A.S.L., Aggarwal, R.K.: Accurate fault locator for EHV transmission lines based on radial basis function neural network. Electr. Power Syst. Res. **71** (3), 195–202 (2004)
42. Kola, V.B., Manoj, T., Asheesh, K.S.: Recent techniques used in transmission line protection: a review. Int. J. Eng. Sci. Technol. **3**(3), 1–8 (2011)
43. Lin, Y.H., Liu, C.W., Yu, C.S.: A new fault locator for three-terminal transmission lines using two-terminal synchronized voltage and current phasors. IEEE Trans. Power Delivery **17**(2), 452–459 (2002)
44. Magnago, F.H., Abur, A.: Fault location using wavelets. IEEE Trans. Power Delivery **13**(4), 1475–1480 (1998)
45. Mahamedi, B., Zhu, J.G.: Unsynchronized fault location based on the negative-sequence voltage magnitude for double-circuit transmission lines. IEEE Trans. Power Delivery **99**, 1 (2014)
46. Mahanty, R. N., Gupta, P. B. D.: A fuzzy logic based fault classification approach using current samples only. Electric Power System Research. **77**(5–6), 501–507 (2007)
47. Mohammad, F., Javad, S.: Transmission line fault location using hybrid wavelet-Prony method and relief algorithm. Int. J. Electr. Power Energy Syst. **61**, 127–136 (2014)
48. Majid, J., Abul, K., Ansari, A.Q., Rizwan, M.: Generalized neural network and wavelet transform based approach for fault location estimation of a transmission line. Appl. Softw. Comput. **19**, 322–332 (2014)
49. Moez, B.H., Houda, J., Souad, C.: Fault detection and classification approaches in transmission lines using artificial neural networks. In: The 2014 IEEE Mediterranean Electrotechnical Conference (MELECON), 13–16 April 2014, pp. 520-524. Beirut (2014) (In press)
50. Moez, B.H., Houda, J., Souad, C.: A new and accurate fault classification algorithm for transmission lines using fuzzy logic system. Wulfenia J. **20**(3), 336–349 (2013)
51. Moez, B.H., Houda, J., Souad, C., Sahbi, M.: Voltage and frequency stabilization of electrical networks by using load shedding strategy based on fuzzy logic controllers. Int. Rev. Electr. Eng. **7**(5), 5694–5704 (2012)
52. Moez, B.H., Sahbi, M., Souad, C., Houda, J., Rabeh, A.: Preventive and curative strategies based on fuzzy logic for voltage stabilization of an electrical network. Int. Rev. Model. Simul. **4**(6), 3201–3207 (2011)
53. Mora, F.J., Melendez, J., Carrillo, C.G.: Comparison of impedance based fault location methods for power distribution systems. Electr. Power Syst. Res. **78**(4), 657–666 (2008)
54. Nan, Z., Kezunovic, M.: Coordinating fuzzy ART neural networks to improve transmission line fault detection and classification. In: The 2005 IEEE Power Engineering Society General Meeting Conference (PES), 12–16 June 2005, pp. 734–740 (2005). doi:10.1109/PES.2005.1489373
55. Pei, Y.L., Tzu, C.L., Chih, W.L.: An intranet-based transmission grid fault location platform using synchronized IED data for the Taiwan power system. In: The 2013 IEEE Innovative Smart Grid Technologies (ISGT), 24–27 Feb 2013, pp. 1–6. Washington (2013). doi:10.1109/ISGT.2013.6497796
56. Rockefeller, G.D.: High speed distance relaying using a digital computer, II. Test results. IEEE Trans. Power Appar. Syst. **91**(3), 1244–1258 (1972)
57. Shehab-Eldin, E.H., Mclaren, P.G.: Travelling wave distance protection-problem areas and solutions. IEEE Trans. Power Delivery **3**(3), 894–902 (1988)
58. Man, B.J., Morrison, I.F.: Digital calculation of impedance for transmission line protection. IEEE Trans. Power Appar. Syst. **90**(1), 270–279 (1971)

59. Man, B.J., Morrison, I.F.: Relaying a three phase transmission line with a digital computer. IEEE Trans. Power Appar. Syst. **90**(2), 742–750 (1971)
60. Soon, R.N., Sang, H.K., Seon, J.A., Joon, H.C.: Single line-to-ground fault location based on unsynchronized phasors in automated ungrounded distribution systems. Electr. Power Sys. Res. **86**, 151–157 (2012)
61. Spoor, D., Zhu, J.G.: Improved single-ended traveling-wave fault-location algorithm based on experience with conventional substation transducers. IEEE Trans. Power Delivery **21**(3), 1714–1720 (2006)
62. Tabatabaei, A., Mosavi, M.R., Farajiparvar, P.A.: A travelling wave fault location technique for three-terminal lines based on wavelet analysis and recurrent neural network using GPS timing. In: The 2013 IEEE Smart Grid Conference (SGC), 17–18 Dec 2013, pp. 268–272. Tehran (2013). doi:10.1109/SGC.2013.6733830
63. Tahar, B.: Fault location in EHV transmission lines using artificial neural networks. Int. J. Appl. Math. Comput. Sci. **14**(1), 69–780 (2004)
64. Vasilic, S., Kezunovic, M.: Fuzzy ART neural network algorithm for classifying the power system faults. IEEE Trans. Power Delivery **20**(2), 1306–1314 (2005)
65. Xu, Z.Y., Jiao, S.H., Ran, L., Du, Z.Q.: An online fault-locating scheme for EHV/UHV transmission lines. IET Gener. Transm. Distrib. **2**(6), 789–799 (2008)
66. Yilmaz, A.: An alternative approach to fault location on power distribution feeders with embedded remote-end power generation using artificial neural networks. Electr. Eng. **94**(3), 125–134 (2012)
67. Youssef, O. A. S.: A novel fuzzy logic based phase selection tenchinque for power system relaying. Electric Power System Research. **68**(3), 175–184 (2004)
68. Zhao, W., Song, Y.H., Chen, W.R.: Improved GPS traveling wave fault locator for power cables by using wavelet analysis. Int. J. Electr. Power Energy Syst. **23**(5), 403–411 (2001)
69. Zhengyou, H., Ling, F., Sheng, L., Zhiqian, B.: Fault detection and classification in EHV transmission line based on wavelet singular entropy. IEEE Trans. Power Delivery **25**(4), 2156–2163 (2010)
70. Zhu, Y.: Fault location scheme for a multi-terminal transmission line based on current traveling wave. Int. J. Electr. Power Energy Syst. **53**, 367–374 (2013)

A New Approach for Flexible Queries Using Fuzzy Ontologies

Amira Aloui and Amel Grissa

Abstract Motivated by the demand for formalized representation of outcomes of data mining investigations and the successful results of using Formal Concept Analysis (FCA) and Ontology, this chapter addresses the task of constructing an ontology of data mining in order to support flexible query in large Database using FCA and Fuzzy Ontology. A new approach for automatic generation of Fuzzy Ontology of Data Mining (FODM), through the combination of conceptual clustering, fuzzy logic and FCA will be presented. Then, a new algorithm to support database flexible querying using the generated fuzzy ontology will be defined. The approach starts with the organization of the data in homogeneous clusters having common properties which allows to deduce the data's semantic. Then, it models these clusters by an extension of the FCA. This lattice will be used to build a core of ontology. This ontology will be represented, then, as a set of fuzzy rules as an efficient answers to flexible queries. We show that this approach is optimum because the evaluation of the query is not done on the set of starting data which is huge but rather by using the generated fuzzy ontology.

Keywords Data Mining · Clustering · Formal Concept Analysis · Fuzzy Logic · Ontology

1 Introduction

The diversity of the Database (DB) applications showed the limits of the Relational Database Management Systems (RDBMS) in particular in the querying field [11]. The traditional querying of a Relational DB (RDB) is qualified by "Boolean querying"

A. Aloui (✉)
Ecole Nationale d'Ingenieurs de Tunis, LR-SITI, Tunis, Tunisia
e-mail: aloui_amira@yahoo.fr

A. Grissa
Ecole Nationale d'Ingenieurs de Tunis, LIPAH, FST, Tunis, Tunisia
e-mail: amel.touzi@enit.rnu.tn

© Springer International Publishing Switzerland 2015
A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_13

with SQL for example, a query returns a result or nothing at all [47]. This querying surrounds a problem for certain applications. First of all, the user must know all the details concerning the diagram and the data from the database to express his preferences or he should use imprecise linguistic terms as “moderate”, means” to better characterize the sought-after data.

The aim of the database flexible querying is to extend this binary behaviour by introducing preferences into the query criteria [40]. Thus, an element returned over by a query will be “more or less” relevant according to user preferences. Generally, the proposed approaches treat the flexible query in case of the RDB but not in case of the large DB. This work focuses on flexible query in large DB. For this purpose, we suggest the use of ontologies to improve the performance of retrieving information.

In fact, recent research showed that adopting formal ontology to describe heterogeneous data sources has many benefits. It provides not only a uniform and flexible approach to integrate and describe such sources, but it can also support the final user in querying them and improving the usability of the integrated system. Unfortunately, many deficiencies still exist in ontology. On the one hand, it is difficult to determine the granularity of ontology. On the other hand, the depth of concept expression of ontology is still not enough [6]. Thus fuzzy ontology is introduced to solve the above problems. The application of formal concept analysis and concept lattice theory in ontology building and mapping not only makes the building automatic, but also makes the newly generated ontology more formalized. It should be a better way to combine formal concept analysis with ontology to express and process the knowledge. This new proposed method supports the task of formulating a request for a user in a specific domain. In fact, the ontology defines a vocabulary which is often richer than the logical schema of the underlying data and usually closer to the users own vocabulary. The ontology can be effectively exploited by the user in order to formulate a query that best captures their information need. Consequently, the user is constantly guided and assisted in this task because the intelligence is dynamically driven by reasoning over the ontology.

This new approach helps the user in choosing what is more appropriate for him respecting their information need and restricting the possible choices which are more relevant and meaningful in a given context by considering only some parts of the ontology. For those reasons, the user is free to explore the ontology without the worry of making a wrong choice at some point and can thus concentrate on expressing his need. Besides, queries can be specified through a refinement process consisting in the iteration of few basic operations: The user specifies, first, an initial request, then before constructing the ontology, we use the fuzzy logic techniques [3] and Formal Concept Analysis concept [52] to classify the data which will refine or delete some of the not used information, thus the number of concepts constructing the ontology is always less than the number of objects starting on which we apply the classification algorithm [29] because the application of FCA reduces considerably the complexity until the resulting query satisfies the need of the user, changes the level of granularity in the process of the evaluation of the ontology and apply the clustering operation. So, the interrogation will focus necessarily on clusters.

Thus, we start by generating a Meta-DB formed by a set of clusters resulting of a preliminary fuzzy classification on data. This set represents a reduced view of the initial BD and allows to deduct semantics of the initial DB. The data classification aims to divide a data set into subsets, called clusters, so that:

- All data in the same cluster are similar and data from different clusters are dissimilar.
- The number of clusters generated by a classification algorithm is always less than the number of objects starting on which we apply the classification algorithm.
- All objects belonging to the same cluster have the same properties.

In this context, the query is modelled knowing the set of clusters modelling the meta-DB. To generate the meta-DB, we use the concepts of Clustering, Formal Concept Analysis (FCA) and Ontology. Thus, the use of these methods is justified by:

- Fuzzy clustering has been a very successful data analysis technique as demonstrated in diverse areas like signal processing, monitoring, and medical diagnosis [4]. Clustering is a widely used technique in data mining application for discovering patterns in underlying data. Most traditional clustering algorithms are limited in handling datasets that contain categorical attributes. Conventional clustering means classifying the given data objects as exclusive subsets (clusters).That means we can discriminate clearly whether an object belongs to a cluster or not. However such a partition is insufficient to represent many real situations. However, in many real situations, there not exists an exact boundary between different clusters. Therefore a fuzzy clustering object belongs to overlapping clusters with some membership degree. In other words, the essence of fuzzy clustering is to consider not only the belonging status to the clusters, but also to consider to what degree do the object belong to the cluster
- Fuzzy logic is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. Fuzzy logic is used with one of the web mining technique i.e. Clustering [21]. The advantages of Fuzzy Systems as well as various applications are presented thoroughly in [2].
- FCA is a method for knowledge representation that takes advantage of the features of formal concepts [51];
- Ontology is used for knowledge sharing and reuse. It improves information organization, management and understanding. It was introduced for better description of information in objective world. Ontology has a significant role in the areas dealing with vast amounts of distributed and heterogeneous computer based information [8, 23, 24].

The remainder of the chapter is organised as follows: A brief introduction to some basic definitions of Formal Concept Analysis (FCA), ontology and flexible querying in Sect. 2. Before we present our generic and automatic method for Fuzzy Ontology generation in Sect. 5, an overview of existing problems and contributions in Sects. 3 and 4 are given. Section 6 details the step of extracting flexible Query from resulted ontology. Section 7 evaluates the proposed approach. Section 8 summarizes the chapter, enumerates the advantages and concludes with an outlook on future work.

2 Basic Concepts

In this section, we present the basic concepts of flexible querying, ontologies and Formal Concept Analysis (FCA).

2.1 Flexible Querying

Definition A flexible query is a query in which comprise vague descriptions and/or vague terms.

The traditional systems of interrogation distinguish two categories of data: those which satisfy the search criteria and those which do not satisfy them. The principle of the flexible interrogation aims to extend this bipolar behaviour by introducing the concept of approximate pairing. Thus, an element returned by a request will be at least relevant according to its satisfaction degree to the constraints of interrogation.

Four principal approaches have been proposed to express and evaluate the flexible queries:

- Use of the secondary criteria [33, 52].
- Use of the distance and the similarity [19].
- Expression of the preferences with linguistic terms [38].
- Modelling of the inaccuracy by the fuzzy subsets theory [14, 44]. A comparative study of the systems of flexible interrogation has been achieved in [42, 49].

The problem of the expression of the users preferences in the flexible queries received much attention these last years [13, 20, 32]. In general, it is possible to distinguish two families of approaches for the expression of the preferences: implicit and explicit.

In the implicit approach: Mechanisms of numerical scores, commensurable or not, are used to represent the preferences. In the first case, the values of preferences can be aggregated to deliver a total value and to define a total order on the answers. In the second case, when there is not commensurability, only a partial order of the answers, based on the order of Pareto is possible for the incomparable classes of answers are built. This approach is detailed in Lietard and Rocacher [35] and is illustrated by the Skyline operator [10] or in PreferenceSQL [32].

In the explicit approach: The preferences are specified by binary relations of preferences and in the majority of the cases, a partial order is obtained on the tuples.

In addition, the preferences can be divided into constraints (preferences obligatory) and wishes (optional preferences). This reveals that this bipolar vision [27] of the preferences makes it possible to bring a refinement of the set of the answers to satisfy the constraints, then, if possible, wishes. The preferences of the users can also be expressed by criteria of selection based on fuzzy sets. The predicates are not any more "all or nothing" but can be "more or less" satisfied. Other researchers used charts to model the preferences on a great number of alternatives. As an example, we can quote the Conditional Preferences Networks (CP-Nets) [26], which constitute a chart appraisal for modelling the preferences.

Bosc and Pivert [15] suggest the introduction of the preferences in the form of subsets of n-uplets (stratified divisor). Thus, they used the terms of "stratified divisor" and "stratified division". Consequently, an element x of the dividend will be more acceptable as it will be associated with a large number of subsets (Si) defining the divisor. Three types of requests studied by Bosc et al. are expressed in SQL language, where the dividend can be an intermediate relation and the stratified divisor is given explicitly by the user or is the result from sub queries.

As example of principal systems of interrogation with preferences, we can quote, the systems PreferenceSQL and Preference Queries [12] which are based on a partial order, consequently, they deliver to the user the not dominating tuples. Preference SQL also incorporates a concept of bipolarity in the Preferring clause. The system top-K queries [11, 26], Domshlak uses an ad hoc score function f and delivers the k better answers of the total order obtained by f. However, this score function remains difficult to establish. The SQLf language uses the fuzzy set theory to define the preferences and makes the assumption of commensurability. It offers a framework founded to combine obligatory preferences.

Our work is related to the introduction of certain flexibility into the query writing. In fact, the traditional database querying uses a query to find elements satisfying a Boolean condition. In certain applications, the user can find a difficulty to describe in a precise and clear way the information for which he is seeking. It can also express preferences on the search criterion level with various degrees of importance between these criteria. This is why the concept of flexible query was proposed in the database systems. Let us consider for instance the case of a person who is looking, in an advertisement database, an apartment close to the town center with an approachable cost. In order to express such preferences, this person can formulate a flexible query comprising the terms *"near"* and *"accessible"*. It can also express the fact that the price criterion is more significant than that of the distance.

2.2 Ontologies

Ontologies are content theories about the classes of individuals, properties of individuals, and relations between individuals that are possible in a specified field of knowledge [18]. They define the terms for describing our knowledge about the domain. An ontology of a domain is beneficial in establishing a common (controlled) vocabulary when describing a domain of interest. This is important for unification and sharing of knowledge about a domain and its connection with other domains.

In reality, there is no common formal definition of what an ontology is. All the same, most approaches share a few core items such as: concepts, a hierarchical IS-A-relation, and further relations. For the sake of generality, we do not discuss more specific features like constraints, functions, or axioms in this paper, instead we formalize the core in the following way:

Definition 1 A (*core*) *ontology* is a tuple **O = (C, is_a, R,** σ**)** where

- C is a set whose elements are called concepts
- is_a is a partial order on C (i.e., a a binary relation is_a $\subseteq$ C $\times$ C which is reflexive, transitive, and anti symmetric)
- R is a set whose elements are called relation names (or relations for short)
- σ: R $\rightarrow$ C^{+} is a function which assigns to each relation name its arity

In the last years, several languages have been developed to describe ontologies. As example, we can cite, the Resource Description Framework (RDF) [17, 34], the Ontology Web Language (OWL) [7] and extension of OWL language like OWL 2 [37] or Fuzzy OWL [9]. The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies and Description Logics (DL) are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. Today description logic has become a cornerstone of the Semantic Web for its use in the design of ontologies. The Web Ontology Language Description Logics (OWL DL) becomeless suitable in domains in which the concepts to be represented do not have precise definitions. In our case, this scenario is, unfortunately, likely the rule rather than an exception. To handle this problem, the use of fuzzy ontology offers a solution. Classical ontology languages are not appropriate to deal with imprecision or vagueness in knowledge. Therefore, DL for the semantic web can be enhanced by various approaches to handle probabilistic or possibilistic uncertainty and vagueness. Although fuzzy logic was introduced already in the 1960's [54], the research on fuzzy ontologies was almost non-existent before 2000, so we can claim that this is a fairly new research field with a great potential. This is even more surprising considering that Pena (1984) reasoned already in the 1980's why the use of fuzzy logic as the basis for ontology building would be beneficial and solve many problems pertaining to classical ontologies. He proposes "to reject the maximality rule, according to which only altogether true sentences are true, and embracing instead the rule of endorsement, which means that whatever is more or less true is true". Among the advantages of fuzzy ontology he mentions:

- Positing fuzzy predicates usually simplifies our theories in most scientific fields.
- Fuzzy predicates are much more plausible, and give us a much more attractive and cohesive worldview, than their crisp counterpart.
- Degree-talk and comparative constructions.

Also, the number of environments and tools for building ontologies have grown exponentially. These tools provide support for the ontology development process and for the subsequent ontology usage. Among these tools, we can mention the most relevant: Ontolinguav [28], WebOnto [25], WebODE [1], Protégé-2000 [39], OntoEdit [48] and OilEd [7].

In this work, we propose to use fuzzy-OWL2 language iteself to generate automatically scripts from fuzzy ontologies. More precise, we use Protégé 4.2 as an OWL2 editor for fuzzy ontology representation.

2.3 Fuzzy Conceptual Scaling and FCA

Conceptual scaling theory is the key part of a Formal Concept Analysis (FCA). It allows the introduction of the given data and embed much more general scales than the usual chains and direct products of chains. In the direct products of the concept lattices of these scales, the given data can be embedded. FCA starts with the notion of a formal context specifying which objects have what attributes and thus a formal context may be viewed as a binary relation between the object set and the attribute set with the values 0 and 1.

In Tran et al. [50], an ordered lattice extension theory has been proposed: Fuzzy Formal Concept Analysis (FFCA), in which uncertainty information is directly represented by a real number of membership values in the range of [0,1]. This number is equal to similarity which is defined as follows:

Definition 2 The similarity of a fuzzy formal concept $C_1 = (\varphi(A_1), B_1)$ and its subconcept $C_2 = (\varphi(A_2), B_2)$ is defined as:

$$S(C_1, C_2) = \frac{|\varphi(A_1) \cap \varphi(A_2)|}{|\varphi(A_1) \cup \varphi(A_2)|}$$

where ∩ and ∪ refer intersection and union operators on fuzzy sets, respectively.

In Sassi et al. [45], we showed that these FFCA are very powerful as well with the interpretation of the results of the Fuzzy Clustering and in the optimization of the flexible query.

3 Related Work

Many researchers in the field of data mining have tried to find the efficient way to respond to the user query. We study in this section the most important approaches that generate information from data.

- **Approaches based on concept lattices for information retrieval.**

A first detailed formalization of how to use lattices for information retrieval appears to date back to Mooers (1958). His approach is contained in Salton's (1968) famous book and originally received some attention [46] but has not been further elaborated in the mainstream information retrieval community. Most of the few, current applications of lattices in information retrieval are based on formal concept analysis [29], which was invented in the early 1980's and relates lattices to object-attribute matrices or document-term matrices in information retrieval. Formal concept analysis applications to information retrieval are similar to Mooers's ideas but have been developed independently.

Quan Thanh et al. [43] proposed to incorporate fuzzy logic into FCA to make FCA deal with uncertainty in data and reasonably interpret the concept of hierarchy. The proposed framework is known as Fuzzy Formal Concept Analysis (FFCA). They use FFCA for automatic generation of ontology for scholarly semantic web. Concept lattices have been also applied in search of information at the onset of formal concept analysis [29]. A restriction of information retrieval by lattice is the theoretical complexity of the number of concepts for a context large number of objects or properties. More solutions to control the size of the lattice corresponding to the major contexts have been proposed in our approach.

- **Approaches based on domain ontology to improve the performance of information retrieval.**

The ontology building is usually performed manually, but researchers try to build an ontology automatically or semi automatically to save the time and the efforts of building the ontology.

Clerkin et al. [22] used concept clustering algorithm (COBWEB) to automatically discover and generate ontology. They argued that such an approach is highly appropriate to domains where no expert knowledge exists, and they propose how they might employ software agents to collaborate, in the place of human beings, on the construction of shared ontologies.

Wuermli and Joller [53] used different ways to build ontologies automatically, based on data mining outputs represented by rule sets or decision trees. They used the semantic web languages, RDF, RDF-S and DAML + OIL for defining ontologies. The problem with those approaches is that they are constructed ontology that do not describe the complete domain of data mining, but are simply made with a specific task in mind. Also, some existing ontology-based information retrieval approaches use RDF [41].

Mena et al. [36], Baer et al. [5], and Kapetanios et al. [31] provided insufficient knowledge for query reformulation. These approaches also lack the details of what needs to be included in the ontology from the data sources along with the domain knowledge to drive the process of query reformulation. The focus of these approaches (for example Kapetanios et al. [31]) remains towards interactive query generation through nondirected graphs supporting multiple natural languages.

- **Approaches based on Query to improve performance.**

Four principal concepts were proposed in the traditional approaches to express and evaluate the flexible queries:

- The use of the secondary criteria,
- The expression of the preferences with linguistic terms,
- The modeling of the inaccuracy by the fuzzy subset theory.

Ounalli and Belhadj [40] proposed a relieving approach within the fuzzy set framework. This approach appears too promising. The first contribution is to take into consideration the semantic dependencies between the query research criteria and to determine its reliability or not. The second contribution relates to its co-operative aspect in the flexible interrogation. For the dependencies extraction, this approach consists on building TAH's and MTAH from relieving attributes. The problem here lies in storage, indexing of such structures and the incremental update of these structures. To fullfill such works, fundamental research was focused on the following problems:

- Flexible queries formulation and evaluation,
- Vague or fuzzy data description and processing,
- Definition and use of fuzzy dependencies,
- Fuzzy Data Mining [30].

4 Motivation and Contributions

We have faced two types of problems:

- **At the level of flexible query:** The majority of the current approaches presented to support flexible queries have several limits, in particular, in The consideration of the dependencies between the search criteria that permit to detect the unreliable requests (having an empty answer) with the user, and the generation of the turned over approximate answers.
- **At the level of the ontology approaches:** several approaches have been proposed, but, generally these authors don't propose any solutions for the evaluation of the queries knowing onthologies generated by their approaches.

Several algorithms for Data Mining try to trace the decision tree or the FCA or one of these extensions to extract the association rules. In this case, researchers always focus on giving an optimum set of rules modeling in a faithful way the starting data unit, after having done a data cleansing step and an elimination of invalid-value elements. Accordingly, the limits of these approaches reside in the extraction of this ontology starting from the data or a data variety, which may be huge. As a result, we note the following limits:

- The rules generated from these data are generally redundant rules.
- These algorithms generated a very big number of rules, almost thousands, that the human brain cannot even assimilate.
- Generally the goal from extracting a set of rules is to help the user to give semantics of data and to optimize the information research. This fundamental constraint is not taken into account by these approaches.

To resolve these problems, we propose:

- A new approach for the ontology generation using conceptual clustering, fuzzy logic, and FFCA. We propose to define rules (Meta-Rules) between classes resulting from a preliminary classification on the data. Indeed while classifying data, we construct homogeneous groups of data having the same properties, so defining rules between clusters implies that all the data elements belonging to those clusters will be necessarily dependent on these same rules. Thus, the number of generated rules is smaller since one processes the extraction of the knowledge on the clusters which number is relatively lower compared to the initial data elements.
- A new algorithm to support database flexible querying using the generated knowledge in the first step. This approach allows the end-user to easily exploit all knowledge generated.

5 Presentation of the Fuzzy Ontology of Data Mining: FODM

5.1 Principe of the FODM

In this section, we present the architecture of the Fuzzy Ontology of Data Mining (FODM) approach and the process of fuzzy ontology construction.

Our FODM approach takes the database records and provides the corresponding fuzzy ontology. Figure 1 shows our proposed FODM approach. We suggest the ontology definition between classes resulting from a preliminary classification of the data. The FODM approach is organized according to two following main steps. Data Organization step and Fuzzy Ontology Generation step.

5.2 Theoretical Foundation of the FODM Model

In this part, we provide the theoretical foundations of the proposed approach, based on the following properties:

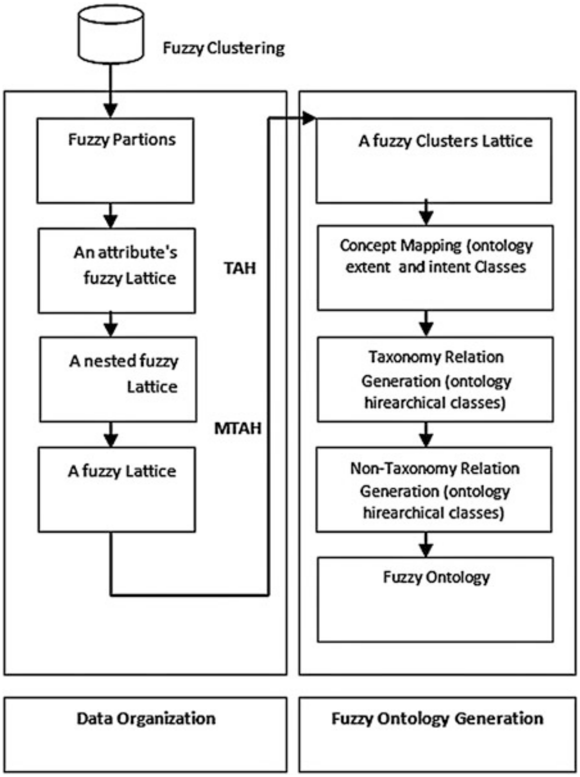

Fuzzy Clustering
Fuzzy Partions
An attribute's fuzzy Lattice
TAH
A nested fuzzy Lattice
MTAH
A fuzzy Lattice
A fuzzy Clusters Lattice
Concept Mapping (ontology extent and intent Classes
Taxonomy Relation Generation (ontology hirearchical classes)
Non-Taxonomy Relation Generation (ontology hirearchical classes)
Fuzzy Ontology
Data Organization
Fuzzy Ontology Generation

Fig. 1 Presentation of the fuzzy ontology of data mining approach

Property 1

- The number of clusters generated by a classification algorithm is always lower than the number of starting objects.
- All objects belonging to one same cluster have the same proprieties. These characteristics can be deduced easily knowing the center and the distance from the cluster.
- The size of the lattice modeling the properties of the clusters is lower than the size of the lattice modeling the properties of the objects.
- The management of the lattice modeling the properties of the clusters is optimum than the management of the lattice modeling the properties of the objects.

Property 2 Let C1, C2 be two clusters, generated by a classification algorithm and verifying respectively the properties p1 and p2. Then the following properties are equivalent:

$$C1 \Rightarrow C2\,(CR) \Leftrightarrow$$

- ∀ object O1∈C1 ⟹ O1∈C2 (CR),
- ∀ object O1∈C1, O1 checks the property p1 of C1 and the property p2 of C2 (CR)

Property 3 Let C1, C2 and C3 be three clusters generated by a classification algorithm and verifying respectively the properties p1, p2 and p3 respectively. Then the following properties are equivalent:

$$\mathrm{C1}, \mathrm{C2} \Rightarrow \mathrm{C3}\ (\mathrm{CR}) \Leftrightarrow$$

- ∀ object O1 ∈ C1 ∩ C2 ⟹ O1 object ∈ C3 (CR).
- ∀ object O1 ∈ C1 ∩ C2 then O1 checks the properties p1, p2 and p3 with (CR).

The validation of the two properties come from to the fact that all objects which belong to a same cluster check necessarily the same attribute as their cluster.

5.3 Data Organization Step

This step gives a certain number of clusters for each attribute. Each tuple has values in the interval [0,1] representing these membership degrees. Linguistic labels, which are fuzzy partitions, will be assigned to the attributes. This step consists on TAH's and MTAH generation of relieving attributes. This step is very important in the Fuzzy ontology generation process because it allows to define and interprete the distribution of objects in the various concepts.

Example Let's have a relational database table presented by Table 1 containing the list of AGE and SALARY of Employee

Table 2 presents the results of fuzzy clustering applied to Age and Salary attributes. For Salary attribute, fuzzy clustering generates three clusters (C1, C2 and C3). For the attribute AGE, two clusters have been generated (C4 and C5).

Table 1 Relational database table

	Salary	Age
A_1	800	30
A_2	600	35
A_3	400	26
A_4	900	40
A_5	1,000	27
A_6	500	30

Table 2 This fuzzy conceptual scales for age and salary attributes

	Salary			Age	
	C1	C2	C3	C4	C5
A_1	0.1	0.5	0.4	0.5	0.5
A_2	0.3	0.6	0.1	0.4	0.6
A_3	0.7	0.2	0.1	0.7	0.3
A_4	0.1	0.4	0.5	0.2	0.8
A_5	–	0.5	0.5	0.6	0.4
A_6	0.5	0.5	–	0.5	0.5

Table 3 This fuzzy conceptual scales for age and salary attributes with α-cut

	Salary			Age	
	C1	C2	C3	C4	C5
A_1	–	0.5	0.4	0.5	0.5
A_2	0.3	0.6	–	–	0.6
A_3	0.7	–	–	0.7	–
A_4	–	0.4	0.5	–	0.8
A_5	–	0.5	0.5	0.6	–
A_6	0.5	0.5	–	0.5	0.5

We apply an α-cut to the set of membership degrees, to replace these last by values 1 and 0 and to deduce the binary reduced formal context.

In our example, α-cut (Salary) = 0.3 and, α-cut (Age) = 0.5, so, the Table 2 can be rewritten as shown in Table 3.

The corresponding fuzzy concept lattices of fuzzy context presented in Table 3, noted as TAH's are given by the line diagrams presented in the Figs. 2 and 3.

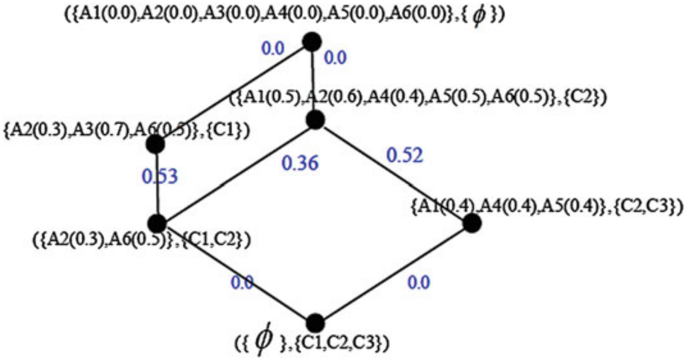

({A1(0.0),A2(0.0),A3(0.0),A4(0.0),A5(0.0),A6(0.0)},{ ϕ })
0.0
0.0
({A1(0.5),A2(0.6),A4(0.4),A5(0.5),A6(0.5)},{C2})
{A2(0.3),A3(0.7),A6(0.5)},{C1})
0.53
0.36
0.52
{A1(0.4),A4(0.4),A5(0.4)},{C2,C3})
({A2(0.3),A6(0.5)},{C1,C2})
0.0
0.0
({ ϕ },{C1,C2,C3})

Fig. 2 Salary TAH

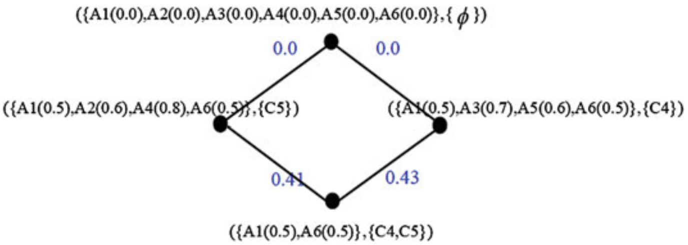

({A1(0.0),A2(0.0),A3(0.0),A4(0.0),A5(0.0),A6(0.0)},{ ϕ })
0.0
0.0
({A1(0.5),A2(0.6),A4(0.8),A6(0.5)},{C5})
({A1(0.5),A3(0.7),A5(0.6),A6(0.5)},{C4})
0.41
0.43
({A1(0.5),A6(0.5)},{C4,C5})

Fig. 3 Age TAH

The minimal (resp. maximal) value of each cluster corresponds to the lower (resp. higher) interval terminal of the values of this last. Each cluster of a partition is labelled with a ***linguistic label*** provided by the user or a domain.

The Table 4 presents the correspondence between the linguistic labels and their designations for the attributes Salary and Age.

The fuzzy concept lattices of fuzzy context presented in Table 5, noted as TAH's are given by the line diagrams presented in Figs. 2 and 3.

This very simple sorting procedure gives us for each many-valued attribute the distribution of the objects in the line diagram of the chosen fuzzy scale. Figure 4 shows the fuzzy nested lattice constructed from Figs. 2 and 3.

Table 4 Correspondence of the linguistic labels and their designations

Attribute	Linguistic labels	Designation
Salary	Low	C1
Salary	Medium	C2
Salary	High	C3
Age	Young	C4
Age	Adult	C5

Table 5 Fuzzy conceptual scales for age and salary attributes with α-cut

	Salary			Age	
	C1	C2	C3	C4	C5
	Low	Med	High	Young	Adult
A_1	–	0.5	0.4	0.5	0.5
A_2	0.3	0.6	–	–	0.6
A_3	0.7	–	–	0.7	–
A_4	–	0.4	0.5	–	0.8
A_5	–	0.5	0.5	0.6	–
A_6	0.5	0.5	–	0.5	0.5

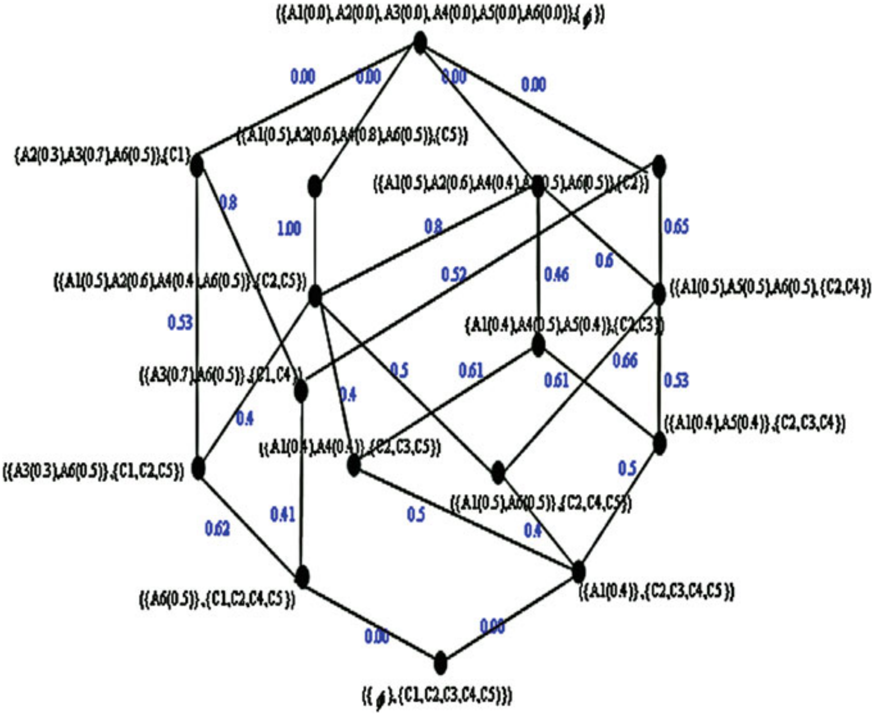

({A1(0.0), A2(0.0), A3(0.0), A4(0.0),A5(0.0),A6(0.0)},{ϕ})
0.00
0.00
0.00
0.00
({A1(0.5),A2(0.6),A4(0.8),A6(0.5)},{C5})
{A2(0.3),A3(0.7),A6(0.5)},{C1}
({A1(0.5),A2(0.6),A4(0.4),A5(0.5),A6(0.5)},{C2})
0.8
1.00
0.8
0.65
({A1(0.5),A2(0.6),A4(0.4),A6(0.5)},{C2,C5})
0.52
0.46
0.6
({A1(0.5),A5(0.5),A6(0.5)},{C2,C4})
0.53
{A1(0.4),A4(0.5),A5(0.4)},{C2,C3})
({A3(0.7),A6(0.5)},{C1,C4})
0.5
0.61
0.61
0.66
0.53
0.4
0.4
({A1(0.4),A5(0.4)},{C2,C3,C4})
({A1(0.4),A4(0.4)},{C2,C3,C5})
({A3(0.3),A6(0.5)},{C1,C2,C5})
0.5
({A1(0.5),A6(0.5)},{C2,C4,C5})
0.62
0.41
0.5
0.4
({A1(0.4)},{C2,C3,C4,C5})
({A6(0.5)},{C1,C2,C4,C5})
0.00
0.00
({ϕ},{C1,C2,C3,C4,C5})

Fig. 4 Fuzzy lattice: MTAH

5.4 Fuzzy Ontology Generation Step

This step consists on constructing the Fuzzy Ontology. It aims to deduce the Fuzzy Cluster Lattice corresponding to MTAH lattice generated in the first step, then to generate Ontology Extent and Intent Classes, Ontology hierarchical Classes, Ontology Relational Classes and finally the Fuzzy Ontology.

Definition (*Fuzzy Clusters Lattice*) A *fuzzy Clusters Lattice* (FCL) of a Fuzzy Formal Concept Lattice, consists on a Fuzzy concept lattice for which each equivalence class (i.e., a node of the lattice) contains only the intentional description (intent) of the associated fuzzy formal concept. This lattice will be used to build the core of the ontology.

Definition (*Level of FCL*) A *level i* of FCL is a is the set of nodes of FCL with cardinality equal to i.

Definition (*Concept Hierarchy*) A concept hierarchy is a poset (partially ordered set) (H, <), where H is a finite set of concepts and < is a partial order on H.

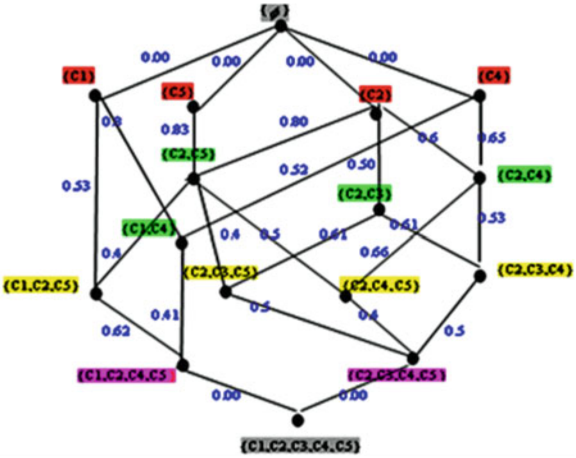


Fig. 5 Fuzzy clusters lattice (*FCL*)

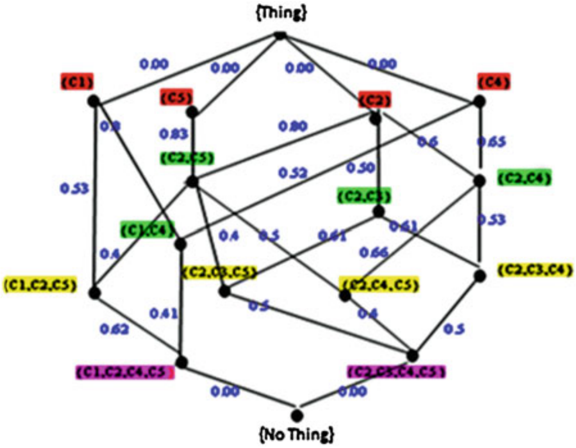


Fig. 6 Fuzzy ontology lattice

We make in this case a certain abstraction on the list of the objects with their degrees of membership in the clusters. The nodes of FCL are clusters ordered by the inclusion relation. As shown in the Fig. 5, we obtain a lattice more reduced, simpler to traverse and to store. Figure 6 illustrates the hierarchical relations constructed from the conceptual clusters given in 5. Each concept in the concept hierarchy is represented by a set of its attributes.

The supremum and infimum of the lattice are considered respectively as "Thing" and "Nothing" concepts.

The next step constructs fuzzy ontology from a fuzzy context using the concept hierarchy created by fuzzy conceptual clustering. This is done because both FCA and ontology support formal definitions of concepts. Thus, we define the fuzzy ontology as follows:

Definition (Fuzzy Ontology) A fuzzy ontology Fo contains of four elements (C, A^C, R, X), where:

- C represents a set of concepts,
- A^C represents a collection of attribute sets, one for each concept,
- R = (R_T; R_N) represents a set of relationships, having two elements: R_N is a set of non taxonomy relationships and R_T is a set of taxonomic relationships.
- X is a set of axioms. Each axiom in X is a constraint on the concept's and relationship's attribute values or a constraint on the relationships between concept objects.

We briefly describe the ontology mapping process from context to an ontology. The principal schema of one to one corresponding relations among the elements of FCA and OWL ontology are shown in Fig. 7.

- **Concept Mapping:** The mapping of concepts is one of the important stages of construction of a Fuzzy Ontology from a lattice of fuzzy concepts. It maps the extent and intent of the fuzzy context into the extent and intent classes of the ontology.
- **Taxonomy Relation Generation:** It expands the intent class of the ontology as a hierarchy of classes using the concept hierarchy. The process can be considered as an isomorphic mapping from the concept hierarchy into taxonomy classes of the ontology.
- **Non-taxonomy Relation Generation:** It generates the relation between the extent class and intent classes. This task is quite straight forward. However, we still need to label the non-taxonomy relation.
- **Instances Generation:** It generates instances of the extent class. Each instance corresponds to an object in the initial fuzzy context. Based on the information available on the fuzzy concept hierarchy, instances, attributes are automatically furnished with appropriate values.
- **Semantic Representation Conversion:** The generated ontology with concept hierarchies in Protégé-2000 [39] is shown in Fig. 8. This schema introduces the transformation rules for the automatic generation of OWL ontology based on the analysis of the concept hierarchy derived from FCA.

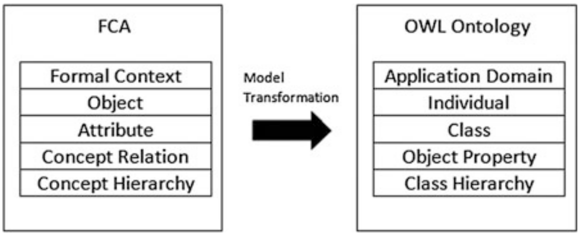


Fig. 7 From FCA to OWL ontology

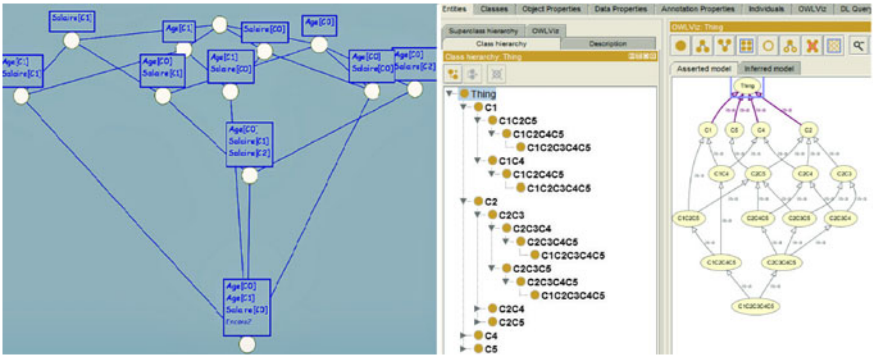


Fig. 8 From cluster FCA to Protégé-2000 to querying

5.5 *Algorithm for Automatically Generating a Fuzzy Ontology*

This section presents the main steps of our algorithm:

1. Assign the superclass "Thing" to the root of the concept lattice noted TOP.
2. For each concept of the level 1 of the lattice:
   (a) Create a subclass of the root class: Thing.
   (b) Assign the value 0 as a membership degree to the Thing class.
3. From the level 1, traverse the lattice and for each concept:
   (a) Find its successors/sub-concepts.
   (b) Put the given subconcepts in a set noted SetSucc.
4. Browse all successors of the SetSucc: For each successor belonging to SetSucc:
   (a) Create a subclass of the previous class.
   (b) Assign a membership value from each super concept.

The algorithm for generating a Fuzzy Ontology is described bellow:

Algorithm 1: Generation algorithm from FCL to Fuzzy Ontology

```
Input: Fuzzy Concept Lattice
Output: Fuzzy Ontology
begin
    initialization;
    Nf= Number of fuzzy concept;
    Nf1 = Number of fuzzy concept from level 1;
    hasValue = Create DataProperty;
    Thing Concept = 0, TOP;
    for i :=1 to Nf1 do
        Concept (i) = class (i);
        Class (i) = Create subclass From (Thing);
        Class (i) = HasValue (0.0) From (Thing);
    end
    for i:=1 to Nf do
        SetSucc = Succ (concept (i));
        for j:=1 to Number (SetSucc) do
            Under Class (i) = SetSucc (j);
            Under Class (i) = HasValue (SetSucc (j)) From SetSucc (j)
        end
    end
end
```

An example of ontology mapping is illustrated in Fig. 8

In this example, we used the domain "Employe" to illustrate the building process and to evaluate the resulted queries. We deduce the fuzzy lattice from our platform "ClusterFCA", then we construct the ontology using Protégé-4.2 while taking the FCA as a guideline.

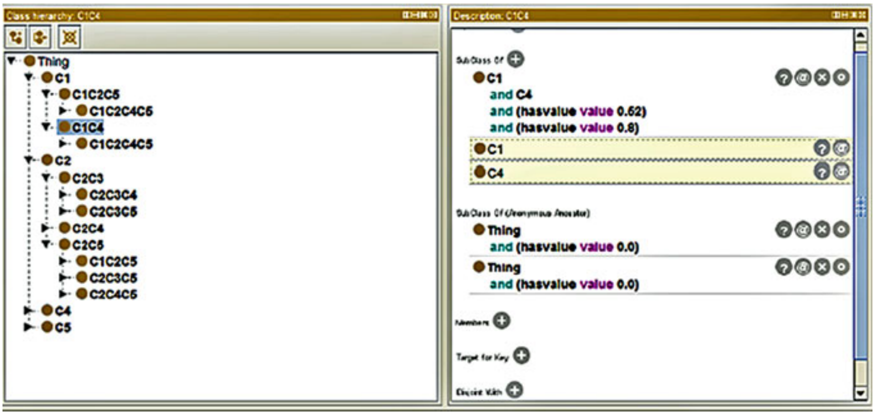
We consider nodes as concepts. The name of the concept as linguistics label. Nevertheless, taxonomic relationships between concepts are presented in the lattice. These classes visualization is done in Asserted class hierarchy of Protégé and the view is offered by OWLViz Plugin.

5.6 *Mapping Ontology to Queries*

Next step is to provide means for transforming concept lattice based ontology expression to associations rules. This process enables to produce logical expression of ontology lattice and specify intended semantics of the descriptions in first order logic. Once the ontology are defined, thus we can model the resulted rules deduced from our Fuzzy Ontology using Protege 4.2 software as bellow (Fig. 9):

In order to define non-taxonomic relationships the following groups of rules are defined:

- **Properties of concepts:** For properties of concepts definition, the following predicate can be used: has property (Concept name, Property name).
- **Inheritance of properties:** Inheritance of properties can be represented by the following rule: has property (C1, X) ← is a (C1,C2), has property (C2, X).



C1
and C4
and (hasvalue value 0.62)
and (hasvalue value 0.8)

C1

C4

Thing
and (hasvalue value 0.0)

Thing
and (hasvalue value 0.0)

Fig. 9 Fuzzy association rules

- **Ontological relationships:** Like part-of, related-to, etc., can be easily represented via predicates. The following predicates demonstrate opportunities adding other ontological relationships: part of (C1, C2), related (C1, C2), synonyms (C1, C2), etc.

6 Presentation of FODM-FQ: FODM to Flexible Query

The architecture of the FODM-FQ is detailed in Fig. 10.

In this section, a new method to support database flexible querying using the generated knowledge deduced from the FODM model is defined.

It has three steps: the first and the second step performs the data organization and Fuzzy ontology generation described on the Sect. 5. We focus now on step3.

6.1 *Flexible Query Algorithm*

A flexible and cooperative database flexible querying approach within the fuzzy ontology framework has been proposed. This approach takes into account the semantic dependencies between the query and the search criteria to determine its realizability or not. Thus the idea is to change the level of granularity and apply the clustering operation, so the interrogation will focus necessarily on clusters.

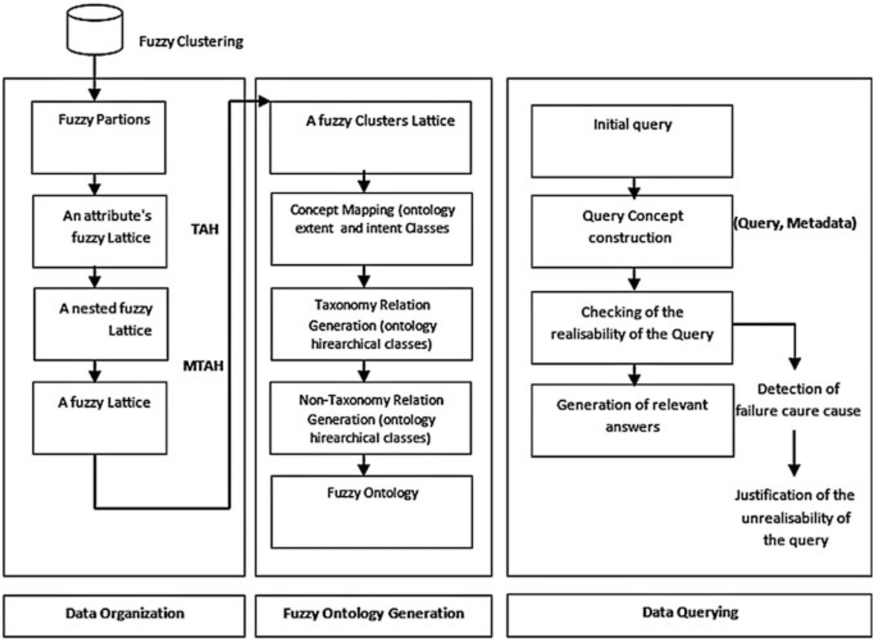

Fuzzy Clustering
Fuzzy Partions
An attribute's fuzzy Lattice
TAH
A nested fuzzy Lattice
MTAH
A fuzzy Lattice
A fuzzy Clusters Lattice
Concept Mapping (ontology extent and intent Classes
Taxonomy Relation Generation (ontology hirearchical classes)
Non-Taxonomy Relation Generation (ontology hirearchical classes)
Fuzzy Ontology
Initial query
Query Concept construction
(Query, Metadata)
Checking of the realisability of the Query
Generation of relevant answers
Detection of failure caure cause
Justification of the unrealisability of the query
Data Organization
Fuzzy Ontology Generation
Data Querying

Fig. 10 Presentation of the new approach: FODM to flexible query

Next step presents our flexible query algorithm using the generated ontology in the second step. Let R the user Query. The pseudo-code for flexible query generation algorithm is given:

```
Algorithm 2: Flexible query generation
Input: The user Query R
Output: List of answers
begin
    Concept_Query(R,$\Phi_B$);
    let i =Cardinality($\Phi_B$);
    Examine the rules in level i;
    if there is a rule which uses all the elements of $\Phi$ then
        R is realizable with CF = 100%;
        Extract(R, i);
    else if there is a rule which uses at least one elements of the $\Phi_B$
    then
        R is realizable with CF< 100%;
        Extract(R, i);
    else
        R is not realizable;
    end
end
```

Note that:

- **Concept_Query (R, Φ_B):** is a procedure that determine the concept Φ_B of R.
- **Extract (R, i):**is a procedure that determines answers of the request while using the Backward chaining. This procedure calls upon all the rules closely related to the request of level ≤ i

Applying this algorithm, the generated knowledge is in the form of rules. We obtain a query concept $\Phi = (\Phi_A, \Phi_B)$.

6.2 Construction of the Query Concept

We define a query concept $\Phi = (\Phi_A, \Phi_B)$ where Φ_A is a name to indicate a required extension and Φ_B is the set of classes describing the data reached by the query.

The set of classes Φ_B is determined by the following procedure:

```
Input: Vector V(A)=vj : j=1,...,C(A) of $\Phi$ cluster centres of relievable
       attribute A and the value of associated to this last.
Output: Query concept $\Phi$=($\Phi_A$,$\Phi_B$)
begin
    Step 1: Calculate the membership degrees of the specified clusters
    for each value of the criterion of $\Phi$ associated to the relievable
    attribute A;
    Step 2: Apply $\alpha$-Cut to generate the fuzzy context;
    Step 3: Form the set $\Phi_B$ of clusters whose membership is higher
    than the $\alpha$-Cut value;
end
```

Example For better explaining this step, we consider a relational database table describing apartment announces. The query is as follows:

$$Q\begin{cases}\textbf{Select} & \mathit{ref\,An}, & \mathit{price}, & \mathit{surface}\\ \textbf{From} & \mathit{Announce}, & \mathit{Appartment} & \\ \textbf{Where} & \mathit{price} = & 105 & \\ (A1) & & & \\ \textbf{and} & \mathit{surface} = & 75 & \\ (A2) & & & \\ \textbf{and} & \mathit{city} = & '\mathit{Paris}' & \end{cases} \tag{1}$$

In this query, the user wishes that its preferences be considered according to the descending order: Price and Surface. In other words, returned data must be ordered and presented to the user according to these preferences. Without this flexibility, the user must refine these search keys until obtaining satisfaction if required since he does

Table 6 Query memberships degrees

Price			Surface	
C1	C2	C3	C4	C5
0.1	0.3	0.6	0.2	0.8

Table 7 Query memberships degrees with α-Cut

Price			Surface	
C1	C2	C3	C4	C5
–	0.3	0.6	–	0.8

not have precise knowledge on the data which he consults. According to the criteria of the query ϕ only the **A1** and **A2** criteria correspond to relievable attributes.

Initially, we determine starting from the DB the tuples satisfying the non relievable criteria (A_3, A_4, A_5), result of the following query:

$$Q\left\{\begin{array}{llll} \textbf{Select} & refAn, & price, & surface \\ \textbf{From} & Announce, & Appartment & \\ \textbf{Where} & city\ = & 'Paris' & \\ (A3) & & & \\ \textbf{and} & place\ = & '16^{eme}arrondissement' & \\ (A4) & & & \end{array}\right. \tag{2}$$

These tuples is broken up into clusters according to labels of the relievable attributes *Price* and *Surface*. The query concept are given with part of the fuzzy clustering operation to determine the objects membership's degrees in the various clusters. Table 6 present the membership degrees associated to the query. These degrees are obtained while basing on memberships matrix obtained by a fuzzy clustering algorithm.

Then, we apply the α-Cut for each attribute to minimize the number of concepts. We obtain the reduced context request presented by the Table 7.

According to our example, the query Q seek the data sources having the metadata $Q\{C2, C3, C5\}$.

6.3 Checking of the Query Realisability

If the query criteria are in contradiction with their dependences extracted from the database, it is known as unrealisable.

Proposition *Let a query Q having the concept* $\Phi = (\Phi_A,\Phi_B)$. *A query* Φ_A *is unrealisable if and only if there is no data source in* Φ_A *which divide any metadata of the set* Φ_A.

This proposition of relevance is at the base of the research process. It is different from the vicinity concept used in [16], which can lead to obtain data that don't share any metadata with the initial query and don't correspond to the end-user needs.

Table 8 Characteristics of datasets

Data Set	Nb of objects	Size of objects	Nb of items
C20d10 K	10,000	20	386
T25i10d10 K	1,000	25	1,000
Mushrooms	8,416	23	128
Car	1,728	7	26
Achat	28	5	5

7 Evaluation of the Proposed Approach

The performance of the proposed algorithm for Discovering Fuzzy queries can be measured in order to evaluate the generated ontology. To do this, we compare two approaches using 4 datasets known on the ECD field (Table 8).

The first approach does not apply the clustering concept and the second uses the formal concepts for structuring and building ontology-based classification with AFC adopted by "ClusterFCA". ClusterFCA is a java platform developed by our team. It includes a classification module containing algorithms for binary and fuzzy clustering. It also includes an AFC module for the construction of simple and nested lattice.

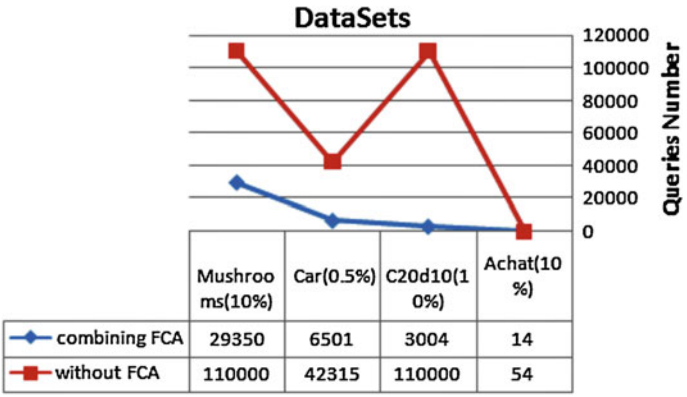
In this chart we show the number of rules resulting from these data sets: Mushrooms (8,416 objects), C20d10 K (10,000 objects), Car (1,728 objects), Achat (28 objects).

The existing algorithms dont take into account any semantics of the data. All the researchers focused themselves on the reduction of the set of rules, by proposing the concept of metadata, or on the method of visualization of this rules. Our main contribution resides in extracting the ontology from datasets by using FCA and transforming it to a rule language in order to model the expression of the user's preferences and generate the relevant answers. Thus, we prove in Fig. 11 that with FCA, we minimize the space complexity of the resulting lattice. The combination of two concepts: FCA and ontology models a certain abstraction of the data that is fundamental in the case of an enormous number because the defined ontology is deduced from clusters not from the initial objects. The flexible query approach proposed the followings contributions compared to the similar approaches

- The automatic generation of TAH's and MTAH from relieving attributes.
- The research of relevant data sources for a given query.
- A detection of the query unrealisability.
- The scheduling of the results.

Different advantages are granted by the proposed approach. This approach is:

- More reliable compared to the classic one (without clustering). In the examples, the number of classes generated in the case of the application of our new approach is less than the number of classes of input ontology. The decrease



	Mushrooms(10%)	Car(0.5%)	C20d10(10%)	Achat(10%)
combining FCA	29350	6501	3004	14
without FCA	110000	42315	110000	54

Fig. 11 Metrics of the proposed approach

mainly depends on the number of clusters that choice, leads to a considerable reduction in the number classes composed the ontology. It retrieve pertinent informations which still more meaningful to the end-user and allows him to easily exploit all knowledge generated. The decrease of association rules mainly depends on the number of chosen clusters.

- Applicable to any type/amount of data: As part of our support system to build an ontology using the AFC, the experts in each field could reach a mass of acceptable information. Indeed, formal Concept Analysis helped us to structure and then build ontologies because the FCA was able to express ourselves in an fuzzy ontology format. The lattice is easy for people to understand and can be used as a guideline for the construction of ontologies. We gave also an example to show the power of the formal concept analysis.
- Applicable with any fuzzy classification algorithm to classify the initial data.

8 Conclusion

This paper focuses on how the future retrieval information from large dataset might look like and the interaction between the sources of information to yield perfect and real time results with unique power of intelligence is done to interpret the best possible solution for the user query.

The main idea is to collaborate the search to be more informative and provide it an intelligence in order to retrieve user oriented results. The model defined for this approach is FODM-FQ. It consists of four steps: The first organizes the database records in homogeneous clusters having common properties to deduce the data's semantic. This step consists of TAH's and MTAH generation of relieving attributes. The second step called Discovering Knowledge is used to deduce the Fuzzy Cluster Lattice corresponding to MTAH lattice generated in the first step. Then, on the third step, the FCL is mapped to an owl ontology design. From this ontology, the rules modeling the Knowledge (Set of Fuzzy Associations Rules on the attributes SFR) are extracted. We prove that the discovered rules does not contain any redundant rule. The fourth step ensures database flexible querying using the generated ontology.

An example of an ontology was simulated using Protege and results were analyzed. The keywords entered by the user were given priority and bases on that we have discarded certain resulted query using FCA methodology which make the search more compact and effective. The future scope of this method is to first integrate the current into large domains resulting in expansion of knowledge base. Secondly, an intelligent distributed ontology query processing method will be proposed to deal with the growth of the data size and the number of distributed queries which access the common part of the resources and successfully meet the user preferences.

References

1. Arpirez, J., Corcho, O., Fernandez-Lopez, M., Gomez-Perez, M.: WebODE: a workbench for ontological engineering. In: First International Conference on Knowledge Capture (K-CAP01), Victoria, pp. 613 (2001)
2. Azar, A.T.: Fuzzy Systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
3. Azar, A.T.: Overview of type-2 fuzzy logic systems. Int. J. Fuzzy Syst. Appl. (IJFSA) **2**(4), 1–28 (2012)
4. Azar, A.T.: Adaptive neuro-fuzzy systems. In: Azar, A.T. (ed.) Fuzzy Systems. IN-TECH, Vienna, ISBN 978-953-7619-92-3 (2010a)
5. Baer, P.G.D., Kapetanios, E., Keuser, S.: A Semantics Based Interactive Query Formulation Technique, UserInterfaces to Data Intensive Systems. Second International Workshop on User Interfaces to Data IntensiveSystems, Zurich, 43–49 (2001)
6. Bao-xiang, X., Zhang, Y.: Research on the development of information system modeling theory. J. Intell. **29**(5), 70–74 (2010)
7. Bechhofer, S., Horrocks, I., Goble, C., Stevens, R.: OilEd-a reason-able ontology editor for the semantic web. In: Joint German-Austrian Conference on Artificial Intelligence (KI01), Vienne, pp. 396–408 (2001)
8. Berners-Lee, T.: Weaving the Web. HarperCollins, New York, ISBN 006-251-5861 (1999)
9. Bobillo, F., Straccia, U.: Fuzzy description logics with general t-norms and datatypes. Fuzzy Sets Syst. **160**(23), 3382–3402 (2009)
10. Borzsonyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: International Conference on Data Engineering (ICDE), Heidelberg (2001)
11. Bosc, P., Galibourg, M., Hamon, G.: Fussy quering with SQL: extensions and implementation aspects. Fussy Sets Syst. **28**(3), 333–349 (1988)
12. Bosc, P., Liétard, L.: Aggregates computed over fuzzy sets and their integration into SQLf. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **16**(6), 761–792 (2008)
13. Bosc, P., Lietard, L., Pivert, O.: Databases and flexibility: gradual queries. TSI-Technique et Science Informati ques-RAIRO **17**(3), 355–378 (1998)
14. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. Comput. J. IEEE Trans. Fuzzy Syst. **3**(1), 1–17 (1995)

15. Bosc, P., Pivert, O.: A propos de requetes a preferences et diviseur stratifie. 28me Congrs INFORSID, France, pp. 311–326 (2010)
16. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. **19**(1), 1–27 (2001)
17. Carroll, J., Klyne, G.: RDF concepts and abstract syntax. Recommendation, W3C (2004), http://w3c.org/TR/rdf-concepts
18. Chandrasekaran, B., Josephson, J., Benjamin, V.: What are ontologies, and why do we need them? IEEE Intell. Syst. **14**(1), 20–26 (1999)
19. Chang, C.: Decision support in an imperfect world. In: Trends and Applications on Automating Intelligent Behavior-Applications and Frontiers, Denmark, p. 25 (1983)
20. Chomicki, J.: Preference formulas in relational queries. ACM Trans. Database Syst. **28**(4), 427–466 (2003)
21. Chu, W., Yang, H., Minock, M., Chow, G., Larson, C.: CoBase-a scalable and extensible cooperative information system. J. Intell. Inf. Syst. **6**(2–3), 223–259 (1996)
22. Clerkin, P., Cunningham, P., Hayes, C.: Ontology discovery for the semantic web using hierarchical clustering. In: European Conference Machine Learning (ECML) and European Conference Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD-2001) (2001)
23. Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: The semantic web-the roles of XML and RDF. IEEE Internet Comput. **5**(4), 63–74 (2000)
24. Ding, Y., Foo, S.: Ontology research and development: Part 1 a review of ontology generation. J. Inf. Sci. **28**(2), 123–136 (2002)
25. Domingue, J., Motta, E.: Knowledge modeling in webonto and ocml. http://kmi.open.ac.uk/projects/ocml (1999)
26. Domshlak, C., Hoos, H., Boutilier, C., Brafman, R., Poole, D.: Cp-nets : a tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res. **21**, 135–191 (2004)
27. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Proceedings of the 5th International Conference on Flexible Query Answering Systems (FQAS) 02, London, UK, pp. 174–182 (2002)
28. Farquhar, A., Fikes, R., Rice, J.: The ontolingua server: a tool for collaborative ontology construction. In: The 10th Knowledge Aqcuisition for Knowledge-Based Systems (KAW96), Canada, pp. 174–182 (1996)
29. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations, vol. 1640. Springer, Heidelberg (1999)
30. Grissa Touzi, A., Sassi, M., Ounelli, H.: An innovative contribution to flexible query through the fusion of conceptual clustering, fuzzy logic, and formal concept analysis. Int. J. Comput. Appl. 16(4), 220–233 (2009)
31. Kapetanios, E., Baer, D., Glaus, B., Groenewoud, P.: MDDQL-Stat: data querying and analysis through integration of intentional and extensional semantics. Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004), Switzerland, pp. 353 (2004)
32. Kiessling, W.: Data querying and analysis through integration of intentional and extensional semantics. In: Foundations of Preferences in Database Systems. Very Large Data Base (VLDB) Endowment Inc, pp. 311–322 (2002)
33. Lacroix, M., Lavency, P.: Preferences-putting more knowledge into queries. In: Proceedings of the 13th International Conference on Very Large Data Bases, University of Vienna, Austria, pp. 217–225 (1987)
34. Lassila, O., Swick, R.: Resource description framework (RDF) model and syntax specification. Recommendation, W3C (1999)
35. Lietard, L., Rocacher, D.: On the definition of extended norms and co-norms to aggregate fuzzy bipolar conditions. In: The European Society of Fuzzy Logic and Technology Conference, pp. 513–518 (2009)

36. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.: OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. J. Distrib. Parallel Databases **8**(2), 223–271 (2000)
37. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 web ontology language: profiles. Recommendation, W3C. http://www.w3.org/TR/owl2-profiles/ (2009)
38. Motro, A.: VAGUE-A user interface to relational databases that permits vague queries. ACM Trans. Office Inf. Syst. **6**(3), 187–214 (1988)
39. Noy, N., Fergerson, R., Musen, M., IENG, R. D., CORBY, O. The knowledge model of protg2000 : combining interoperability and flexibility. In: 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW00), Juan-les-Pins, France, pp. 17–32 (2000)
40. Ounalli, H., Belhadj, R.: Interrogation flexible et coopérative d'une BD par abstraction conceptuelle hirarchique, pp. 41–56. INFORSID, Biarritz, France (2004)
41. Paton, N.W., Stevens, R., Baker, P., Goble, C.A., Bechhofer, S., Brass, A.: Query Processing in the TAMBIS Bioinformatics Source Integration System. Proceedings of the IEEE International Conference on Scientific and Statistical Databases (SSDBM), 138–147 (1999)
42. Pivert, O.: Contribution a l'interrogation flexible de bases de donnees: expression et evaluation de requetes floues. PhD thesis (1991)
43. Quan Thanh, T., Hui, S.C., Fong, A., Cao, T.H.: Automatic fuzzy ontology generation for semantic web. IEEE Trans. Knowl. Data Eng. **18**(6), 842–856 (2006)
44. Rabitti, F., Savino, P.: Retrieval of multimedia documents by imprecise query specification. In: Advances in Database Technology-EDBT90, pp. 203–218. Springer, Berlin (1990)
45. Sassi, M., Grissa Touzi, A., Ounelli, H.: Clustering quality evaluation based on fuzzy FCA. In: 18th International Conference on Database and Expert Systems Applications, (DEXA07), Regensburg, Germany, pp. 62–72. LNCS (2007)
46. Soergel, D.: Some remarks on information languages, their analysis and comparison. Inf. Storage Retrieval **3**(4), 219–291 (1967)
47. Spoerri, A.: InfoCrystal: a visual tool for information retrieval management. In: Second International Conference on Information and Knowledge Management, Washington, pp. 11–20 (1993)
48. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: collaborative ontology engineering for the semantic web. In: First International Semantic Web Conference (ISWC02) of Lecture Notes in Computer Science, Chia, Sardaigne, Italie, vol. 2342, pp. 221–235 (2002)
49. Tahani, V.: A conceptual framework for fuzzy query processing: a step toward very intelligent database systems. Inf. Process. Manage. **13**(5), 289–303 (1977)
50. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query graph candidates for efficient keyword search on rdf. In: IEEE Computer Society (ed.) Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 405–416. IEEE Computer Society (2009)
51. Uri, K., Jianjun, Z.: Fuzzy clustering principles, methods and examples, vol. 17(3), p. 13. Technical Report, Technical University of Denmark, IKS, Denmark, (1998)
52. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, vol. 83. Springer, Berlin (1982)
53. Wuermli, O., Wrobel, A.C. a. H. & Joller, J. (2003). Data mining for ontology building: semantic web overview. PhD thesis, Nanyang Technological University
54. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)

An Efficient Multi Level Thresholding Method for Image Segmentation Based on the Hybridization of Modified PSO and Otsu's Method

Fayçal Hamdaoui, Anis Sakly and Abdellatif Mtibaa

Abstract In the area of image processing, segmentation of an image into multiple regions is very important for classification and recognition steps. It has been widely used in many application fields such as medical image analysis to characterize and detect anatomical structures, robotics features extraction for mobile robot localization and detection and map procession for lines and legends finding. Many techniques have been developed in the field of image segmentation. Methods based on intelligent techniques are the most used such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO) called metaheuristics algorithms. In this paper, we describe a novel method for segmentation of images based on one of the most popular and efficient metaheuristic algorithm called Particle Swarm optimization (PSO) for determining multilevel threshold for a given image. The proposed method takes advantage of the characteristics of the particle swarm optimization and improves the objective function value to updating the velocity and the position of particles. This method is compared to the basic PSO method, also, it is compared with other known multilevel segmentation methods to demonstrate its efficiency. Experimental results show that this method can reliably segment and give threshold values than other methods considering different measures.

F. Hamdaoui (✉)
Laboratory of EμE, Faculty of Sciences of Monastir (FSM),
University of Monastir, Monastir, Tunisia
e-mail: faycel_hamdaoui@yahoo.fr

A. Sakly
Industrial Systems Study and Renewable Energy (ESIER),
National Engineering School of Monastir (ENIM), Electrical Department,
University of Monastir, Monastir, Tunisia
e-mail: sakly_anis@yahoo.fr

A. Mtibaa
Laboratory of EμE, Faculty of Sciences of Monastir (FSM),
National Engineering School of Monastir (ENIM), Electrical Department,
University of Monastir, Monastir, Tunisia
e-mail: abdellatif.mtibaa@enim.rnu.tn

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_14

Keywords Multilevel segmentation · Particle swarm optimization · Image processing

1 Introduction

Segmentation is to divide an image into region corresponding to objects. These objects include more information than pixels. Indeed, interpretation of images based on objects gives more illustrative and meaningful information's than information gives based on individual pixel interpretation. The principle aim of image segmentation is to apply a specific treatment or to interpret the image content.

If human knows naturally separate objects in an image due to high-level knowledge that consist on understanding of the objects and the scene. Developing segmentation algorithms still one of the most topics research common in the field of image processing. So far, there are many segmentation methods that can be classified into four main types including region based segmentation like region-growing [1, 2] and region based split and merging [3, 4], edge-based segmentation [5, 6], histogram thresholding based method [7, 8] and Segmentation based on hybridization between two of the first three segmentations.

Thresholding method is one of the most common methods for the segmentation of images into two or more clusters [9, 10]. It is a simple and popular method for digital image processing that can be divided into three different types: global thresholding methods [11, 12], local thresholding methods [13, 14] and optimal thresholding methods [15, 16]. In the former, global thresholding methods are used to determinate a threshold for the entire image, it concern only the binarization and the result after segmentation is a binary image. Local thresholding methods are fast and for this reason they are suitable for the case of multilevel thresholding. However the major drawback of this method is the determination of the number of thresholds. The advantage of the optimal thresholding methods is the objective function. Indeed, the determining of the best threshold values amounts to optimize the objective function.

Several algorithms have been widely proposed in the literature for the bi-level [17–21] and also for the multi-level thresholding problem. For two level thresholding, solving the problem is same as finding the threshold value called T which satisfies this condition: pixels which are lower than T represent the object and the other pixels the background. This problem could be extended to n-level thresholding when distinct objects are depicted within a given scene, multiple threshold values called $T_1, T_2, \ldots, T_i$ with $i \geq 2$ have to be determinate. A variety of multilevel thresholding approaches have been proposed for image segmentation including the Otsu criterion [22–24]. This method is simple but it has a disadvantage that it is computationally expensive. To overcome this problem, many papers have been published in the literature designed especially for computation acceleration of the specific objective function [25, 26]. Among this category, we find methods that utilize the meta-heuristic optimization methods namely GA [27–29], ABC [30–32], PSO [33–35]. Meta-heuristic algorithms have been inspired from nature and they are used to solve difficult optimization problems. Those optimization algorithms could be used to solve many complex optimization problems, which are non-linear, non-differentiable and multi-model.

In this paper, a proposed algorithm for image segmentation based on PSO is used to automatically determinate the threshold values in multilevel thresholding problem. The PSO algorithm is based on swarm behavior of birds where particles (in our case: pixels), fly through the search space using two simples equations for velocity and position. This algorithm has been widely used to solve global and local optimization problems. This algorithm has undergone several evolutions, we cite the Darwinian Particle Swarm Optimization (DPSO) [36] and the fractional calculus based PSO called FODPSO [37]. This work mainly focuses on using a new variant of PSO denoted as MMPSO (Multithresholding based on Modified PSO) and it is the first time to verify and apply this method to multilevel segmentation. We start by presenting a brief review on the most known meta-heuristic algorithms. Next in Sect. 2, we introduce the n-level thresholding problem formulation. Section 3 presents a brief review of the traditional PSO and the MMPSO. In Sect. 4, benchmarks images used to demonstrate the advantages given by the proposed method in comparison with other commonly used algorithm such as genetic algorithm (GA) and traditional PSO. Finally, Sect. 5 outlines conclusions.

2 Related Work Based Meta-Heurestic Algorithm

2.1 Genetic Algorithm

Genetic algorithms (GAs) are meta-heuristics search methods belonging to the class of evolutionary algorithms (EAs). They are inspired by the analogy between the optimization process and the evolution of organisms. A GA is used to search for global or optimal solutions when no deterministic method exists or when the deterministic method is computationally complex. GA is a population based algorithm that was proposed by John Holland in 1975 [38]. Then by Goldberg in 1989 [39], Holland in 1992 [40], Man et al. in 1996 [41], Schmitt and Petrowski in 2001 [42, 43]. Later, this technique has been used in many fields such as image segmentation, which has been transformed into an optimization problem like in [44–47].

Multithresholding amounts to finding more than a threshold, this means several solutions. Each solution is represented as chromosome, each chromosome is constructed from genes and solutions generated per iteration are called population. The size of the population is the number of solutions per iteration. Let n the population size of randomly generated individuals. The genetic algorithm starts with n random solution. Then the best member solutions are selected to generate new solutions, so

Table 1 Pseudo-code of the GA algorithm

```
1. Initialize a population of random individual solutions
2. While stopping criterion not met, do:
    2-1. Create a new population
    2-2. Until creation of the entire population, do:
        i. Selecta pair of individuals that has the lowest value of fitness
        ii. Cross-over the two individuals to produce two new individuals
    2-3. Randomize each individual of the new population to mutate
    2-4. Replace the population with the new population
3. Display the best solution found over the search
```

the best generated solutions will be added to the next iteration and all bad solutions will be rejected. The selection of the best solution in each generation is based best fitness evaluation values of every individual in the population to form a new population. The stopping criterion is determinate by the number of generations that has been produced, or on a satisfactory fitness value that has been reached for the population. Generally, the genetic algorithm is based on four steps: population initialization, evaluation of fitness, reproduction and termination criterion (Table 1).

2.2 Ant Colony Optimization (ACO)

The ant colony algorithms are a family of meta-heuristics inspired from nature using swarm intelligence. The behavior of ants in their search for food has been studied and applied to solve complex optimization problems. Simply, ants initially start by moving randomly. Once the food has been found, again they join their colony by filing in the way a chemical substance called pheromone [48]. Other ants that are experiencing the same path have a high probability to stop their random movements and follow the path marked by the substance (pheromone): this was called the phenomenon of stigmergic optimization [49]. After some research, there will be several paths that lead to food. The shortest path will be covered without necessarily having a global vision of the path [50, 51]: this phenomenon is based positive feedback. Therefore, the long paths ultimately disappear. Finally, all the ants will follow the shortest path.

Dorigo [52] and Colorni [53] are the first who have trying to implement an inspired ACO analogy to solve the problem of searching for an optimal path in a graph. Then, several problems have emerged and drawing on various aspects based behavior of ants. Multithresholding is among the areas in which the ACO algorithm was implemented to obtain the optimal thresholds in the field of image segmentation [54–57].

Table 2 below gives the overall description of the ACO algorithm:

Table 2 Pseudo-code of the ACO algorithm

```
1. Set parameters and initialize pheromone trails
2. for i from 1 to number_of_iterations do
for j from 1 to population_size do
Building solutions based on the probability of state transition
Stop when all ants have been generated
3. Evaluate all solutions and select the best one even iteration
4. Apply the pheromone update rule
5. Continue until reaching the stopping criterion
```

2.3 *Artificial Bee Colony (ABC)*

Artificial Bee Colony algorithm was firstly proposed by Karaboga [58] in 2005 for searching numerical optimization problems based on intelligent foraging behavior of honey bee swarm. Later, further improvements have been carried out for the ABC algorithm by Karaboga and Basturk in 2006 and 2008 [59, 60]. Colony of ABC model consists of three groups of bees: employed bees, onlookers and scouts [61].

The bee which discovered and a source of food to exploit belong to the employed bees group. The second groups of bees called onlookers are those waiting in the hive for information about the sources of food from the employed bees. The third group of scouts' bees is the set of bees which will randomly search for the food sources around the hive. After exploiting a source of food, a bee belonging to the employed bees group returns to the hive and shares information about the nectar amount produced in the food source with other bees. The employed bee starts dancing in the dance area of the hive. Communication among bees related to the quality of food sources takes place in the dancing area. This dance is called a waggle dance and it is made to share information with a probability proportional to the profitability of the food source. More profitable the source is, more the dancing duration is so longer. An onlooker on the dance floor watches numerous dances and selects to employ herself at the most profitable source. After watching several dances, an onlooker bee chooses a source of food and becomes employed bee. In a similar way, a scout is called employed when it finds a source of food. After completely exploiting a source of food, all employed bees abandoned it and change into onlookers or scouts [62, 63]. Typically algorithm for the ABC algorithm is given in Table 3.

The ABC algorithm finds optimal solutions for the optimization problems. Many researchers use this algorithm to determine the threshold values for the multilevel thresholding problem [64–68].

Table 3 Pseudo-code of the ABC algorithm

1. **Initialize** the population of solution and generate food sources
2. **Assign the employed** bees on their food sources
3. **Assign onlooker** bees to the food sources depending on their amount of nectar
4. The scout bees **randomly research** the neighbor area to discover new food sources
5. The best solution is **recorded and increases** the cycle by 1
6. **The algorithm is end** if the cycle is equal to the maximum cycle number (max cycle), otherwise go to Step 2

2.4 Shuffled Frog Leaping Algorithm (SFLA)

SFLA is a recent meta-heuristic algorithm proposed by Eusuff and Lansey in 2003 [69] that mimics the principle of a group of frogs evolution that searches discrete locations containing as much food as available. SFLA combines the advantages of the local search tool of the PSO algorithm and the idea of mixing information from parallel local searches to move toward a global solution [70]. The SFLA algorithm has been tested on several combinatorial problems and has demonstrated effectiveness [71] in various global solutions [72, 73].

In general case, when we apply the SFLA algorithm to found an optimum solution, each frog has a different solution from others frogs. This solution is determined according to the fitness function and its adaptability. SFLA algorithm involves a population defined by a set of frogs (solutions). The entire population is partitioned into a predefined number of subsets referred to as memplexes. Those mememplexes are considered as different crops of frogs to performing a local search. Frogs of each memplex have their own strategy to explore the environment in different directions. After a predefined number of memetic evolution, the exchange of information between memplexes takes place in a procedure of shuffling [74]. This procedure must ensure that the evolution toward a particular interval is free from all prejudices. Memetic evolution and shuffling are performed alternatively until reaching the convergence criterion or otherwise until a stopping criterion. Steps of SFLA are given below [12].

Step 1: Initial population of F frogs, in which individual frogs are equivalent to the GA chromosomes, is created randomly.

Step 2: All frogs are sorted in descending order based on their fitness values and divided into m memplexes, each memplex containing p frogs; the frog that is placed first moves to the first memplex, the second one moves to the second memplex, the pth one to the pth memplex, and the (p + 1)th returns to the first memplex, etc.

Step 3: Within each memplex, the frogs having the best and the worst fitness are identified. The frog with the best fitness in the whole population is identified. During the evolution of memplexes, worst frogs jump to reach the best ones.

Table 4 Pseudo-code of the SFLA algorithm

```
Begin;
Generate random population ofPsolutions (individuals);
For each individual frog in the population: calculate fitness (i);
Sort the whole population P in descending order of their fitness;
Divide the population P into m memeplexes;
For each memeplex;
Determine the best and worst individuals;
Improve the worst individual position
Repeat for a specific number of iterations;
End;
Combine the evolved memeplexes;
Sort the population P in descending order of their fitness;
Check if termination = true;
End;
```

Step 4: After a defined number of memplex evolution stages, all frogs of memplexes are collected and sorted in descending order again based on their fitness. Step 2 divides frogs into different memplexes again, and then step 3 is achieved.

Step 5: If a predefined solution or a fixed iteration number is reached, the algorithm stops.

Table 4 shows the proposed algorithm based SFLA technique:

The SFLA algorithm had been recently used in determining the optimal thresholding in the field of image segmentation exactly in the identification of the bi-level [75, 76] and multi-level thresholding [77–82].

3 Proposed Approach

3.1 Image Multilevel Thresholding: Optimization Problem

Multilevel thresholding segments images into several distinct regions. Using this process, it is possible to determinate more than one threshold value for a given gray-level image and segments it into certain brightness regions, which correspond to one background and several objects. Let consider a gray-level image that contains N pixels distributed as objects and background. The multilevel threshold selection can be considered as the problem of finding a set $T(l), l = 1, 2, \ldots, L$ of threshold values with L is defined as the intensity level of the image. As a result of thresholding, the original image will be transformed to an image with $L + 1$ levels. If $T(l)$, $l = 1,2, \ldots, L$ are the threshold values with $T(1) < T(2) < T(3), \ldots, < T(L)$ and $f(x, y)$ is the image function which gives the gray-level value of the pixel with coordinates (x, y). The resultant image $F(x, y)$ as explain before, is defined as:

$$F(x,y)\begin{cases} 0, & \text{if} \quad f(x,y) \le T(1) \\ 1, & \text{if} \quad T(1) \le f(x,y) \le T(2) \\ . & . \quad . \\ . & . \quad . \\ L & \text{if} \quad f(x,y) \ge T(L) \end{cases} \tag{1}$$

The problem of multilevel thresholding can be reduced to an optimization problem. The goal becomes to search and found the threshold values that maximize the fitness function ϕ of the gray-level component. This method [22] requires initially a normalization of the histogram h in order to be independent of number of pixels in the image. Considering $N = W \times H$ the total number of pixels included in the image, and assume n_i the number of pixels to a gray-level i included in the range of [0, 255]. So, $h(i)$ is determinates using Eq. (2) below:

$$h(i) = \frac{n_i}{N} \tag{2}$$

And ϕ is defined as:

$$\begin{aligned} \phi &= \max \sigma^2(T) \\ T(1) < T(2) < T(3), \ldots, &< T(L) \end{aligned} \tag{3}$$

With σ^2 is the between-class variance generally defined by:

$$\sigma^2 = P_1\sigma_1^2 + P_2\sigma_2^2 \tag{4}$$

And:

$$\sigma_1^2 = \frac{1}{T}\sum_{i=0}^{T-1} (h(i) - \mu_1)^2;\ \sigma_2^2 = \frac{1}{256 - T}\sum_{i=T}^{255} (h(i) - \mu_2)^2 \tag{5}$$

$$\mu_1 = \frac{1}{T}\sum_{i=0}^{T-1} h(i);\ \mu_2 = \frac{1}{256 - T}\sum_{i=T}^{255} h(i) \tag{6}$$

$$P_1 = \frac{1}{W \times H}\sum_{i=0}^{T-1} h(i);\ P_2 = \frac{1}{W \times H}\sum_{i=T}^{255} h(i) \tag{7}$$

With h is the histogram of this image and N and M are respectively width and height of the image.

The major drawback of this problem is the computational effort that is much larger as the number of threshold levels increase. In the last decade, biologically inspired methods have been used as computationally efficient alternatives to analytical methods to solve optimization problems [83, 84].

3.2 The MMPSO Algorithm

Particle Swarm Optimization (PSO) is a population-based optimization algorithm belonging to the evolutionary computation paradigm [85]. It is proposed by Kennedy [86] to solve problems with continuous variables. It is very suitable to solve complex problem with multiple decision at low cost of computational time. As compared with other evolutionary computation algorithms, PSO has many advantages such as non-use of genetic operation; like crossover and mutation with Genetic Algorithm, PSO has a memory so it can learn from others neighbor or itself; after moving to the new group it has more information from its parents and can find the best threshold value in short time and it requires only mathematical operations which make its implementation very easy and not cost in terms of execution time.

PSO is an efficient algorithm that is based on a population initialized with a random solution called particle. Each particle represents an approximate solution to a complex problem in the search space. This solution is determined based on the collective experiences of the same swarm.

In this section, we present another algorithm based on the introduction of PSO in the Otsu method to overcome this problem of the speed of convergence. Indeed, the Otsu method represents is ineffective when the number of thresholds to be determined increases because it requires a very high execution time. The parameters used in the PSO method are empirically identified such as the population size and the stopping criterion. This algorithm is shown in the block diagram shown in Fig. 1.

In PSO, each particle is characterized by its own position vector and velocity vector. The movement of these vectors in the search space is controlled by the following recursive equations:

$$v_{im} = w * v_{im} + c_1 * rand1() * (p_{im} - x_{im}) + c_2 * rand2() * (p_{gm} - x_{im}) \tag{8}$$

$$x_{im} = x_{im} + v_{im} \tag{9}$$

where x_{im} is the ith position of the particle of the swarm; v_{im} the velocity of this particle; p_{im} the best previous position of the ith particle; p_{gm} is the best position of particle in the swarm; $1 \leq m \leq M$ with M is the search space; *rand1()* and *rand2()* are the two independents random number with uniform distribution in the range (0, 1); c_1 and c_2 are two positives constants of accelerations coefficients called cognitive and social parameter respectively; w is called inertia weight and it is used to control the balance between exploration and search space exploitation.

The PSO algorithm given in the following diagram in Fig. 1 and described in the above paragraph is briefly detailed in this Table 5.

PSO approach is based on the memory and the social interaction among individuals. In the general case, the fitness function allows determining the best position for a particle i to make moves from its current (x_i, t) to the next $(x_i, t + 1)$. Moving process is depends on three stages:

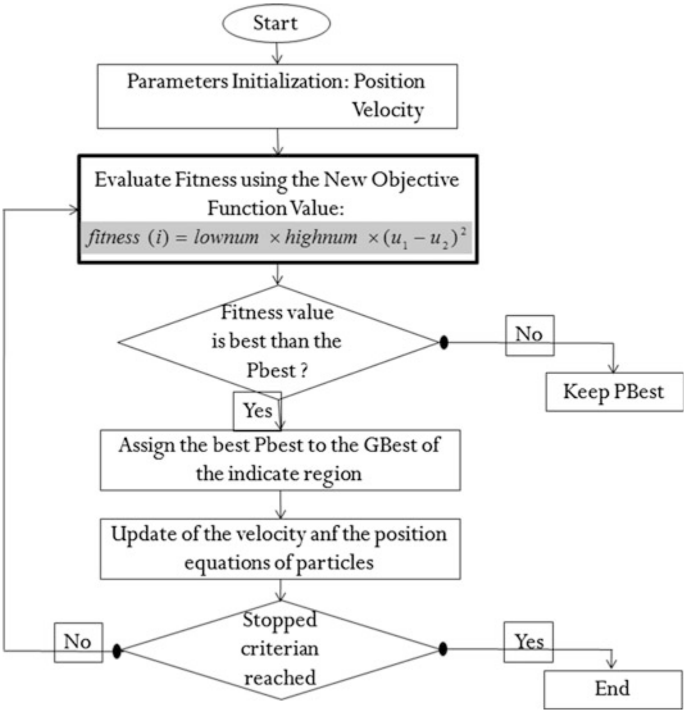

Start
Parameters Initialization: Position
Velocity
Evaluate Fitness using the New Objective Function Value:
$fitness\ (i) = lownum \times highnum \times (u_1 - u_2)^2$
Fitness value is best than the Pbest ?
No
Keep PBest
Yes
Assign the best Pbest to the GBest of the indicate region
Update of the velocity anf the position equations of particles
Stopped criterian reached
No
Yes
End

Fig. 1 Detailed block diagram of the proposed MMPSO method

Table 5 Pseudo-code of the PSO algorithm

```
PSO algorithm
Initialization:
Initialize the position $x_i$ and the velocity $v_i$ of each particle as follows:
$x_{im} = x_{\min} + (x_{\max} - x_{\min}) * rand()$ (10)
$v_{im} = v_{\min} + (v_{\max} - v_{\min}) * rand()$ (11)
do
for $i$ from 1 to $N$
Update:
if $fitness(x_i) \prec fitness(p_i)$ then
up to date of $p_{best}$; $p_i = x_i$
up to date of $g_{best}$
for $i$ from 1 to $M$
up to date of $v_i$ and $x_i$
end
end
end
end
```

1. The current velocity $(v_i, t)c$
2. The best performance (evaluated by the objective function: fitness (f_i, t))
3. The best position of its neighbors (g, t)

Fitness function is used to assess pixels' (particles') for selecting the best individual. It is the most important step that directly affects results of the best position for each individual. So, the overall contribution of this work is to introduce a new fitness function that will give advantages for fast determination of threshold values so that the solutions for the optimal problem. For this, the new introduced algorithm based on the new fitness is called MMPSO and it will be well explained.

Firstly, let calculated a weight P_j for each particle i according to its location using Eq. (12) and then sum SP_i for all particles using Eq. (13):

$$P_j = 2^{p-j} x(i, j) \tag{12}$$

$$SP_i = \sum_{j=0}^{j=p} P_j \tag{13}$$

With p is the number of iterations.

Then, the sum is normalized using Eq. (14)

$$N(i) = \frac{255 \times SP_i}{2^{p-1}} \tag{14}$$

After that, let be introduced four parameters *lowsum lownum highsum highnum* to zero. Then, each particle of the original image with intensity $L(i)$ is compared with $N(i)$ as explained in the pseudo code given in Table 6.

The computation of the new fitness function depends to the final image after segmentation. For each particle i, after comparing all pixels with *N(i)*, two coefficients u_1 and u_2 are calculated according to Eqs. (15) and (16). Finally, the fitness function of the particle i is calculated using Eq. (17).

Table 6 Pseudo-code of the comparison

```
for i from 1 to M
    if $L(i) < N(i)$ then
        $lowsum = lowsum + L(i)$
        $lownum = lownum + 1$
    else
        $highsum = highsum + L(i)$
        $highnum = highnum + 1$
    end
end
```

$$u_1 = \frac{lowsum}{lownum} \tag{15}$$

$$u_2 = \frac{highsum}{highnum} \tag{16}$$

$$fitness(i) = lownum \times highnum \times (u_1 - u_2)^2 \tag{17}$$

This new fitness function increases the probability to use more positions. So, it guarantees and allows the best speed of the convergence to the sought threshold value. This is demonstrated by the following experimental results.

The proposed MMPSO in this work is shown in the Table 7.

4 Experimental Results

MMPSO-based image segmentation which is proposed in this paper was implemented in MATLAB (2011b) on a computer having Intel Core 2 Duo T5800 processor (2.00 GHz) and 3 GB of memory. The proposed methods are tested on a few benchmark images (256 × 256 pixels in size) including: Airplane, Hunter and Map. Figure 2 illustrates all images with their histogram.

The performance of the proposed methods is evaluated by comparing their results with a few popular methods such as GA and PSO. As well, MMPSO is a parameterized algorithm depends on many factors like cognitive, social and inertial weights. They were chosen in reference to several works focusing on the convergence analysis of the traditional PSO [83, 87] to give result in faster convergence CPU time (Table 8).

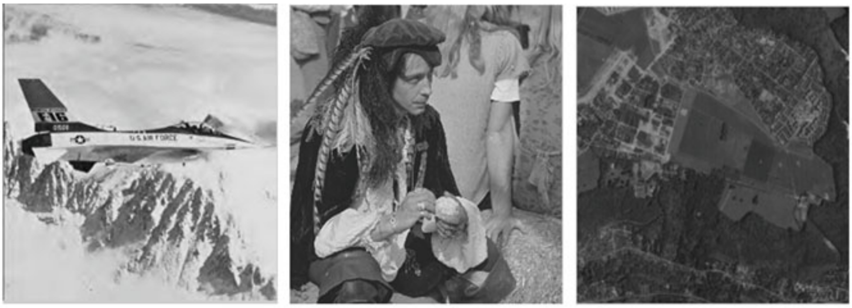
The computational time, threshold values and the fitness evaluation are among the most important indicators which determinate the ability of the algorithm [88]. For many applications, such as medical images, in particular MRI images, data are considerably large in most cases so using a high speed and highly efficient algorithm is preferable. Therefore, the evaluation of the execution time of the CPU process, threshold values and the fitness evaluation seems essentially and necessary to determinate the performance of the new method based MMPSO algorithm.

4.1 CPU Processing Time

CPU processing time is the amount of time for which a metaheuristic method was used for processing and found the best threshold value for each region. It is measured in second. For many values of thresholds (2, 3, 4, and 5) we calculated the CPU processing time for both Particle Swarm Optimization (PSO) method and Otsu's method [22]. Firstly, we start by giving the experimental results of the

Table 7 Pseudo-code of the MMPSO algorithm

```
Final MMPSO code
1. Parameter initialization; including swarm size $M$, inertia weight $w$, cognitive and social parameter the position $c_1$ and $c_2$, maximum and minimum velocity values $V_{\max}$ and $V_{\min}$, number of iterations $N_{iter}$
2. Initialize population of particles with random positions and velocities using the two equations below:
$x_{im} = x_{\min} + (x_{\max} - x_{\min}) * rand()$
$v_{im} = v_{\min} + (v_{\max} - v_{\min}) * rand()$
3. Evaluate the fitness function of each particle:
While the stopping criterion is not met
    for each particle $i$ in the swarm
        $Pi = 2^{p-i} x(i,j)$
        $SP_i = \sum_{j=0}^{j=p} P_j$
        $N(i) = \frac{255 \times SP_i}{2^{p-1}}$
    end
    $lowsum = 0$; $lownum = 0$; $highnum = 0$; $highsum = 0$;
    for $i$ from $1$ to $M$
        if $L(i) < N(i)$ then
            $lowsum = lowsum + L(i)$
            $lownum = lownum + 1$
        else
            $highsum = highsum + L(i)$
            $highnum = highnum + 1$
        end
    end
    if $lownum = 0$ then
        $u_1 = 0$
    else
        $u_1 = \frac{lowsum}{lownum}$
    end
    if $highnum = 0$ then
        $u_2 = 0$
    else
        $u_2 = \frac{highsum}{highnum}$
    $fitness(i) = lownum \times highnum \times (u_1 - u_2)^2$
    end
    4.if $fitness(x_i) \prec fitness(p_i)$ then
        up to date of $p_{best}$;$p_i = x_i$
        up to date of $g_{best}$
    for $i$ from 1 to $N$
        5. up to date of $v_i$ and $x_i$
    end
    end
end
```



Airplane Hunter Map

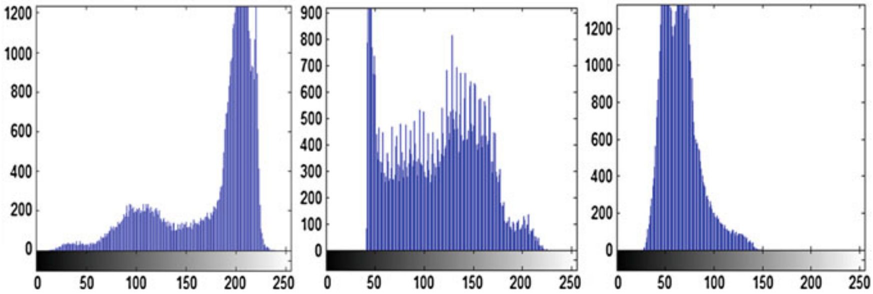


Fig. 2 Benchmark images with their histograms

Table 8 Initial parameters of the PSO and MMPSO

Parameters	PSO	MMPSO
Num of Iterations	8	8
Population	200	200
c_1	1.5	1.5
c_2	1.5	1.5
w	1.2	1.2
$V_{\max}$	2	2
$V_{\min}$	−2	−2
$X_{\max}$	255	255
$X_{\min}$	0	0

average CPU process time for the MMPSO, PSO method and Otsu's method. MMPSO is the hybridization of the PSO and Otsu's method, so the determination of the Otsu's method is in order to compare the performance between the MMPSO and basic Otsu's method without amelioration. PSO is a bio-inspired stochastic algorithm and is like all evolutionary algorithms based random initialization. So, results are not similar in each run. For this, MMPSO and PSO algorithms are executed 20 times. The average CPU process time values are brought in Table 9.

Table 9 The average CPU process time of MMPSO, PSO and Otsu methods

Image	Thresholds	MMPSO (s)	PSO (s)	Otsu’s method
Airplane	2	0.4318	0.4382	0.9942
	3	0.5012	0.4844	2.2472
	4	0.5502	0.5516	37.0792
	5	0.6066	0.6065	40.3196
Hunter	2	0.3814	0.3966	1.1023
	3	0.4766	0.4761	2.0373
	4	0.5516	0.5517	35.2146
	5	0.5913	0.6031	39.7638
Map	2	0.3014	0.3158	0.8578
	3	0.3897	0.3875	2.2944
	4	0.4719	0.4882	37.2169
	5	0.5032	0.5142	40.4277

In refer to Table 9 above, we conclude that the hybridization of the Modified PSO with the Otsu’s method to obtain the MMPSO method is the best idea based on the given results. In fact, the CPU process time given by the PSO is better than the CPU given by the Otsu’s method and when then number of the threshold values number increases the difference becomes more important until it reached 66 times greater than the PSO method. The major advantage is that the reduction in terms of CPU process time continues to improve when the Otsu’s method is combined with the basic PSO algorithm. Another thing to note, whenever the image contains more detail, greater the difference is significant, i.e., the gap CPU process time between PSO and MMPSO and also between Otsu’s method and MMPSO is higher for the map image than the airplane and hunter images.

Now, we will only focus on the metaheuristic methods and we give in Table 10 the CPU process time for the GA method and the Maximum Entropy based Artificial Bee Colony Thresholding method MEABCT [89].

In the literature, it has been proven that the MMPSO requires less CPU processing time to find the threshold values in comparison to GA and MEABCT based ABC metaheuristic algorithm and this is clear in refer to Table 10. It is very clear that MMPSO presents the best processing time compared to all types of metaheuristics. So, it is able to found in less CPU time than MEABCT and of course GA the threshold values.

Note that each time the number of thresholds to be determined increases, the difference between approaches becomes more notable. Given the example of both MMPSO and MEABCT methods; When the number of the threshold values to be determined is equal to *2*, the CPU process time is valuable as 4×10^{-1} ms approximately (to a single number after the comma) for both MMPSO and MEABCT methods, when the number of the threshold values is equal to 3, the CPU process time is valuable as 5×10^{-1} ms for both MMPSO and MEABCT methods, when the number of the threshold values is equal to 4, the CPU process time steel

Table 10 The average CPU process time of metaheuristic methods

Image	Thresholds	MMPSO (s)	GA (s)	MEABCT
Airplane	2	0.4318	0.8113	0.4417
	3	0.5012	0.8642	0.5524
	4	0.5502	0.9189	0.5865
	5	0.6066	0.9862	0.7018
Hunter	2	0.3814	0.7013	0.3570
	3	0.4766	0.7862	0.4798
	4	0.5516	0.9014	0.6016
	5	0.5913	0.9914	0.6859
Map	2	0.3014	0.6821	0.3018
	3	0.3897	0.7532	0.3994
	4	0.4719	0.8852	0.5023
	5	0.5032	0.9365	0.5984

equal to 5×10^{-1} ms for both two methods, and finally when the number of the threshold values to be determined is equal to 5, the CPU process time for the MEABCT method increases and becomes equal 7×10^{-1} ms unlike the MMPSO method where the threshold value becomes only equal to 6×10^{-1} ms.

4.2 Threshold Values

The aim of the proposed algorithm is to determinate the best threshold values for the optimal problem. Since all evolutionary methods among of them MMPSO, PSO, GA and MEABCT are stochastic and random, the results are not completely the same in each run and in each number of threshold. For this, different level for image segmentation are applied and classified in Tables 11 and 12.

Like with the CPU processing time, we begin firstly in this section by giving the results for the MMPSO, PSO and Otsu's method. This, to ensure the performance of the method MMPSO compared with the basic PSO and also basic Otsu methods. Secondly, we compare our MMPSO method with others metaheuristic methods such as GA and MEABCT.

Table 11 shows the selected thresholds of the three test images. It is clear that the selected thresholds by the MMPSO algorithm are very close (for all 2, 3, 4 and 5 threshold problem) to the ones PSO algorithm; nevertheless, there are significant differences of selected thresholds with regard to the Otsu's method. This result reveals that the multithresholding results depend heavily on the objective function that is selected.

Table 12 shows the selected thresholds derived by the MMPSO, GA and MEABCT algorithms for 3, 4, 5 and 6 different levels. We find that the selected thresholds of MMPSO algorithm are equivalent to optimal thresholds derived by the GA and the MEABCT methods when the number of thresholds is less than 4.

Table 11 Average thresholds of MMPSO, PSO and Otsu methods

Image	Th	MMPSO	PSO	Otsu's method
Airplane	2	116, 176	117, 174	114, 172
	3	95, 149, 196	99, 158, 193	86, 132, 202
	4	84, 130, 173, 203	84, 125, 168, 201	69, 117, 162, 188
	5	78, 115, 150, 187, 206	60, 101, 138, 177, 204	67, 113, 152, 200
Hunter	2	54, 118	52, 116	51, 114
	3	42, 88, 144	39, 86, 135	35, 89, 132
	4	36, 88, 133, 159	36, 84, 130, 157	37, 90, 139, 160
	5	41, 84, 128, 159, 184	37, 85, 125, 154, 177	35, 88, 125, 162, 198
Map	2	61, 91	60, 90	56, 84
	3	59, 78, 104	57, 75, 101	56, 72, 96
	4	54, 68, 84, 108	52, 68, 82, 103	45, 62, 74, 94
	5	47, 57, 64,76,98	46, 56, 63, 75, 97	46, 51, 54, 65, 89

Table 12 Average thresholds of metaheuristic algorithms

Image	Th	MMPSO	GA	MEABCT
Airplane	2	116, 176	116, 175	117, 174
	3	95, 149, 196	86, 133, 204	99,157,190
	4	84, 130, 173, 203	71, 119, 164, 200	82, 125, 166, 202
	5	78, 115, 150, 187, 206	84, 124, 164, 188, 204	61, 101, 140, 178, 206
Hunter	2	54, 118	51, 115	52, 115
	3	42, 88, 144	36, 89, 133	39, 87, 136
	4	36, 88, 133, 159	39, 93, 142, 163	33, 85, 132, 159
	5	41, 84, 128, 159, 184	39, 94, 130, 169, 204	37, 86, 126, 156, 178
Map	2	61, 91	56, 85	60, 91
	3	59, 78, 104	57, 74, 97	57, 77, 103
	4	54, 68, 84, 108	49, 66, 78, 100	52, 69, 83, 104
	5	47, 57, 64, 76, 98	44, 55, 60, 72, 95	46, 56, 62, 74, 96

However, when the number of the thresholds exceeds 4, the thresholds of the MMPSO algorithm are different and better than other methods. Also, for the Map image which is more complex, the thresholds of the MMPSO is more significant and the best. Thus, the proposed MMPSO algorithm is suitable for more complex image analysis.

In Fig. 3, it gives results of different segmented images with various threshold levels. Qualitative results given below show that image with higher level of segmentation have more details than others. Along the same lines, Tables 11 and 12 show that for 6-level threshold, the MMPSO method is the method that always has the best threshold value regarding the Otsu's method, PSO, GA and MEABCT methods.

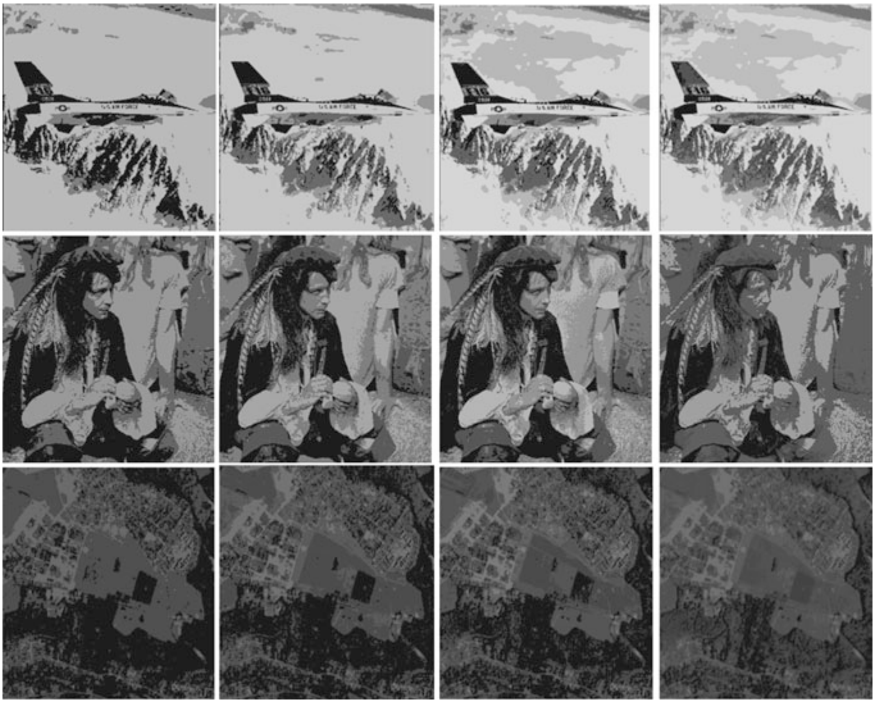


Fig. 3 Results of segmentation with 2, 3, 4, 5 thresholds, respectively from (*left to right*)

4.3 Fitness Evaluation

The objective function value is the main idea of our work. We introduce a new fitness function for the basic PSO algorithm. So, it is very necessary to compare results given by our method called MMPSO to Otsu's method and others meta-heuristics algorithms to determine the best.

We give in Tables 13 and 14 the standard deviation fitness values provided by the MMPSO method, PSO method, Otsu's method, GA method and MEABCT to evaluate the stability of all algorithms. For this, we use the following index in Eq. (18) below [90]:

$$STD = \sqrt{\sum_{i=1}^{n} \frac{(\sigma_i - \mu)^2}{N}} \tag{18}$$

Where STD is the standard deviation, σ_i is the best fitness value of the ith run of the algorithm, μ is the average value of σ and N is the repeated times of each

Table 13 Average STD fitness values of MMPSO, PSO and Otsu's methods

Image	Th	MMPSO	PSO	Otsu's method
Airplane	2	1,837.8596	1,837.6326	1,835.1342
	3	1,911.2235	1,910.2235	1,909.4021
	4	1,955.0526	1,954.8572	1,947.7842
	5	1,978.1365	1,977.8865	1,970.1322
Hunter	2	3,062.9875	3,062.2865	3,062.0051
	3	3,214.5263	3,213.8456	3,213.4752
	4	3,251.5632	3,250.5246	3,240.1785
	5	3,281.2431	3,280.5263	3,253.8542
Map	2	2,255.0895	2,254.8652	2,254.2156
	3	2,526.9856	2,526.1326	2,525.2036
	4	2,619.0256	2,618.2252	2,615.4125
	5	2,666.8564	2,665.1956	2,653.7542

Table 14 Average STD fitness values of metaheuristic algorithms

Image	Th	MMPSO	GA	MEABCT
Airplane	2	1,837.8596	1,837.6288	1,837.8442
	3	1,911.2235	1,844.5265	1,910.8526
	4	1,955.0526	1,950.0562	1,954.3252
	5	1,978.1365	1,972.8562	1,977.9235
Hunter	2	3,062.9875	3,062.1875	3,062.5326
	3	3,214.5263	3,211.5962	3,214.0235
	4	3,251.5632	3,231.0025	3,250.6235
	5	3,281.2431	3,243.2865	3,280.6659
Map	2	2,255.0895	2,252.3364	2,254.9965
	3	2,526.9856	2,503.1420	2,526.1753
	4	2,619.0256	2,617.8852	2,618.5623
	5	2,666.8564	2,658.5243	2,665.8562

method (here N = 20 times). Note that all objective function values are calculated for 2, 3, 4 and 5 thresholds.

Table 13 presents the standard deviation of the fitness values for the MMPSO method, PSO method and Otsu's method. Higher STD means a higher stability of the fitness used for the algorithm. The MMPSO method and PSO method have very close STD values with a slight advantage in favor of the MMPSO method against the PSO method. Both of those methods have better values compared to the Otsu method. Thereby, the fitness value of the MMPSO method more stable objective function.

We can easily note that the difference between all algorithms is very small. Also, we remark that our method is the best method in refer to values given to Table 13. In other terms, whenever increasing the number of thresholds, the difference is more noteworthy and the MMPSO leads with the higher fitness value than other methods.

5 Conclusion

In this paper, we have proposed a method called MMPSO inspired from the Particle Swarm Optimization algorithm based on a new fitness function and the Otsu's method for multilevel thresholding. This method is able to determine optimal threshold values from complex gray-level images. In this purpose, a new fitness function is developed to ensure best threshold values in less CPU process time and with the best stability due to the best STD value. Experimental results demonstrated by computing optimal threshold values in 4 different levels (3, 4, 5 and 6 levels) for three different benchmark images. Results indicate that the MMPSO is more efficient than basic PSO, Otsu's method, GA and MEACBT methods. In particular, this method is better when the level of segmentation increase and the image is with more details.

Moreover, due to the low computational complexity of the algorithm and the higher stability of the MMPSO algorithm, this algorithm (MMPSO) will be applied to classify the MRI medical images. Also, the segmentation results are promising and it encourage further researches for applying the MMPSO algorithm to complex and real-time MRI image segmentation problem.

References

1. Melouah, A.: A novel region growing segmentation algorithm for mass extraction in mammograms. Model. Approaches Algorithms Adv. Comput. Appl. Stud. Comput. Intel. **488**, 95–104 (2013)
2. Chakraborty, J., Mukhopadhyay, S., Singla, V., Khandelwal, N., Rangayyan, R.M.: Detection of masses in mammograms using region growing controlled by multilevel thresholding. In: The 25th International Symposium on Computer-Based Medical Systems (CBMS), Rome, pp. 1–6, 20–22 June 2012. doi: 10.1109/CBMS.2012.6266308
3. Dragon, R., Ostermann, J., Van Gool, L.: Robust realtime motion-split-and-merge for motion segmentation. In: The 2013 35th German Conference on Computer Science, GCPR. Saarbrücken, Germany, pp. 425–434, 3–6 Sept 2013. doi:10.1007/978-3-642-40602-7_45
4. Chaudhuri, D., Agrawal, A.: Split-and-merge procedure for image segmentation using bimodality detection approach. Defence Sci. J. **60**(3), 290–301 (2010)
5. Cao, X., Ding, W., Hu, S., Su, L.: Image segmentation based on edge growth. In: Proceedings of the 2012 International Conference on Information Technology and Software Engineering, pp. 541–548 (2013). doi:10.1007/978-3-642-34531-9_57

6. Sharif, M., Raza, M., Mohsin, S.: Face recognition using edge information and DCT. Sindh Univ. Res. J. (Sci. Ser.) **43**(2), 209–214 (2011)
7. Baakek, T., Chikh Mohamed, A.: Interactive image segmentation based on graph cuts and automatic multilevel thresholding for brain images. J. Med. Imaging Health Inform. **4**(1), 36–42 (2014)
8. Martin-Rodriguez, F.: New tools for gray level histogram analysis, applications in segmentation. In: 10th International Conference in Image analysis and recognition, ICIAR, Póvoa do Varzim-Portugal, pp. 326–335, 26–28 June 2013. doi:10.1007/978-3-642-39094-4_37
9. Qifang, L., Zhe, O., Xin, C., Yongquan, Z.: A multilevel threshold image segmentation algorithm based on glowworm swarm optimization. J. Comput. Inf. Syst. **10**(4), 1621–1628 (2014)
10. Kulkarni, R.V., Venayagamoorthy, G.K.: Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. IEEE Trans. Syst. Man Cybern. **40**(6), 663–675 (2010)
11. Hamdaoui, F., Ladgham, A., Sakly, A., Mtibaa, A.: A new images segmentation method based on modified PSO algorithm. Int. J. Imaging Syst. Technol. **23**(3), 265–271 (2013)
12. Ladgham, A., Hamdaoui, F., Sakly, A., Mtibaa, A.: Fast MR brain image segmentation based on modified shuffled frog leaping algorithm. DOI, Signal Image Video Process. (2013). doi:10.1007/s11760-013-0546-y
13. Sun, H.J., Deng, T.Q., Jiao, Y.Y.: Remote sensing image segmentation based on rough entropy. In: 4th International Conference in Advances in Swarm Intelligence ICSI, pp. 11–419, 12–15 June 2013. doi:10.1007/978-3-642-38715-9_49
14. Sarkar, S., Sen, N., Kundu, A., Das, S., Chaudhuri, S.S.: A differential evolutionary multilevel segmentation of near infra-red images using Renyi's entropy. In: International Conference on Frontiers of Intelligent Computing: Theory and Applications FICTA, pp. 699–706, (2013). doi:10.1007/978-3-642-35314-7_79
15. Daisne, J.F., Sibomana, M., Bol, A., Doumont, T., Lonneux, M., Grégoire, V.: Tri-dimensional automatic segmentation of PET volumes based on measured source-to-background ratios: influence of reconstruction algorithm. Radiother. Oncol. **69**(3), 247–250 (2003)
16. Huang, D.Y., Lin, T.W., Hu, W.C.: Automatic multilevel thresholding based on two-stage Otsu's method with cluster determination by valley estimation. Int. J. Innovative Comput. Inf. Control **7**(10), 5631–5644 (2011)
17. Ningning, Z., Tingting, Y., Shaobai, Z.: An improved FCM medical image segmentation algorithm based on MMTD. Comput. Math. Methods. Med. (2014). http://dx.doi.org/10.1155/2014/690349
18. Yasmin, M., Mohsin, S., Sharif, M., Raza, M., Masood, S.: Brain image analysis: a survey. World Appl. Sci. J. **19**(10), 1484–1494 (2012)
19. Raza, M., Sharif, M., Yasmin, M., Masood, S., Mohsin, S.: Brain image representation and rendering: a survey. Res. J. Appl. Sci. Eng. Technol. **4**(18), 3274–3282 (2012)
20. Al-azawi, M.: Image thresholding using histogram fuzzy approximation. Int. J. Comput. Appl. **83**(9), 36–40 (2013)
21. Nakib, A., Roman, S., Oulhadj, H., Siarry, P.: Fast brain MRI segmentation based on two-dimensional survival exponential entropy and particle swarm optimization. In: International Conference of the IEEE EMBS. Lyon, France, pp. 5563–5566, 23–26 Aug 2007. doi:10.1109/IEMBS.2007.4353607
22. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66
23. Yao, C., Chen, H.J.: Automated retinal blood vessels segmentation based on simplified PCNN and fast 2D-Otsu algorithm. J. Cent. S. Univ. Technol. **16**(4), 640–646 (2009)
24. Huang, D.Y., Wang, C.H.: Optimal multi-level thresholding using a two-stage Otsu optimization approach. Pattern Recogn. Lett. **30**(3), 275–284 (2009)
25. Wu, B.F., Chen, Y.L., Chiu, C.C.: Recursive algorithms for image segmentation based on a discriminant criterion. Int. J. Sig. Process. **1**, 55–60 (2004)

26. Hammouche, K., Diaf, M., Siarry, P.: A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. Eng. Appl. Artif. Intell. **23**(5), 676–688 (2010)
27. Hammouche, K., Diaf, M., Siarry, P.: A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Comput. Vis. Image Underst. **109**(2), 163–175 (2008)
28. Tao, W.B., Tian, J.W., Liu, J.: Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. Pattern Recogn. Lett. **24**(16), 3069–3078 (2003)
29. Yang, Z., Pu, Z., Qi, Z.: Relative entropy multilevel thresholding method based on genetic optimization. In: The 2003 IEEE International Conference on Neural Networks and Signal Processing, Nanjing, pp. 583–586, 14–17 Dec 2013. doi:10.1109/ICNNSP.2003.1279340
30. Hancer, E., Ozturk, C., Karaboga, D.: Artificial bee colony based image clustering method. In: IEEE International Congress on Evolutionary Computation, Brisbane, QLD, pp. 1–5, 10–15 June 2012. doi:10.1109/CEC.2012.6252919
31. Zhang, Y., Wu, L.: Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. Entropy **13**(4), 841–859 (2011)
32. Geng, R.: Color image segmentation based on self-organizing maps, advances in key engineering materials. Adv. Mater. Res. **214**, 693–698 (2011)
33. Bhandari, A.K., Singh, V.K., Kumar, A., Singh, G.K.: Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. Expert Syst. Appl. **41**(7), 3538–3560 (2014)
34. Gao, H., Kwong, S., Yang, J., Cao, J.: Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation. Inf. Sci. **250**(20), 82–112 (2013)
35. Ghamisi, P., Couceiro, M.S., Benediktsson, J.A., Ferreira, N.M.F.: An efficient method for segmentation of images based on fractional calculus and natural selection. Expert Syst. Appl. **39**(16), 12407–12417 (2012)
36. Tillett, J., Rao, T.M., Sahin, F., Rao, R., Brockport, S.: Darwinian particle swarm optimization. In: The 2nd Indian International Conference on Artificial Intelligence, pp. 1474–1487 (2005)
37. Couceiro, M.S., Ferreira, N.M.F., Machado, J.A.T.: In fractional order Darwinian particle swarm optimization. In FSS'11, Symposium on Fractional Signals and Systems, Coimbra, Portugal, pp. 2382–2394, 4–5 Nov 2011. doi:10.1109/TGRS.2013.2260552
38. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
39. Goldberg, D.E.: Algorithmes Génétiques: Exploration, optimisation et apprentissage automatique, Edition Wesley (1989)
40. Holland, J.H.: Genetic algorithms, pour la science. Ed. Sci. Am. **179**, 44–50 (1992)
41. Man, K.F., Tang, K.S., Kwong, S.: Genetic algorithms: concepts and applications. IEEE Trans. Industr. Electron. **43**(5), 519–534 (1996)
42. Schmitt, L.M.: Fundamental study: theory of genetic algorithms. Theoret. Comput. Sci. **259**(1–2), 1–61 (2001)
43. Petrowski, A.: Une introduction à l'optimisation par algorithmes génétiques, (2001). http://www-inf.int-evry.fr/~ap/EC-tutoriel/Tutoriel.html
44. Phulpagar, B.D., Kulkarni, S.S.: Image segmentation using genetic algorithm for four gray classes. In: IEEE International Conference on Energy, Automation and Signal, 28–30 Dec 2011. Bhubaneswar, Odisha, pp. 1-4. doi:10.1109/ICEAS.2011.6147093
45. Phulpagar, B.D., Bichkar, R.S.: Segmentation of noisy binary images containing circular and elliptical objects using genetic algorithms. IJCA **66**(22), 1–7 (2013)
46. Janc, K., Tarasiuk, J., Bonnet, A.S., Lipinski, P.: Genetic algorithms as a useful tool for trabecular and cortical bone segmentation. Comput. Methods Programs Biomed. **111**(1), 72–83 (2013). doi:10.1016/j.cmpb.2013.03.012

47. Manikandan, S., Ramar, K., Willjuice, I.M., Srinivasagan, K.G.: Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm. Measurement **47**, 558–568 (2014)
48. Dorigo, M., Gambardella, L.M.: Guest editorial special on ant colony optimization. IEEE Trans. Evol. Comput **6**(4), 317–319 (2002)
49. Ajith, A., Crina, G., Vitorino, R.: Stigmergic Optimization. Stud. Comput. Intel. **31**, 1–299 (2006)
50. Beckers, R., Deneubourg, J.L., Goss, S.: Trails and U-turns in the selection of a path by the Ant Lasius Niger. J. Theor. Biol. **159**(4), 397–415 (1992)
51. Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M.: Self-organized shortcuts in the argentine ant. Naturwissenchaften **76**(12), 579–581 (1989)
52. Dorigo, M., Maniezzo, V., Colorni, V.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B Cybern. **26**(1), 29–41 (1996)
53. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: The First European Conference on Artificial Life. MIT Press, Paris, France, pp. 134–142, (1991)
54. Mousa, A.A., El-Desoky, I.M.: Stability of Pareto optimal allocation of land reclamation by multistage decision-based multipheromone ant colony optimization. Swarm Evol. Comput. **13**, 13–21 (2013)
55. Liang, Y.C., Yin, Y.C.: Optimal multilevel thresholding using a hybrid ant colony system. J. Chin. Inst. Ind. Eng. **28**(1), 20–33 (2011)
56. Ma, L., Wang, K., Zhang, D.: A universal texture segmentation and representation scheme based on ant colony optimization for iris image processing. Comput. Math. Appl. **11**(12), 1862–1866 (2009)
57. Tao, W., Jin, H., Liu, L.: Object segmentation using ant colony optimization algorithm and fuzzy entropy. Pattern Recogn. Lett. **28**(7), 788–796 (2007)
58. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey (2005)
59. Basturk, B., Karaboga, D.: An artificial bee colony (abc) algorithm for numeric function optimization. In: IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, May 2006
60. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. **8**(1), 687–697 (2008)
61. Karaboga, D., Basturk, B.: Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: Foundations of Fuzzy Logic and Soft Computing. Lecture Notes in Computer Science, vol. 45(29), pp. 789–798 (2007)
62. Hadidi, A., Azad, S.K., Azad, S.K.: Structural optimization using artificial bee colony algorithm. In: The second International Conference on Engineering Optimization. Lisbon, Portugal, 6–9 Sept 2010
63. Tereshko, V., Loengarov, A.: Collective decision-making in honeybee foraging dynamics. Comput. Inf. Syst. J. **9**(3), 1–7 (2005)
64. Horng, M.H.: Multilevel minimum cross entropy thresholding using artificial bee colony algorithm. Telkomnika **11**(9), 5229–5236 (2013)
65. Akay, B.: A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl. Soft Comput. **13**(6), 3066–3091 (2013)
66. Charansiriphaisan, K., Chiewchanwattana, S., Sunat, K.: A comparative study of improved artificial bee colony algorithms applied to multilevel image thresholding. Math. Prob. Eng., 1–17 (2013). http://dx.doi.org/10.1155/2013/927591
67. Cao, Y.F., Xiao, Y.H., Yu, W.Y., Chen, Y.C.: Multi-level threshold image segmentation based on PSNR using artificial bee colony algorithm. Res. J. Appl. Sci. Eng. Technol. **4**(2), 104–107 (2012)

68. Horng, M.H., Jiang, T.W: Multilevel image thresholding selection using the artificial bee colony algorithm. In: International Conference on Artificial Intelligence and Computational Intelligence, Sanya, China, pp. 318–325, 23–24 Oct 2010. doi:10.1007/978-3-642-16527-6_40
69. Eusuff, M.M., Lansey, K.E.: Optimization of water distribution network design using the shuffled frog leaping algorithm. J. Water Resour. Plan. Manag. **129**(3), 210–225 (2003)
70. Duan, Q.Y., Gupta, V.K., Sorooshian, S.: Shuffled complex evolution approach for effective and efficient global minimization. J. Optim. Theory Appl **76**(3), 502–521 (1993)
71. Fang, C., Chang, L.: An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. Comput. Oper. Res. **39**(5), 890–901 (2012)
72. Narimani, M.R.: A new modified shuffle frog leaping algorithm for non-smooth economic dispatch. World Appl. Sci. J. **12**(6), 803–814 (2011)
73. Wang, N., Li, X., Chen, X.H.: Fast three-dimensional Otsu thresholding with shuffled frog-leaping algorithm. Pattern Recognit. Lett. Meta-heuristic Intel. Based Image Process. **31**(13), 1809–1815 (2010)
74. Liong, S.Y., Atiquzzaman, M.: Optimal design of water distribution network using shuffled complex evolution. J. Inst. Eng. **44**(1), 93–107 (2004)
75. Gu, Y.J., Jia, Z.H., Qin, X.Z., Yang, J., Pang, S.N.: Image segmentation algorithm based on shuffled frog-leaping with FCM. Commun. Technol. **2**, 042 (2011)
76. Yang, C.S., Chuang, L.Y., Ke, C.H.: A combination of shuffled frog-leaping algorithm and genetic algorithm for gene selection. J. Adv. Comput. Intell. Intell. Inf. **12**(3), 218–226 (2008)
77. Horng, M.H.: Multilevel image threshold selection based on the shuffled frog-leaping algorithm. J. Chem. Pharm. Res. **5**(9), 599–605 (2013)
78. Ouadfel, S., Meshoul, S.: A fully adaptive and hybrid method for image segmentation using multilevel thresholding. Int. J. Image Graph. Sig. Process. (IJIGSP) **5**(1), 46–57 (2013)
79. Horng, M.H.: Multilevel image thresholding by using the shuffled frog-leaping optimization algorithm. In: 15th North-East Asia Symposium on Nano Information Technology and Reliability (NASNIT), Macao, pp. 144–149, 24–26 Oct 2011. doi:10.1109/NASNIT.2011.6111137
80. Jiehong, K., Ma, M.: Image Thresholding Segmentation Based on Frog Leaping Algorithm and Ostu Method. Yunnan University (Natural Science Edition), pp. 634–640 (2012)
81. Liu, J., Li, Z., Hu, X., Chen, Y.: Multiobjective optimization shuffled frog-leaping biclustering. In: IEEE International Conference on Bioinformatics and Biomedicine Workshops, Atlanta, pp. 151–156, 12–15 Nov 2011. doi:10.1109/BIBMW.2011.6112368
82. Bhaduri, A., Bhaduri, A.: Color image segmentation using clonal selection-based shuffled frog leaping algorithm. In: International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom '09. Kottayam, Kerala, pp. 517–520, 27–28 Oct 2009. doi:10.1109/ARTCom.2009.115
83. Couceiro, M.S., Luz, J.M.A., Figueiredo, C.M., Ferreira, N.M.F., Dias, G.: Parameter estimation for a mathematical model of the golf putting. In WACI'10, Workshop Applications of Computational Intelligence ISEC-IPC, Coimbra, Portugal, pp. 1–8, 2 Dec 2010 (2010a)
84. Couceiro, M.S., Ferreira, N.M.F., Machado, J.A.T.: Application of fractional algoritms in the control of a robotic bird. J. Commun. Nonlinear Sci. Numer. Simul. (Special Issue) **15**(4), 895–910 (2010b)
85. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th Symposium on Micro Machine and Human Science, Nagoya, pp. 39–43, 4–6 Oct 1995. doi:10.1109/MHS.1995.494215
86. Kennedy, J., Eberhart, R. C. (1995). Particle swarm optimization. In IEEE International Conference Neural Network, 27 Nov–01 Dec 1995, Perth WA, pp. 1942–1948 (2005). doi:10.1109/ICNN.1995.488968
87. Jiang, M., Luo, Y.P., Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf. Process. Lett. **102**(1), 8–16 (2007)

88. Fan, J., Han, M., Wang, J.: Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation. Pattern Recogn. **42**, 2527–2540 (2009)
89. Horng, M.H.: Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst. Appl. **38**(11), 13785–13791 (2011)
90. Ghamisi, P., Couceiro, M.S., Benediktsson, J.A., Ferreira, M.F.N.: An efficient method for segmentation of images based on fractional calculus and natural selection. Expert Syst. Appl. **39**(16), 12407–12417 (2012)

IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm

Nizar Rokbani, Alicia Casals and Adel M. Alimi

Abstract In this paper, a heuristic method based on Firefly Algorithm is proposed for inverse kinematics problems in articulated robotics. The proposal is called, IK-FA. Solving inverse kinematics, IK, consists in finding a set of joint-positions allowing a specific point of the system to achieve a target position. In IK-FA, the Fireflies positions are assumed to be a possible solution for joints elementary motions. For a robotic system with a known forward kinematic model, IK-Fireflies, is used to generate iteratively a set of joint motions, then the forward kinematic model of the system is used to compute the relative Cartesian positions of a specific end-segment, and to compare it to the needed target position. This is a heuristic approach for solving inverse kinematics without computing the inverse model. IK-FA tends to minimize the distance to a target position, the fitness function could be established as the distance between the obtained forward positions and the desired one, it is subject to minimization. In this paper IK-FA is tested over a 3 links articulated planar system, the evaluation is based on statistical analysis of the convergence and the solution quality for 100 tests. The impact of key FA parameters is also investigated with a focus on the impact of the number of fireflies, the impact of the maximum iteration number and also the impact of (α, β, γ, δ) parameters. For a given set of valuable parameters, the heuristic converges to a static fitness value within a fix maximum number of iterations. IK-FA has a fair convergence time, for the tested configuration, the average was about 2.3394×10^{-3} seconds with a position error fitness around

---

N. Rokbani (✉)
High Institute of Applied Sciences and Technology, University of Sousse,
Cité Taffala (Ibn Khaldoun), Sousse 4003, Tunisia
e-mail: nizar.rokbani@ieee.org

N. Rokbani · A.M. Alimi
REGIM-Lab.: REsearch Groups in Intelligent Machines, University of Sfax, ENIS,
BP 1173, Sfax 3038, Tunisia
e-mail: adel.alimi@ieee.org

A. Casals
Institute for Bioengineering of Catalonia and Universitat Politècnica de Catalunya.
BarcelonaTech., Barcelona, Spain
e-mail: alicia.casals@upc.edu

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_15

3.116×10^{-8} for 100 tests. The algorithm showed also evidence of robustness over the target position, since for all conducted tests with a random target position IK-FA achieved a solution with a position error lower or equal to 5.4722×10^{-9}.

Keywords Robotics · Inverse kinematics · Heuristics · Computational kinematics · Swarm intelligence

1 Introduction

Inverse Kinematics, IK, is still a challenging problem in robotic systems, essentially needed for path planning, motion generation and control. In robotics, trajectories generation is prior to control in a large panel of applications including wheeled, legged or articulated systems. Path and motion planning are generally expressed in the Cartesian frame, the control is performed relative to the joint-frames. IK produces the motions of the joints in accordance with the needed Cartesian path or trajectory [33, 46]. For this class of application several techniques for solving IK was investigated, due to the complexity of the analytical solutions essentially when the system has a high number of Degrees of Freedom, DOF.

In computer games applications, IK has to tackle a couple of constraints, being real time and producing precise solutions so that the animations can be perceived as natural as possible. The cyclic coordinate descent technique, CCD, was first proposed for that purpose before being applied to robotics, it is one of the earliest heuristic inverse kinematics solvers, providing a good balance between computational cost and stability [17], CCD was not built to find optimal solutions but to find a fast feasible solution in a limited computing time. Since that several heuristics were investigated with more or less success [20, 23, 29]. In heuristic based IK solver we can identify two main classes: Computational based techniques and learning based techniques. The first class is based on nature inspired heuristics such as Particle Swarm Optimization, PSO [25, 26, 40], Ant Colony Optimization, ACO [20], Genetic Algorithms, GA [12, 46]; Buckley et al. [4] or Ant Bee Colony, ABC [6]. For these methods, the IK problem is addressed as an optimization and the heuristics is used as computing alternatives. The second class includes nature inspired techniques and hybrid techniques with learning capacities such as neural networks and neuro-fuzzy systems [13, 22, 32, 45] where the IK solver is observed as system with a set of inputs and output(s), the system needs first to be trained using a set of targets and solutions generally obtained using the forward kinematic model, it is then validated with a separate validation set and finally used. When the target point is far from the training set this systems generate limited precision solutions compared to Jacobian based conventional methods [5, 7]; they are also time consuming and can not satisfy real time constraints [22, 44].

In humanoid robotics, legs, arms and fingers are typical articulated systems, with a similar kinematic chain while dynamics are different. Inverse kinematics is needed at all levels: for leg motion planning in walking, for arms and fingers motions in handling and grasping [2, 36]. Due to these reasons, it is easier to plan motion in the Cartesian frame, since it is the physical frame of the environment. Then an inverse transformation is needed to compute the needed displacements in the joints frames prior to control [24, 27], such a technique is also used in 3D character animation or gaming applications. In humanoid robotics and 3D humanoids simulations, the human motion analysis was a key source of inspirations for walking gaits, arms motions, grasping [37, 39], body postures and biped walking [1, 3]. At this level also inverse kinematics are needed to generate skeleton's joint motions that could fit a robotic design while satisfying a human like motion in the Cartesian frame [16] form a set of marked human motion primitives. Such inverse solvers should satisfy real-time constraints and should not suffer from any singularity which is not the case of classical IK solvers. Classical techniques consists in finding an approximation of the inverse kinematics of a system when the analytical expression of the inverse is complex and difficult to compute; this class of methods include the pseudo-inverse methods, the Jacobian transpose, the quasi Newton and the damped least square methods; classical inverse methods are time consuming essentially in systems with a high DOF [5, 7].

In inverse kinematics based on PSO [25]; or on GA such as [4, 12, 46], a stochastic search is performed, using a population in GA or a set of individuals in PSO, ABC and ACO; each population or individual is a possible solution, here a set of joints positions of the IK problem. Any possible solution is ranked, using a fitness function and the best is returned as the solution of the problem, for a given input, here a target position. The common aspects of these methods is that they try to solve inverse kinematics by evolving iteratively a set of solutions using a limited set of operations that mimic a natural process, swarm behavior and its social organization in the case of PSO, ABC or ACO [9, 10, 14]; GA tends to solve a problem using the natural evolution mechanisms [21]. The design of the optimality criteria is what makes a heuristic render an acceptable result. In IK, a trivial objective could be used, it consists in minimizing the distance to the target position. For applications such as humanoid gait generation, it is possible to add some constraints to obtain a solution which fits better this specific class of robotic systems. In real world applications, the solutions of an articulated arm or an artificial leg should respect the mechanical design of the system. The adaptation of the heuristic IK methods is possible for any kind of robotic system and do not need to be trained.

On the other hand neural network [1] and neuro-fuzzy techniques tend to do the same but after a training process, where a neural or a neuro-fuzzy network is trained by a set of joint motions and their correspondent Cartesian solutions [13, 32, 44, 45]. The training process has a direct impact on the quality of the obtained IK solver, for these intelligent IK solvers designing a good training set is essential. Neuro-fuzzy has the advantage to be interpretable when compared to neural-network IK solutions;

they are accurate but suffer from computing time, and could not be used in real time applications [22, 37].

The remaining of this paper is organized as detailed below. Section 2 reviews the key issues of kinematics modeling with a focus on inverse kinematics challenges; this section reviews the concept of heuristic solvers with a focus on CCD, which is a reference heuristic for IK. Section 3 stars with a review on the FA heuristic, then a new heuristic approach for inverse kinematics based on the Firefly Algorithm, FA, is proposed, the proposal is called IK-FA. IK-FA proposal is detailed for unconstrained and constrained inverse kinematics problems. In Sect. 4 a set of simulation based experiments are detailed, the key aspects of IK-FA were subject to investigation over a classical 3 links articulated system, investigations concerned the impact of IK-FA parameters on convergence and performances. Finally the paper ends with discussions, conclusions and perspectives.

2 Inverse Kinematics

In Robotics two aspects are important, kinematics and dynamics [36, 37]; Kinematics deals with how the motions of a mechanism are related to the relative positions of the end effectors of the system in accordance with a reference frame [5, 37]. The motion is studied regardless to what produced it. In robotics kinematics analysis are needed to plan a robot motion with respect to the work space geometric configuration and satisfying angular and geometric constraints that the system could be subject to [24]. Kinematics is forward or inverse: In forward kinematics the mechanism motions are known while the end effectors positions need to be computed. In inverse kinematics the end effectors positions are known and the joint motions involved to achieve them need to be computed [5, 35].

2.1 Forward Kinematics

Assume that $X = (X_1, X_2, \ldots X_l)$ is the position of an articulated body of (n) elements subject to a set of elementary rotations and/or translations, $q = (\theta_1, \theta_2, \ldots \theta_n)$ so the forward kinematics is expressed by Eq. (1).

$$X = f(q) \tag{1}$$

The forward kinematics model can be obtained systematically whatever is the complexity of the mechanism. It is decomposed into a set of primitive Transformations according to a coordinate frame and the forward kinematics function is obtained systematically by composing the elementary transformations T^i_{i-1},

$$f(q) = \prod_{i=1}^{n} T_{i-1}^{i}. \tag{2}$$

An elementary transformation refers to a rotation towards an axe of the coordinate's frames or a translation on a given direction.

2.2 Inverse Kinematics

Inverse kinematics consists in finding a possible and feasible joints motions solution allowing a robotic system, typically articulated, achieving a pre-defined position, called target position. For an articulated system such in Fig. 1, let's assume the joint rotations needed to produce the motion to $q = (\theta_1, \ldots, \theta_n)$; the robot position in a Cartesian reference frame $X = (x_1, \ldots, x_l)$, is obtained as the output of the forward kinematics function, f(q) of the system with Qi as input as in (1). If we assume that the forward kinematics are expressed using a mathematical function f(), inverse kinematics, IK, could simply be the inverse of that function meanwhile and considering the nature of f() which is a matrix, its inverse is not for sure defined, retrieving the inverse kinematics function depends on the invertibility of the forward form, the generic formulation of Eq. (3) is in most cases difficult to compute and some approximations are needed to retrieve the IK models [5, 34, 35].

$$q = f^{-1}(X) \tag{3}$$

The main problem in IK is the existence $f^{-1}()$; in the most cases an analytic expression of the IK function is difficult to obtain. Several computational methods are proposed to tackle this problem. The most Known approach is based on the

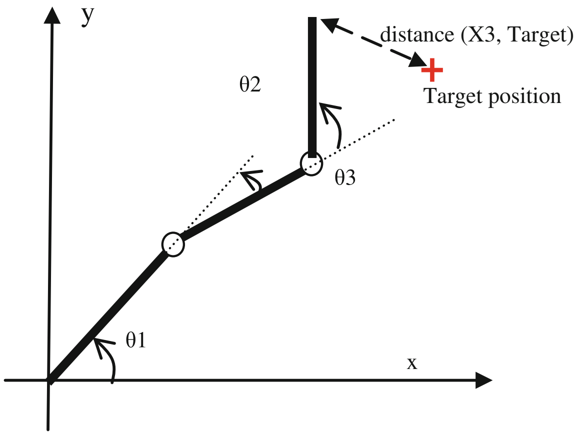


Fig. 1 Simplified representation of an articulated system composed by 3 links and 3 revolute joints

Jacobian of forward kinematics function *f*(). The Jabocian is the multidimensional expression of the classical differential operator as in Eq. (4), it is also possible to compute J(q) iteratively [2, 5, 36].

$$J(q) = \left(\frac{df}{dq}\right) \tag{4}$$

Using (1) and (4) the linear velocity could be expressed according to the angular velocities of the joints angular positions by (5).

$$\frac{dX}{dt} = J(q)\frac{dq}{dt} \tag{5}$$

For very limited motions, the previous equation could be expressed using differential form instead of the derivates; It leads to an expression of the elementary position displacement for a given elementary joint displacements as in (6)

$$\delta X = J(q)\delta q \tag{6}$$

The inverse form of this equation is expressed in (7) allowing to compute the small amounts of joints positions changes for a given small relative variation of the end effectors position. Here the problem of IK is simply transformed in computing or finding the inverse of the Jacobian of the forward kinematics function.

$$\delta q = J(q)^{-1} * \delta X \tag{7}$$

Around this concept, several methods are proposed all of them belong to the same class of IK solutions, the Jacobian based IK. Note that if the dimension of the Cartesian position vector is different from the dimension of the joints angular rotations vector q(), the J(q) matrix is rectangular and is simply not invertible, it could also suffer form singularities. For systems with high DOF, the analytical solution of inverse kinematics is difficult to express [5]. The Jacobian transpose method used the transpose of J(q) instead if its inverse. The pseudo inverse method replaces the Jacobian by its pseudo inverse, while it is not the exact inverse of J is still a good approximation. The main challenge with Jacobian inverse methods is how to compute the Jacobian directly or iteratively [5, 7].

2.3 Heuristic Inverse Kinematics

For a given target position and knowing the current position, the classical method to solve inverse kinematics consists in retrieving a q() using the inverse kinematics function as in Eq. (3). This solution exists only if the inverse kinematics function is defined, while in the most of the cases this function is hard to obtain, and suffer from singularities [5].

An Heuristic solution to this problem consists in using an heuristic search method to find iteratively a set of q(), by reducing the position error, as in Eq. (8), which is the distance of the end-segment to the needed target position, as shown in Fig. 1. In the case of PSO, particle swarm optimization, the heuristic will guide the search using a set of particles; each one is a potential solution of the problem [10, 11]. The quality of a solution is evaluated using a fitness function, which is naturally quantifying the error of the obtained target versus the needed one.

$$e = \|x_t - f(q_i)\| \tag{8}$$

In inverse kinematics solver using PSO, IK-PSO, the fitness function is the distance of the end-effectors, or specific point of the system to the target [25, 26, 28], its general expression of a dimension (d) is given by (9).

$$f_{itness(i)} = \sum_{i=1}^{d} (x_i - x_t)^2 = \sum_{i=1}^{d} (f(q)_i - x_t)^2 \tag{9}$$

where (d) denotes the dimension, in the case of Fig. 1, d = 2. and the fitness function is expressed by (10).

$$f_{itness(i)} = (f(q)_{xi} - x_t)^2 + (f(q)_{yi} - y_t)^2 \tag{10}$$

The fitness function is homogenous with the square of a Euclidian distance. In IK-PSO, the inverse kinematics is solved using the forward kinematics function and a heuristic search strategy. By over coming the computing of the inverse forms IK-PSO has no problem of definition, singularities and does no need any matrix inverse computing. The effectiveness of PSO allowed coming to a solution within reasonable time and position error [25].

In the case of the cyclic coordinate descent (CCD), the computational efforts are reduced by varying one joint variable at a time. The goal is to reduce position and orientation errors. At each iteration, a single traversal of the manipulator is performed from the target position, where the final link is, towards the manipulator base; this allowed to CCD to produce an iterative IK solution [2, 4, 11]. For an articulated system composed of (n) links CCD solves a single joint position q(i) using a minimization of the end-segment to the target point.

The link direction is estimated using classical trigonometric approximation of a virtual link joining the target position to the link base, this mean that the joints rotations are produced iteratively and this could lead to non feasible solutions for a robotic system. On the other hand and from a computational point of view, CCD has the advantage to render a solution in a limited time since it computes (n) elementary motions instead of the solution of an (n) joints system. The method handle a single DOF problem relative to a joint q(i), such a problem is simple to solve using classical trigonometry.

3 Inverse Kinematics Using Firefly Algorithm, IK-FA

In this paragraph we first give a brief on the essentials of Firefly Algorithm, FA, methaheuritic, and then a detailed description on how FA was used to solve inverse kinematics is given.

3.1 The Firefly Algorithm (FA)

The Firefly Algorithm (FA) was proposed by Xin-She Yang, as a bio-inspired heuristic from the flashing behaviour of the fireflies. Fireflies use flash light to attract others, the more a light is visible, the more its attraction capacity rises, only this aspect was considered by Xin-She Yang in his algorithm regardless to what is intended behind [43], it was then tested over the classical test functions [18, 41, 42]. FA, assumed all individual to be unisexual, an individual is attracted by a light with a flashing capacity which is higher to its own. The light intensity naturally decreases when the distance to light source increases. The relative displacement between fireflies is given by a position update equation as in (11).

$$x_i = x_i + \beta(x_j - x_i) + \alpha\varepsilon \tag{11}$$

where, x_i, x_j: are respectively the current position of the fireflies (i) and (j). It is important to note here that the position update of any firefly is adjusted with regard to its own current position and also the current positions of all swarm individuals. This is the main difference between FA algorithm and PSO where an individual position depends on a couple of specific particles, the local best and the global best.

An individual of a FA swarm is moved towards any other with higher brightness, this displacement is moderated by the attractiveness coefficient β and a random displacement (αε). The neighborhood is composed of fireflies within the perception filed of the individual. The firefly with a lower brightness is moved towards the one with a higher brightness [43]; a simplified pseudo-code of FA algorithm is given in Fig. 2.

In Eq. (11) the term β represents the attractiveness coefficient which depends also on the distance separating the firefly (i) to individual (i), it is expressed as in Eq. (12).

$$\beta = \beta_0 \mathrm{e}^{(\gamma r_{ij}^2)} \tag{12}$$

The final term of Eq. (11) could be observed as a step size with a moderation parameter α and were ε could be derived randomly from a Gaussian distribution. In FA, the brightness of a firefly I(x) could be used as the fitness function of the problem to optimize as expressed in (13).

```
Begin
  Initialize FA parameters : $\gamma$, $\alpha$, $\beta$,
  Generate an initial population of fireflies Q =(0,…..xn)
    For i_t=1 : maximum_iteration
      for i = 1 : n (all n fireflies)
        for j = 1 : n (n fireflies)
          Compute Ii, Ij
          if (Ii < Ij )
          move firefly i towards j
          end if
      end for j
      end for i
      Solution = best current fireflies;
      Reduce $\alpha$
    End for i_t
   Return best firefly (Solution);
End
```

Fig. 2 FA Algorithm pseudo code

$$I_o^i = f_{itness}(x_i) \tag{13}$$

The brightness of an individual (i) is also subject to a natural lost when observed from the position of an individual (j), this lost is expressed as in Eq. (14), where (r) is the distance of firefly (j) to firefly (i) and γ the absorption coefficient.

$$I_i = I_o^i \cdot \mathrm{e}^{(-\gamma r)}. \tag{14}$$

3.2 The IK-FA for Inverse Kinematics

This paragraph details how FA is adapted to solve IK for unconstrained inverse kinematics prior to detail how constraints can be handled. The inverse kinematics solver using PSO is called IK-FA, Inverse Kinematics solver using Firefly Algorithm.

3.2.1 Unconstrained IK-FA

To use FA in solving inverse kinematics, the firefly here is assumed to be set motions primitives limited to rotations and translations. The Fireflies positions correspond to robotic motion expressed in the joint frames; then the equivalent position of the end-segment is retrieved in the Cartesian Frame using the Forward kinematics of the system. Here, we have no more to care about singularities or to inverse the forward kinematics function. Only the Forward kinematics, a target position and a satisfaction condition is needed, a trivial satisfaction condition only could simply be a fixed error position.

Given a system composed of (n) joints, a firefly position at iteration (i) can be expressed as in (15), fireflies evolves in swarms of (n) individuals. For an articulated system such in Fig. 1, where the joints are only subject to rotations, the FA position is expressed by Eq. (15).

$$(q_j)_i = (\theta_1^j, \ldots, \theta_n^j)_i \tag{15}$$

The Firefly Algorithm is an iterative computing heuristic and Eq. (15) refers to the position of the firefly (j) at iteration (i). The Cartesian position of end-segment, obtained with the joints solution of firefly (j) at iteration (i) is expressed as in (16).

$$X_{ji} = f(\theta_1^j, \ldots, \theta_n^j)_i \tag{16}$$

where f() stands for the forward kinematics function of the system. A trivial fitness function for the system could be expressed by the square of Euclidian distance of the target point to the end-segment position of the system, see Eq. (17).

$$f_{itness}(q_j)_i = (x_t - x_{ji})^2 \tag{17}$$

The brightness of a firefly is maximized as it comes closer to the target position, while instead of using the exponential function, its first order differential equation is used and the brightness of a firefly is expressed by (18).

$$I_j^i = \frac{1}{1 + \gamma(x_t - x_{ji})^2} \tag{18}$$

where (j) is the firefly identifier and (i) the iteration counter, the brightness is designed so that it comes to a maximum as the target point is achieved, I = 1. Note that the brightness is related to the distance of a firefly to target position while the firefly himself is an angular position. As is PSO, a stop is observed when the maximum number of iteration is achieved or when the fitness function is satisfied.

3.2.2 Constrained IK-FA

To solve inverse kinematics of systems subject to constraints, such as mechanical arms or bio-inspired systems with biomechanical constraints like artificial limbs, the IK-FA should respect Cartesian constraints and joints constraints. In Constrained IK-FA generated solutions are subject to a couple of tests, the first one consists in checking if joint constraints are respected. The second check is about the Cartesian constraints.

Cartesian constraints verification consists in verifying if a set of end-effectors are within a predefined work space, meanwhile the firefly, by mean of solution, is simply ignored.

Joints constraints are needed to limit the motion into a feasible space. The joints constraints could be imposed by the mechanical design, in this case any motion with violation to those constraints should be simply avoided. Joints constraints could also be useful to produce gaits that are close a predefined biomechanical one: for example knee and ankles rotation limits in the case of human walking or shoulder and elbow limits in human arm. A typical illustration of joints and Cartesian constraints appears in Fig. 3.

Two alternatives are possible to handle those constraints:

- Handle as a specific reward within the fitness function.
- Handle as separate condition with a control mechanism in FA.

It is easier to use the second scenario, since only a set of tests are needed to be added to the original IK-FA in order to make it able to handle the constraints. To use the first scenario the brightness of a solution which do not respects the constraints could be simple waved, degreased, so that no other individual of the swarm is attracted to.

Constraints could be added to the joints search space and also to the solutions search space they could be expressed as in Eq. (19) where (J1) is the expression to limit the angular displacement of a specific joint into a specific interval while (J2) is the complementary constraint.

$$J1\colon q_i(j) = \theta_j \in \left[\theta_{j\min}, \theta_{j\max}\right] \tag{19}$$

$$J2\colon q_i(j) = \theta_j \notin \left[\theta_{j\min}, \theta_{j\max}\right] \tag{20}$$

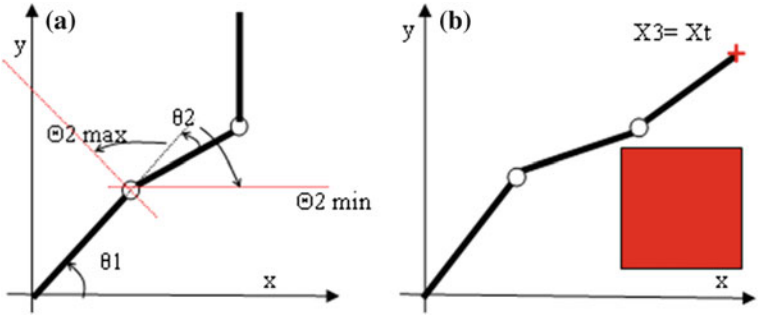


Fig. 3 Typical constraints illustration, **a** illustration of a joint constraint, **b** illustration of Cartesian space constraint

Constraints could also be expressed in the Cartesian space, so that a specific (Xi) position is within a specific convex hull. In this case the constraint is similar to J3, given by Eq. (21).

$$J3\colon X_i(j) = f(q_i) \in C(X) \tag{21}$$

C(x) is the convex hull of the solutions space in the case of Eq. (21). It is the convex hull excluded from the solution space in the case of Eq. (22).

$$J4 : X_i(j) = f(q_i) \notin C(X) \tag{22}$$

Figure 3a gives an illustration of a typical joint constraints, and an example of Cartesian space constraint. This could be expressed by (23).

$$J_4\colon X_i() = f(q_i) \notin [X_{i\min}, X_{i\max}] \tag{23}$$

In all cases, the faulty firefly is replaced by a random one with respect to needed constraints. The pseudo code of the Constrained IK-FA is presented in Fig. 4.

For the pseudo-code of Fig. 4, the fitness function is subject to minimization, as defined in (18) the brightness is maximal as the fitness approaches zero, this mean that the best firefly, will be the brightest one and will be the one with fitness as close to zero as possible. Note that it is possible to code IK-FA with a minimization

```
Begin
  Initialize IK-FA parameters
  Generate an initial population of fireflies Q =(0,⋯..xn)
  Xt = Input (target position)
  for i_t=1 to max_iteration number
          for i = 1 : n (all n fireflies)
           for j = 1 : n (n fireflies)
           if (Ii < Ij )
              move firefly i towards j
           end if
           end for j
          end for i
  Solution = best current fireflies;
  For k= 1 to max_joint number
  If  J1 = True
           Then Q ← random ([qmin, qmax])
  If  J4 = True [Xmin, Xmax]
           Then Q ← random ([qmin, qmax])
  end for k
  if (error < fixed) break
  end for i_t
     Return best firefly
End
```

Fig. 4 IK-FA with Cartesian and joint constraints

formulation directly by imposing to the brightness expression to be equal to the fitness as in (24).

$$I_j^i = f_{itness}(q_j)_i \tag{24}$$

where I_j^i the brightness of firefly (j) by iteration (i); in this case the FA procedure should be slightly adjusted so that the firefly with the higher brightness is moved toward the one with a lower brightness, since lower brightness indicates a better solution. This modification should be made on line (8) of the pseudo-code of Fig. 4.

4 Experimental Results

The experimental simulation process consisted in implementing the IK-FA solver for a 3 links planar articulated system, in a humanoid, such a system could serve as a simplified kinematic model of an arm, a leg or a finger, depending on the segments sizes and parameters. The simulations were dedicated to the performance evaluation of IK-FA, it covers the following aspects:

- The convergence capacities.
- The Impact of FA parameters on the convergence and on the quality of results.
- The impact the swarm size on the quality of the solutions and on the processing time.
- The robustness of the algorithm over the target position.

Experimental results are based on simulations using matlab software, version 7.6 (R2008a), the software runs on a personal computer with 4 Go of DRAM and T4200 processor cadenced at 2 GHz. All results are presented for 100 tests, and an estimation of the density of probability of the fitness function which is the square of the position error between the end-effector position and the target position.

4.1 The Experimental Protocol

The first test bench is a generic articulated system composed by 3 links and a 3 revolute joints, similar to what appears in Fig. 1, it represents a 3 DOF articulated system that could be used for a leg of for an arm planar model. In the case of a leg the links (l1), (l2) and (l3) represents respectively the thigh and the tibia and the foot. To apply the IK-FA, we first write the forward kinematics of that system, as in is the sytem of equations given by (25).

$$\begin{cases} x_1 = l_1 * \cos(\theta_1) \\ y_1 = l_1 * \sin(\theta_1) \\ x_2 = l_1 * \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y_2 = l_1 * \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ x_3 = l_1 * \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y_3 = l_1 * \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{cases} \tag{25}$$

For IK-FA, The inverse kinematics problem, relative to a given target position $X_t = (x_t, y_t)$ for the terminal end-segment position, could be written as follows:

$$\text{Find: } Q = [\theta_1, \theta_2, \theta_3] \text{ Satisfying } f_{p3}(q) = X_t \tag{26}$$

This formulation corresponds to a non constrained IK problem, where $f_{p_3}()$ is the forward kinematics function of system limited to the end-segment of link (3). The fitness function used for both heuristics is given by (27):

$$f_{itness}(q_j) = \left\| f_{p3}(q_j) - X_t \right\|^2 \tag{27}$$

All tests were performed with target position (0.700, −0.500). The impact of parameters is estimated based on the mean results obtained over 100 tests for each variant. A simulation of the 3 links articulated system is also produced for the best solution, such in Fig. 5a.

The couple of parameters that are used to evaluate the performances are the fitness function and the computing time which is related to the iteration number needed to converge. The fitness function used here is the square of the distance error, this consideration allowed, for some tests, to fix the position error by controlling its square instead of computing the square rough.

All tests results were visualized using a Cartesian frame such in Fig. 5a, which shows the best solution found by the end of the processing. We also systematically plot the evolution of the fitness function in order to see if a convergence behavior is observed and to evaluate the precision of the obtained solutions. A typical plot of the fitness function for a solution appears in Fig. 5b, where the fitness is plotted iteratively.

For general conclusions, the mean and the standard deviation for a set of 100 tests are used to evaluate the impact of IK-FA parameters on the results. The mean is the average of the fitness function computed for a given configuration test using the distributions fitting tool of Matlab. This tool allowed also plotting an approximation of the density of probability using a normal distribution.

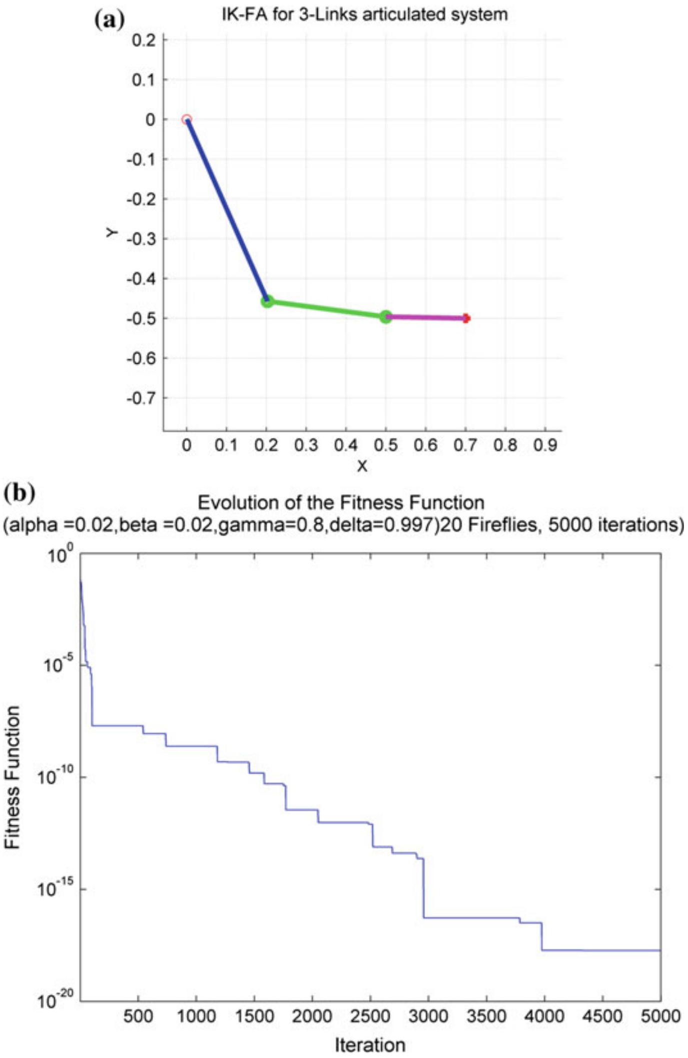


Fig. 5 IK-FA Typical solution. **a** 3 links arm with a target position (0.700, −0.500), **b** fitness function evolution for ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$) and 20 fireflies.

4.2 Performances Analysis of IK-FA

Performances analyses are based on the evaluation of the fitness function of the obtained solutions as well as the convergence time. The test protocol consist in a statistical results over 100 tests. All tests were performed with the same target position (0.700, −0.500), a section is dedicated to the effect of the target position on a specific set of good parameters of IK-FA. Discussions are conducted on the effect of the key FA algorithm parameters on convergence and performance when used in IK-FA. Statics and comparisons are made using the statistical Matlab Toolbox [19].

4.2.1 Investigation on the Possible Convergence of IK-FA

A typical set of solutions of the 3 Links system appears in Fig. 5a for ten tests. The target position is tagged with a red cross, the relative link sizes are respectively ($l1 = 0.5$, $l2 = 0.3$, $l3 = 0.2$); they are obtained by dividing the real length of the link by the length of the articulated system when all links are aligned. The target position is fixed to (0.700, −0.500). Note that IK-FA returns the best solution found by the end of its processing, the simulation results of Fig. 6, corresponds to 10 results obtained from 10 different executions of the solver with the same set of parameters. Result evaluations are based on the fitness function see Fig. 6b.

IK-FA was tested first using a set of FA parameters: ($\alpha = 0.2$, $\beta = 0.2$, $\gamma = 0.8$, $\delta = 0.9$), with this set parameters, a convergence attitude is observed with a fitness mean of about 1.14735×10^{-3}, this average was observed over a statistical test of 100 runs of IK-FA. This number of tests is necessary to measure the quality of the provided solutions. The analysis of the evolution of the fitness function for several runs show that in all cases a solution is provided within 100 iterations for this set of parameters see Fig. 6b. Some results have a high quality with fitness about 1×10^{-18}, meaning that the distance error is about 1×10^{-9}; meanwhile this kind of solutions are far from the mean performances since the worst result, obtained for this test, is a distance error of about 10^{-1}.

What could be also underlined here is the fast convergence time, since all results were reported in less then 100 iterations. Meanwhile we could not speak about a stable inverse kinematics solver due to the big range in fitness variations limits which is the square of the position error. This first test confirms the possible convergence of IK-FA to high quality solutions even if solutions with fitness under 1×10^{-16}, were only 31 over 100 tests. Solutions with fitness less than 1×10^{-6}, were 59 over 100. Week fitness's solutions, equal or higher than 1×10^{-5}, were 41 over 100 tests. This first investigation allowed confirming that it is possible to achieve a convergence using IK-FA, meanwhile deep investigations are needed to define a good set of parameters.

4.2.2 Impact of FA Parameters on Convergence

In Firefly Algorithm $\alpha \in [0, 1]$ and $\gamma \in [0, \infty)$ in theory, but in practice, it typically varies from 0.01 to 100 [10]. In this investigation a couple of IK-parameters sets are compared respectively ($\alpha = 0.2$, $\beta = 0.2$, $\gamma = 0.8$, $\delta = 0.9$) and ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$), 10 fireflies, a maximum iteration number of (1,000), for the same target point (0.700,−0,500). This simulation showed that for the first set of parameters as discussed in the previous paragraph only 59 % of the solutions had an error lower than 1×10^{-6} and the position error ranged form 10^{-1} to 10^{-8}, see Fig. 6b.

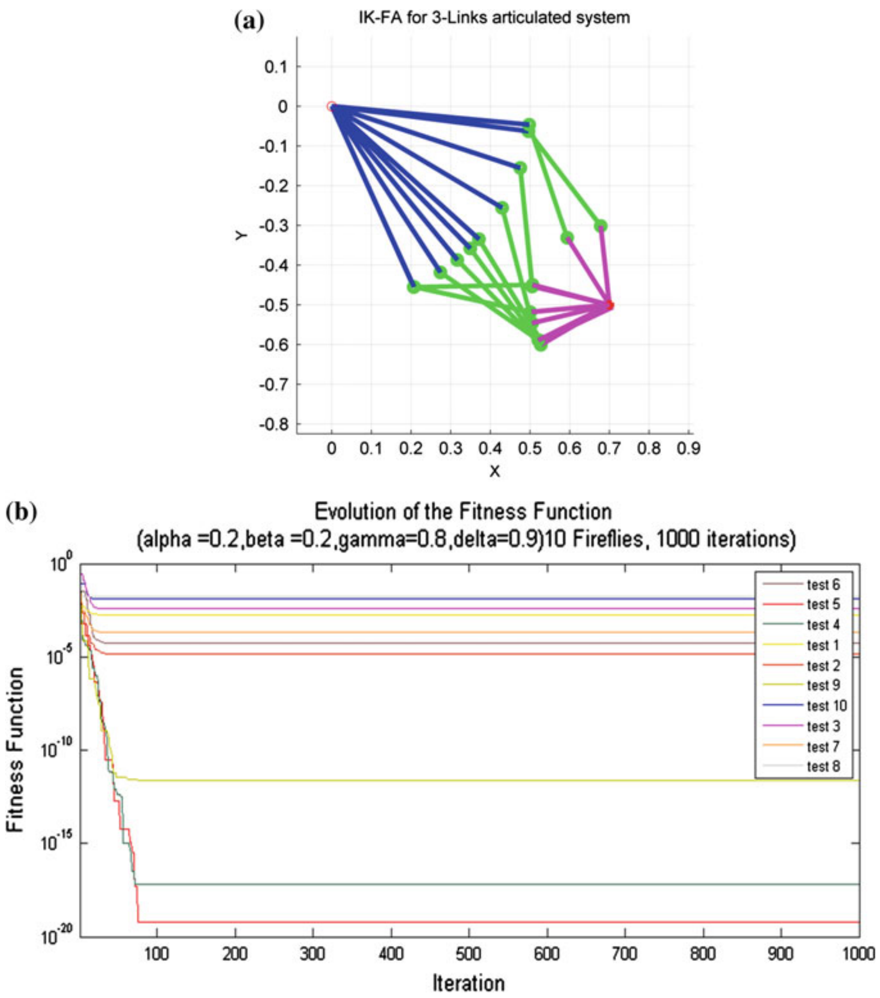


Fig. 6 IK-FA solver for a 3 links system. **a** possible solutions for a target position (0.700, −0.500), **b** evaluation of the fitness functions using (α = 0.2, β = 0.2, γ = 0.8, δ = 0.9) and 10 fireflies.

By using the set of parameters (α = 0.02, β = 0.02, γ = 0.8, δ = 0.997) we observed similar results for all the 10 tests and 100 % of the solutions where with a fitness ranging around 10^{-9} with an iteration number limited to 1,000. The evolution of the fitness function for ten tests of this configuration are displayed in Fig. 7, it showed that globally the IK-FA algorithm evolves toward decreasing its fitness function, meaning that it is evolving toward decreasing the distance to the target point. Note that this set of parameters was experimentally adjusted. For the remaining of the investigations, this set will be used.

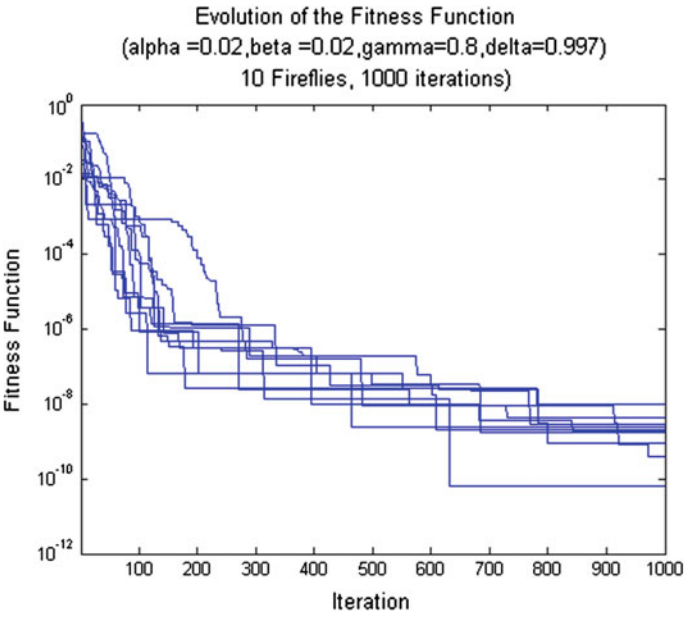


Fig. 7 Evolution of the fitness function for 10 tests, ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$) and 10 fireflies

4.2.3 Impact of the Maximum Iteration Number

A first investigation of the impact of the maximum iteration for the set of parameters, ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$), 10 fireflies, and a target position (0.7, 0.5) was done over 500, 1,000, 2,000, 3,000, 5,000 and 10,000 iterations. This experiment showed that the best fitness value decreases as the maximum iteration number increases; the best fitness for 10,000 iterations is about 1.27102×10^{-18}, and the worst result, at iteration 10,000, is 1×10^{-17}.

The fitness values observed around 500, 1,000 and 2,000 were decreasing but did not show a static fitness value, this was only achieved for 5,000 iterations, and clearly confirmed by the test of 10,000 iterations, see Fig. 8.

Using the distribution fitting tool of Matlab, the fitness is approximated with a normal distribution with an average, mean, about 1.50×10^{-17}. For 10,000 iterations a static convergence comportment is observed around 4,500, (4,489.44) iterations. These results are confirmed by 100 tests, see Fig. 9. This experiment confirm that a valuable balance consist in fixing the maximum iteration number to 5,000. The only conclusion that could be taken at this level is that for this specific set of parameters, IK-Firefly convergence is ensured with fitness around 5×10^{-17} or lower by a maximum iteration of 5,000, see Fig. 9b.

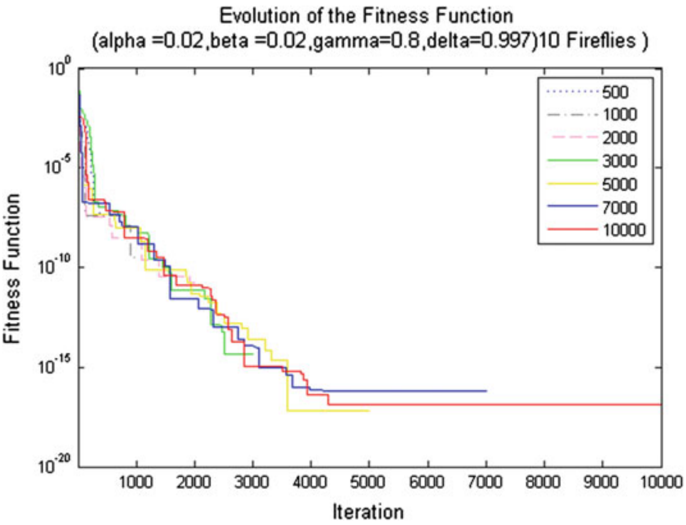


Fig. 8 Impact of the iteration number on the IK-FA convergence; Investigation for several maximum iteration numbers ranging from 500 to 10,000

4.2.4 Impact of the Swarm Size

The swarm size is the number of individuals composing the swarm. The impact of the number of fireflies is an important parameter in any swarm based heuristic; it has a direct impact on the procession time and also on the quality of the solutions. In swarm based techniques, such as PSO or GA, population sizes of 10 to 60 are commonly used [8, 38]. The investigation on the effect of the FA swarm size conducted here is specific to the IK-FA algorithm.

The number of fireflies are waved from 10 to 60 for a fixed target point (0.700, −0.500), the set of ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$) parameters and a fixed maximum iterations number = 5,000. For any given set of firefly's numbers the tests are repeated 100 times prior to any interpretations. The fitness functions are then subject to a statistical investigation using the distribution fitting tool of Matlab statistics Toolbox [19]. Interpretations are based on the mean and standard deviation values of the fitness's on the tests. The fitness corresponding to a given swarm size is approximated by a normal distribution of the density of probability function, DPF, and the mean is used to compare the impact of the swarm size, see Fig. 10.

For all swarm sizes ranging from 10 to 60 the fitness mean ranges respectively from 1.27×10^{-17} to 1.79×10^{-18}, as in Table 1, allowing to conclude that as the swarm size increases the fitness decreases and the position error which is the square root of the fitness decreases, the obtained solutions are more precise.

For a swarm size of 60 individuals, the probability to obtain a result with a fitness of 1.5e−18 is 99.8 %. For a swarm size of 10 fireflies, the mean of the normal distribution used to approximate the results is 1.27148×10^{-17}, with a

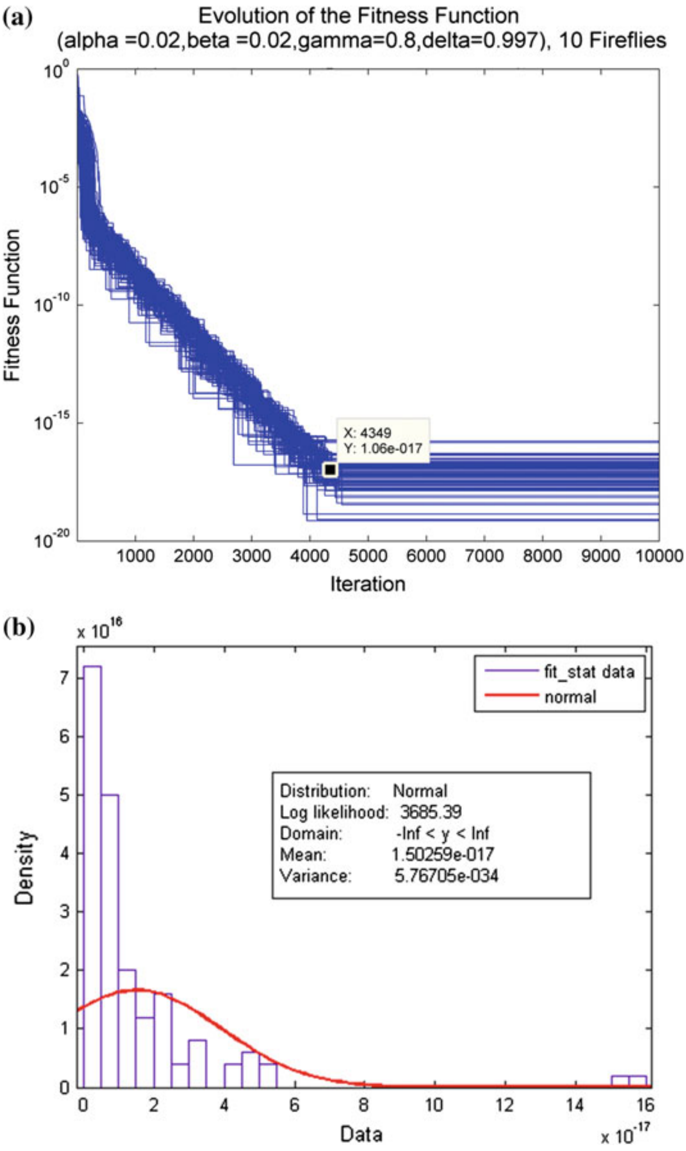


Fig. 9 Impact of the iteration number on the IK-FA convergence. **a** the evolution of the fitness function over 100 tests, **b** approximation of the fitness density of probability by a normal distribution

variance of 1.38555×10^{-34}, the probability to obtain a result with a fitness of 10^{-16} is 100 %, which could be considered as a proof of convergence of the IK-FA algorithm. Note also that results for 50 fireflies are very close to those of 60 fireflies, see Fig. 7, where the yellow distribution represents the results for 50 fireflies its

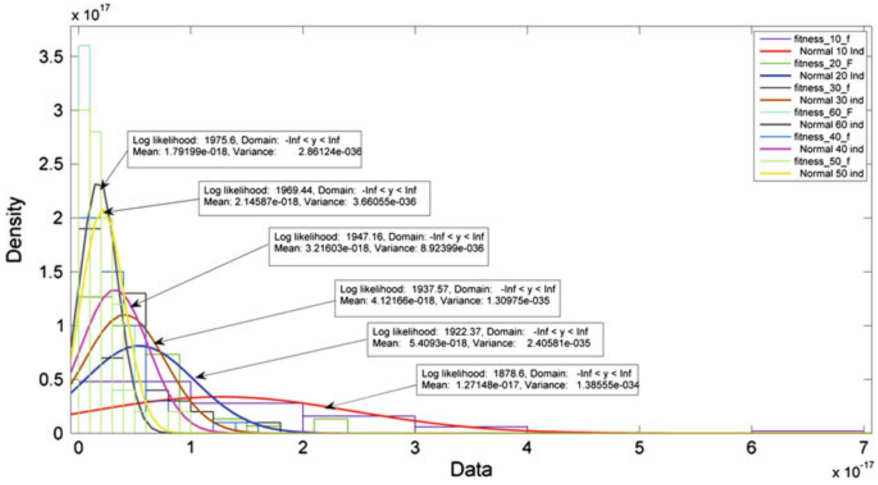


Fig. 10 Impact of the swarm size

Table 1 Computing time with maximum iteration stop condition at 5,000

Swarm size	Mean fitness function	Variance of fitness function
10	1.2714×10^{-17}	1.3855×10^{-34}
20	5.4093×10^{-18}	2.4058×10^{-35}
30	4.1216×10^{-18}	1.3097×10^{-35}
40	3.2146×10^{-18}	8.9239×10^{-36}
50	2.1458×10^{-18}	3.6605×10^{-36}
60	1.7891×10^{-18}	2.8612×10^{-36}

mean is 2.145×10^{-18} with a variance of 3.660×10^{-36}. Results for 40, 30 and 20 fireflies are also close with respective fitness means of 3.2146×10^{-18}, 4.1216×10^{-18} and 5.4093×10^{-18}. Results are resumed in Table 1.

Globally we can deduce that for a swarm size of 40–60 the fitness mean is ranged from $3.21*10^{-18}$ to 1.79×10^{-18} respectively with a variance ranging from 8.92×10^{-36} to 2.86×10^{-36}, for swarm's size of 10 fireflies the fitness is 10 times lower, with a mean of 1.27×10^{-17} and a variance of 1.34×10^{-34}.

Results comparison based on normalized distributions on the fitness functions' over 100 tests, showed that as the swarm size increased the variance of the fitness functions decreased, best results are obtained with 60 fireflies, see Fig. 8, meanwhile results with 50 and 40 fireflies are very close, see Fig. 10. Known the impact of the swarm size on the processing a good balance between fitness, swarm size and processing time is the next investigation issue.

Table 2 Computing time with maximum iteration stop condition at 5,000

Computing time (s)	Swarm size
$1.215262075163430 \times 10^{-3}$	10
$1.755538920139096 \times 10^{-3}$	20
$2.733618017054081 \times 10^{-3}$	30
$4.241389728678068 \times 10^{-3}$	40
$5.328678837890005 \times 10^{-3}$	50
$7.154939660265453 \times 10^{-3}$	60

4.2.5 IK-FA Computing Time

The next investigation concerns the impact of swarm size on the computing time; results reported on Table 2, concern the time needed for a fixed maximum iteration number of 5,000 iterations. In Table 3, the impact of the population size on computing time for a given error position, The IK-FA stop condition is modified so that it ends treatments when the position error is less or equal to 10^{-6}, meaning that the fitness function is less or equal to 10^{-12}. If the error position is not achieved the algorithm will stop at its maximum iteration fixed to 5,000 as in the previous test. The time values presented in Table 2 are average time observed over 100 tests.

Crossing the impact of the number of the population size and the computing time, it appeared that a swarm size of 20 individual is a valuable choice, since it allowed achieving a fitness function of 5.48×10^{-18} in a computing time relatively close to what we can obtain with a limited swarm size of 10 individuals. This choice is confirmed when the stop condition is modified so that the swarm stops when it achieved a desired fitness, by mean of error, details of this experimentation appears in Table 3.

4.2.6 Robustness Over the Target Point Position

In order the check the robustness of the results over the target position, 100 tests were performed with a randomly generated target position at each attempt. The test configuration parameters are: ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$), a swarm size of 20 individual's and a maximum iteration number of 5,000. The random target positions are generated within a circle of radius (1), as in Fig. 11a. The fitness of each solution is returned and subject to a statistical analysis using the distribution fitting Matlab tool.

Statistics analysis showed that the probability to obtain a solution with a fitness lower that 3×10^{-17} is 100 %, this means that for any random target position IK-FA will generate at 100 % a solution with a fitness lower that 3×10^{-17}. We can conclude that that for any target position within the definition space of the system, here a circle of radius (1), we are sure that an inverse kinematics solution exists and we are also sure at 100 % that this solution has a position error of 5.4722×10^{-9} as in Fig. 11b.

Table 3 Computing time with a stop condition (fitness = or <1e−12)

Computing time (s)	Swarm size	Iteration of convergence
6.7354192×10^{-4}	10	2,226 (min 1,711, max 2,569)
$7.63017534 \times 10^{-4}$	20	2,399 (min 2,300, max 2,661)
$1.46698068 \times 10^{-3}$	30	2,400 (min 2,110, max 2,605)
$2.10200931 \times 10^{-3}$	40	2,377 (min 2,100, max 2,535)
$2.54710426 \times 10^{-3}$	50	2,390 (min 2,230, max 2,546)
$3.50012342 \times 10^{-3}$	60	2,453 (min 2,175, max 2,569)

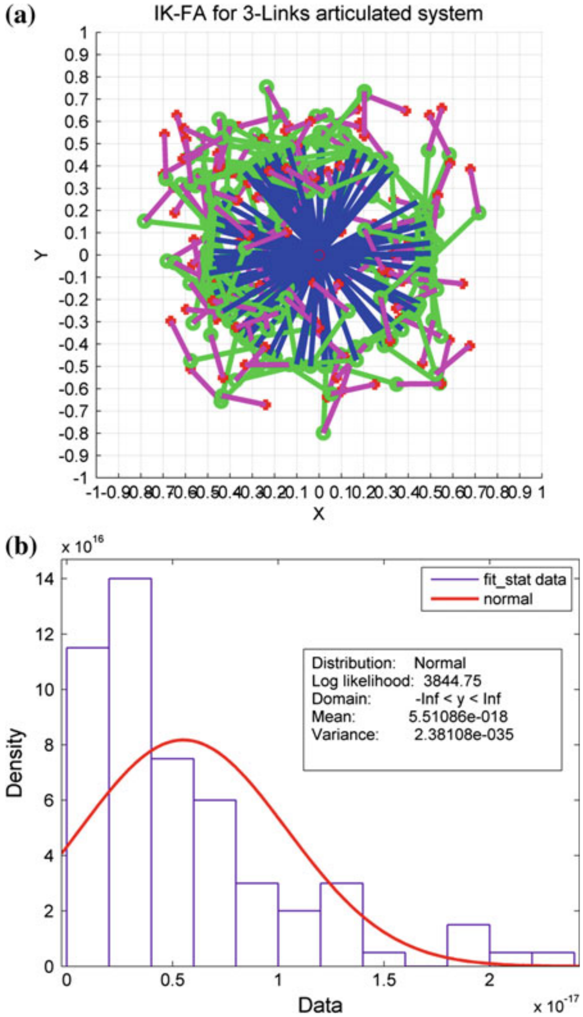


Fig. 11 IK-FA with random target positions. **a** Plot of solutions with random target. **b** Density of probability distribution of the fitness function

Table 4 Computing time of IK-FA compared to CCD for similar conditions

IK method	Position error	Computing time (s)
IK-FA	10^{-4}	2.1918973×10^{-3}
CCD	10^{-4}	0.4329001
IK-FA	10^{-8}	2.3394578×10^{-3}
CCD	10^{-8}	0.7112541

4.3 Comparison with CCD Method

After investigating the key aspects of IK-PSO, a comparative test with CCD inverse kinematics method is conducted for a 3 links articulated system with three revolute joints as in Fig. 1. The test is conducted under the same conditions using IK-FA and the CCD method. CCD was selected because it is a reference method for inverse kinematics solver and is assumed to be a real time IK method, real time methods are time stressed and have a large panel of application [37]. For this test both algorithms were asked to solve the inverse kinematics problem for the target point xt (0.700, −0.500), for a given position error.

Results showed that for both configurations the IK-FA is faster than CCD , these results were established for 10^{-4} and 10^{-8} error positions, meaning that IK-FA fitness were respectively of 10^{-8} and 10^{-16}. Test were conducted using unconstrained IK-FA variant. Results showed that IK-FA clearly rendered solutions in a limited time compared to CCD, see Table 4 for details.

5 Conclusions and Future Directions

In this paper a new heuristic method for inverse kinematics based on Firefly Algorithm is proposed, IK-FA. It is based on the firefly algorithm and the forward kinematics model of a robotic system. The method is proposed for a human like articulated system, HLAS, while it could generalized to any kind of robotics. The paper focuses on the IK-FA convergence capacities as well as the impact of FA parameters on the quality of the solutions. A set of good parameters for IK-FA was also established.

As conclusions: IK-FA, is a valuable solver for inverse kinematics, while parameter fitting is still a challenging problem. For a Given set of parameters, the heuristic converges to a static fitness value within a fix maximum number of iterations, in this work about 4,500 for ($\alpha = 0.02$, $\beta = 0.02$, $\gamma = 0.8$, $\delta = 0.997$). IK-FA has a fair convergence time, for the tested configurations, the average was about (2.3394×10^{-3}) s with a position error around (3.116e−08) over 100 tests. The algorithm showed also evidence of robustness over the target position, since for all conducted tests with a randomly generated target positions, IK-FA achieved a solution with a position error lower or equal to 5.4722×10^{-9}.

The investigation of the impact of the swarm size, showed that whatever is the swarm size, form 10 to 60, IK-FA convergences. Meanwhile is has been established in this work that as the swarm size increases the variance of the obtained solutions decreases. This means that the probability of finding a solution closer to the mean is higher. When the swarm size increases the computing time do so. A balance, between swarm size and computing time, need to be defined, in this work 20 FA individuals is an interesting choice.

Further developments are needed to deeply investigate the impact of FA variants on IK-FA. The implementation of IK-FA as an inverse kinematics solver of a robotic system such in [29, 31] should be introduced soon.

In This paper IK-FA was introduced as new heuristic inverse kinematics solver for constrained and unconstrained problems. The experimental investigations were limited to the unconstrained variant with an application to an articulated system composed by 3 links and 3 revolute joints. The impact of constraints on performances and computing time are under developments.

Acknowledgment The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

1. Ammar, B., Chouikhi, N., Alimi, A.M., Chérif, F., Rezzoug, N., Gorce, P.: Learning to walk using a recurrent neural network with time delay. In: Artificial Neural Networks and Machine Learning–ICANN, pp. 511–518. Springer, Heidelberg (2013)
2. Asfour, T., Dillmann, R.: Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In: Intelligent Robots and Systems (IROS 2003), vol. 2, pp. 1407–1412 (2003)
3. Azevedo, C., Andreff, N., Arias, S.: BIPedal walking: from gait design to experimental analysis. Mechatronics **14**(6), 639–665 (2004)
4. Buckley, K.A., Simon H., Brian C.H.T.: Solution of inverse kinematics problems of a highly kinematically redundant manipulator using genetic algorithms. IET, pp. 264–269 (1997)
5. Buss, S.R.: Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. IEEE J. Robot. Autom. **17** (2004)
6. Çavdar, T., Mohammad, M., Milani, R.A.: A new heuristic approach for inverse kinematics of robot arms. Adv. Sci. Lett. **19**(1), 329–333 (2013)
7. Chiaverini, S., Siciliano, B., Egeland, O.: Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. IEEE Trans. Control Syst. Technol. **2**(2), 123–134 (1994)
8. De Jong, K.A., Spears, W.M.: An analysis of the interacting roles of population size and crossover in genetic algorithms. In: Parallel Problem Solving from Nature, pp. 38–47. Springer, Heidelberg (1991)
9. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1** (4), 28–39 (2006)
10. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, vol. 1, pp. 39–43 (1995)
11. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 81–86 (2001)

12. Edison, E., Shima, T.: Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. Comput. Oper. Res. **38**(1), 340–356 (2011)
13. Juang, J.G.: Fuzzy neural network approaches for robotic gait synthesis. IEEE Trans. Syst. Man Cybern. B Cybern. **30**(4), 594–601 (2000)
14. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. 1–37 (2012)
15. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots. In: Robotics Research, pp. 365–374. Springer, Heidelberg (2005)
16. Kulpa, R., Multon, F.: Fast inverse kinematics and kinetics solver for human-like figures. In: Proceedings of Humanoids, pp. 38–43 (2005)
17. Lander, J., CONTENT, G.: Making kine more flexible. Game Developer Mag. **1**, 15–22 (1998)
18. Łukasik, S., Żak, S.: Firefly algorithm for continuous constrained optimization tasks. Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, pp. 97–106. Springer, Heidelberg (2009)
19. MATLAB Statistics Toolbox User's Guide (2014). The MathWorks Inc. http:www.mathworks.com/help/pdf_doc/stats/stats.pdf
20. Mohamad, M.M., Taylor, N.K., Dunnigan, M.W.: Articulated robot motion planning using ant colony optimisation. In: 3rd International IEEE Conference on Intelligent Systems, pp. 690–695 (2006)
21. Pant, M., Gupta, H., Narayan, G.: Genetic algorithms: a review. In: National conference on frontiers in applied and computational mathematics (FACM-2005), Allied Publishers, p. 225, 04–05 Mar 2005
22. Pérez-Rodríguez, R., Marcano-Cedeño, A., Costa, Ú., Solana, J., Cáceres, C., Opisso, E., Gómez, E.J.: Inverse kinematics of a 6 DoF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical neurorehabilitation. Expert Syst. Appl. **39**(10), 9612–9622 (2012)
23. Pham, D.T., Castellani, M., and Le Thi, H.A.: Nature-inspired intelligent optimisation using the bees algorithm. In: Transactions on Computational Intelligence XIII, pp. 38–69. Springer, Heidelberg (2014)
24. Pollard, N.S., Hodgins, J.K., Riley, M.J., Atkeson, C.G.: Adapting human motion for the control of a humanoid robot. In Proceedings of IEEE International Conference on Robotics and Automation, ICRA'02, vol. 2, pp. 1390–1397 (2002)
25. Rokbani, N., Alimi, A.M.: Inverse kinematics using particle swarm optimization, a statistical analysis. Procedia Eng. **64**, 1602–1611 (2013)
26. Rokbani, N., Alimi, A.M.: IK-PSO, PSO inverse kinematics solver with application to biped gait generation. Int. J. Comput. Appl. **58**(22), 33–39 (2012)
27. Rokbani, N., Alimi, A.M., Cherif, B.A.: Architectural proposal for an intelligent humanoid. In: Procedings of IEEE Conference on Automation and Logistics (2007)
28. Rokbani, N., Benbousaada, E., Ammar, B., Alimi, A.M.: Biped robot control using particle swarm optimization. In: IEEE International Conference Systems on Man and Cybernetics (SMC), pp. 506–512 (2010)
29. Rokbani, N., Boussada, E.B., Cherif, B.A., Alimi, A.M.: From gaits to ROBOT, A Hybrid methodology for A biped Walker. Mobile Robotics: Solutions and Challenges. In: Proceedings of Clawar, vol. 12, pp. 685–692 (2009)
30. Rokbani, N., Cherif B.A., Alimi, A.M.: Toward intelligent biped-humanoids gaits generation. In: Choi, B. (eds.) Humanoids. Chap 14, InTech (2009)
31. Rokbani, N., Zaidi, A., Alimi, A.M.: Prototyping a biped robot using an educational robotics kit. In: IEEE International Conference on Education and E-learning Innovations. Sousse, Tunisia (2012)
32. Rutkowski, L., Przybyl, A., Cpalka, K.: Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation. IEEE Trans. Industr. Electron. **59**(2), 1238–1247 (2012)

33. Schmidt, V., Müller, B., Pott, A. Solving the forward kinematics of cable-driven parallel robots with neural networks and interval arithmetic. In: Computational Kinematics, pp. 103–110. Springer, Netherlands (2014)
34. Tchoń, K., Jakubiak, J.: Endogenous configuration space approach to mobile manipulators: a derivation and performance assessment of Jacobian inverse kinematics algorithms. Int. J. Control **76**(14), 1387–1419 (2003)
35. Tchon, K., Jakubiak, J.: Jacobian inverse kinematics. In: Advances in Robot Kinematics: Mechanisms and Motion, p. 465 (2006)
36. Tevatia, G., Schaal, S.: Inverse kinematics for humanoid robots. In: Proceedings of IEEE International Conference on Robotics and Automation, (ICRA'00), pp. 294–299 (2000)
37. Tolani, D., Goswami, A., Badler, N.I.: Real-time inverse kinematics techniques for anthropomorphic limbs. Graph. Models **62**(5), 353–388 (2000)
38. Van den Bergh, F., Engelbrecht, A.P.: Effects of swarm size on cooperative particle swarm optimizers (2001)
39. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. In: ACM Transactions on Graphics (TOG), ACM, vol. 28, No. 3, p. 63 (2009)
40. Xu, Q., Li, Y.: Error analysis and optimal design of a class of translational parallel kinematic machine using particle swarm optimization. Robotica **27**(1), 67–78 (2009)
41. Yang, X.S. (2010). Firefly algorithm, Levy flights and global optimization. In: Research and Development in Intelligent Systems XXVI. Springer, London, 209–218
42. Yang, X.S.: Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio-Inspired Comput. **2**(2), 78–84 (2010)
43. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Stochastic algorithms: foundations and applications, pp. 169–178, Springer, Heidelberg (2009)
44. Zaidi, A., Rokbani, N., Alimi, A.M.: A hierarchical fuzzy controller for a biped robot. In: Proceedings of ICBR 2013. Sousse, Tunisia (2013)
45. Zaidi, A., Rokbani, N., Alimi, A.M.: Neuro-Fuzzy gait generator for a biped robot. J. Electron. Syst. **2**(2), 48–54 (2012)
46. Zhang, X,.Nelson, C.A.: Multiple-criteria kinematic optimization for the design of spherical serial mechanisms using genetic algorithms. J. Mech. Des. **133**(1) (2011)

Computer Aided Intelligent Breast Cancer Detection: Second Opinion for Radiologists—A Prospective Study

J. Dheeba and N. Albert Singh

Abstract Breast cancer is the common form of cancer and leading cause of mortality among woman, especially in developed countries. In western countries about 53–92 % of the population has this disease. As with any form of cancer, early detection and diagnosis of breast cancer can increase the survival rate. Mammography is the current diagnostic method for early detection of breast cancer. Breast parenchymal patterns are not stable between patients, between left and right breasts, and even within the same breast from year to year in the same patient. Breast cancer has a varied appearance on mammograms, from the obvious spiculated masses, to very subtle asymmetries noted on only one view, to faint calcifications seen only with full digital resolution or a magnifying glass. The large volume of cases requiring interpretation in many practices is also daunting, given the number of women in the population for whom yearly screening mammography is recommended. It seems obvious that this difficult task could likely be made less error prone with the help of computer algorithms. Computer-aided detection (CAD) systems have been shown to be capable of reducing false-negative rates in the detection of breast cancer by highlighting suspicious masses and microcalcifications on mammograms. These systems aid the radiologist as a 'second opinion' in detecting cancers and the final decision is taken by the radiologist. A supervised machine learning algorithm is investigated—Differential Evolution Optimized Wavelet Neural Network (DEOWNN) for detection of abnormalities in mammograms. Differential Evolution (DE) is a population based optimization algorithm based on the principle of natural evolution, which optimizes real parameters and real valued functions. By utilizing the DE algorithm, the parameters of the Wavelet Neural Network (WNN) are optimized. To increase the detection accuracy a feature extraction methodology is used to extract the texture based features of the abnormal

---

J. Dheeba (✉)
Department of Computer Science and Engineering, Noorul Islam University,
Kumaracoil, Tamil Nadu, India
e-mail: deeps_3u4@yahoo.com

N. Albert Singh
BSNL, Nagercoil, Tamil Nadu, India
e-mail: albertsingh@rediffmail.com

© Springer International Publishing Switzerland 2015

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_16

breast tissues prior to classification. Then differential evolution optimized wavelet neural network classifier is applied at the end to determine whether the given input data is normal or abnormal. The performance of the computerized decision support system is evaluated using a mini database from Mammographic Image Analysis Society (MIAS) and images collected from mammogram screening centres.

Keywords Breast cancer · Mammograms · Differential evolution · Wavelet neural network

1 Introduction

Cancers are abnormal growth of cancer cells in the body. The cell growth in the body regulates and replaces the old cells by new cells and the old cells die. But in certain conditions the cell divides in an abnormal way, producing more cells just like it and forming a tumor. Cancer cells often travel to other parts of the body, where they begin to grow and form new tumors that replace normal tissue. This process is called metastasis. It happens when the cancer cells get into the bloodstream or lymph vessels of the body.

The estimated number of new cases each year is expected to rise from 10 million in 2002 to 15 million by 2025, with 60 % of those cases occurring in developing countries. Not all tumors are cancerous. Tumors that aren't cancer are called benign. Benign tumors can cause problems—they can grow very large and press on healthy organs and tissues. But they cannot grow into (invade) other tissues. These tumors are almost never life threatening. Malignant tumors are cancerous. Left unchecked, malignant cells eventually can spread beyond the original tumor to other parts of the body.

Among all cancers Breast cancer remains a leading cause of cancer deaths among women in many parts of the world. Worldwide, breast cancer comprises 22.9 % of all cancers in women in developed countries [1]. Breast cancer starts in the tissues of the breast that produce milk and in the cell lining of the small milk ducts. Breast cancer may be invasive during which cancer cells spreads from the milk duct or lobule to other tissues in the breast and noninvasive, in which the cancer cells remains within the ductal system. Breast cancer is a malignant tumor that starts in the cells of the breast. A malignant tumor is a group of cancer cells that can grow into (invade) surrounding tissues or spread to distant areas of the body. The disease occurs almost entirely in women, but men can get it, too [2].

To understand breast cancer, it helps to have some basic knowledge about the normal structure of the breast. The female breast is made up mainly of lobules (milk-producing glands), ducts (tiny tubes that carry the milk from the lobules to the nipple), and stroma (fatty tissue and connective tissue surrounding the ducts and lobules, blood vessels, and lymphatic vessels). Usually breast cancer either begins in the cells of the lobules, which are the milk-producing glands, or the ducts, the passages that drain milk from the lobules to the nipple.

Over time, cancer cells can invade nearby healthy breast tissue and make their way into the underarm lymph nodes, small organs that filter out foreign substances in the body. If cancer cells get into the lymph nodes, they then have a pathway into other parts of the body [3]. The more lymph nodes with breast cancer cells, the more likely it is that the cancer may be found in other organs as well. Because of this, finding cancer in one or more lymph nodes often affects the treatment plan.

1.1 Benign Stage Breast Lumps

The benign stage breast lumps include fibrocystic changes, cysts, fibro adenomas, infections and trauma.

Fibrocystic Changes
Fibrosis is the formation of scar-like (fibrous) tissue. Fibrocystic changes are any lumpiness, thickening or swelling in the women's breast.

Cysts
Cysts are fluid filled lumps that can range from very tiny to about the size of an egg.

Fibro Adenomas Benign breast tumors such as fibroadenomas are abnormal growths, but they are not cancerous and do not spread outside the breast to other organs. They are not life threatening. Fibro adenomas are a solid round rubbery lump that moves under the skin when touched, and occurs mostly in younger women.

Infections and Trauma
Infections and trauma are red lumpiness in the breast skin due to bruises.

1.2 Types of Malignant Breast Cancer

Breast cancers can be, Noninvasive Breast cancer or Invasive Breast Cancer.

Noninvasive breast cancer. The Noninvasive (in situ) breast cancer is a type of cancer than does not invade the nearby cell. Instead, it stays in a part of the breast where it forms. One type of noninvasive cancer called ductal carcinoma in situ (DCIS) is considered a precancerous lesion. This means that if it were left in the body, DCIS could eventually develop into an invasive cancer.

Invasive breast cancer, this type of breast cancer invades the nearby tissues of the breast and spreads to the other parts of the body. The cancer cells can then travel to other parts of the body, such as the lymph nodes. If the breast cancer is stage I, II, III or IV, then its an invasive breast cancer.

The cause of breast cancer is not understood and there is no immediate hope of prevention. Advances in surgery, radiotherapy, chemotherapy, and hormone therapy have achieved only small increases in survival. One reason for this is that effective treatment is related to the stage at which the disease is detected and treated. Early detection will not prevent breast cancer, but it can help find it when the likelihood of successful treatment is greatest. In general, the earlier the detection and treatment the better the chance of survival. Early breast cancer usually does not cause symptoms. Prognosis and survival rate varies greatly depending on cancer type, staging and treatment.

1.3 Breast Cancer Screening Method

Breast cancer screening includes tests to detect breast cancer at an early stage, before a woman discovers a lump. The chance of dying from breast cancer has declined by about a third over the past few decades. Screening refers to tests and exams used to find a disease, like cancer, in people who do not have any symptoms. The goal of screening exams, such as mammograms, is to find cancers before they start to cause symptoms. Breast cancers that are found because they can be felt tend to be larger and are more likely to have already spread beyond the breast. In contrast, breast cancers found during screening exams are more likely to be small and still confined to the breast. The size of a breast cancer and how far it has spread are important factors in predicting the prognosis for a woman with this disease.

Most doctors feel that early detection tests for breast cancer save many thousands of lives each year, and that many more lives could be saved if even more women and their health care providers took advantage of these tests. Following the American Cancer Society's guidelines for the early detection of breast cancer improves the chances that breast cancer can be diagnosed at an early stage and treated successfully.

Mammography Mammography plays a major role in early detection of breast cancers, detecting about 75 % of cancers at least a year before they can be felt. It is estimated that 48 million mammograms are performed each year in US. Mammography is a special type of x-ray imaging used to create detailed images of the breast. An illustration of the digital mammogram image is shown in Fig. 1. The arrow mark shows the possible abnormality in the mammogram image.

There are two types of mammography examinations namely screening and diagnostic mammogram. Screening mammography is done in asymptomatic women. Early detection of small breast cancers by screening mammography greatly improves a woman's chances for successful treatment. Screening mammography is recommended every 1–2 years for women once they reach 40 years of age and every year once they reach 50 years of age. Diagnostic mammography is performed in symptomatic women, for example when a breast lump or nipple discharge is

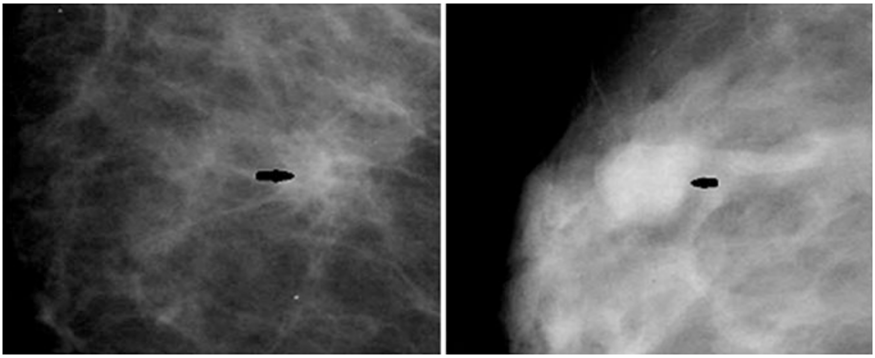


Fig. 1 Digital mammogram image having abnormality

found during self-examination or an abnormality is found during screening mammography. Diagnostic mammography is more involved and time-consuming than screening mammography and is used to determine exact size and location of breast abnormalities and to image the surrounding tissue and lymph nodes.

Digital mammography helps spotting early signs of cancer than film mammography. Having digital files, rather than stacks of film, will let hospitals and mammography clinics store and transfer information more easily. Laurel [4] says “in my mind breast cancer detection is one of the most important and most challenging engineering problems”.

1.4 Breast Cancer Signs in Digital Mammograms

Breast abnormalities that can indicate breast cancer are **masses, calcifications, architectural distortion**. Two main types of feature need to be recognized in mammograms: soft tissue masses of the order of 1 cm in diameter and usually with only very subtle differences in density to the surrounding normal structures, and small (of the order of 0.lmm) irregularly shaped microcalcifications which can be associated with malignant disease.

A mass is defined as a space occupying lesion seen in at least two different projections. If a potential mass is seen in only a single projection it should be called ‘Asymmetry’ or ‘Asymmetric Density’ until its three-dimensionality is confirmed. Masses have different density, different margins and different shape [5]. Benign masses generally cause no skin change and are smooth, soft to firm, and mobile, with well-defined margins. Diffuse, symmetric thickening, which is common in the upper outer quadrants, may indicate fibro-cystic changes. Malignant masses generally are hard, immobile, and fixed to surrounding skin and soft tissue, with poorly defined or irregular margins. However, mobile or nonfixed masses can be cancerous.

Architectural distortion is defined as distortion of the normal architecture with no definite mass visible. Architectural distortion of breast tissue can indicate malignant changes especially when integrated with visible lesions such as mass, asymmetry or calcifications. Architectural distortion can be classified as benign when including scar and soft-tissue damage due to trauma [6, 7].

Bilateral asymmetries of concern are those that are changing or enlarging or new, those that are palpable and those that are associated with other findings, such as microcalcifications or architectural distortion [8, 9]. If a palpable thickening or mass corresponds to an asymmetric density, the density is regarded with a greater degree of suspicion for malignancy.

Microcalcifications are quite tiny bits of calcium, and may show up in clusters or in patterns (like circles or lines) and are associated with extra cell activity in breast tissue. Benign calcifications are usually larger and coarser with round and smooth contours. Malignant calcifications tend to be numerous, clustered, small, varying in size and shape, angular, irregularly shaped and branching in orientation [10, 11].

In many cases, the microcalcifications and the cancer masses are hidden in the intense breast tissues especially in younger women, making both the diagnosis and detection more complex and intricate [12]. While mammography has been proven to be a powerful tool in the fight against breast cancer, the accurate reading of mammograms can sometimes be difficult [13]. Even the most trained radiologist can miss subtle variations in tissue that might be of concern.

As screening mammography is also associated with potential harms, detecting abnormalities in mammograms is a challenging task. One among them is the false-positive results occurring when radiologists decide mammograms are abnormal but no cancer is actually present. All abnormal mammograms should be followed up with additional testing i.e. a biopsy to determine whether cancer is present. False-positive mammogram results can lead to anxiety and other forms of psychological distress in affected women. The additional testing required to rule out cancer can also be costly and time consuming and can cause physical discomfort.

False-negative results occur when mammograms appear normal even though breast cancer is present. Overall, screening mammograms miss up to 20 percent of breast cancers that are present at the time of screening. False-negative results occur more often among younger women than among older women because younger women are more likely to have dense breasts. As a woman ages, her breasts usually become more fatty, and false-negative results become less likely. False-negative results can lead to delays in treatment and a false sense of security for affected women.

An automated system can overcome these problems by reducing the number of false positive and false negative readings from radiologists and increase the chance of detecting abnormalities early. This is a favorable prognosis for patients, as incorrect or late detections often result in mortality. As with many labor-intensive occupations, radiologists use computer-aided detection (CAD) systems that can identify potential cancers on mammography images. Recent studies have also shown that CAD systems, when used as an aid, have improved radiologists' accuracy of detection of breast cancer [3].

These software systems are entering clinical practice as a way to improve radiologists' ability to detect the few cancer cases in the sea of normal-looking images they see every day. In a typical screening situation, a mammographer will uncover only 1.5–6 cancers for every 1,000 films he or she reads. Some cancers are inevitably overlooked, but having two radiologists looking at each film has been shown to improve the detection rate by as much as 15 %, according to studies cited in the Institute of Medicine report [13]. Computer-aided detection systems evaluate the conspicuous structures and helps in reducing the false positives and false negatives [14,15]. This allows the radiologist to draw conclusions about the condition of the pathology.

The studies indicate the importance in analyzing the problem and efforts done to improve the performance of the cancer detection in digital mammograms. Researchers are responsible to conceive new and improved analytical tools to solve a problem. When a new tool is available the problem should be reexamined to find better and more accurate solutions. In recent years, Soft Computing and Intelligent algorithms are gaining more importance and giving promising results in medical applications [16, 17]. These issues motivate in applying intelligent and soft computing paradigms for analyzing and improving the performance of detection and classification of abnormal and normal tissues.

This chapter is organized as follows: Sect. 2 describes the related works. Database description is explained in Sect. 3. Section 4 presents the methods used in this chapter. The preliminary of wavelet neural network and differential evolution algorithm is explained in Sect. 5. Section 6 explains how WNN is evolved using differential evolution algorithm. Section 7 deals with experimental work and analysis of results. Performance analysis and conclusion is presented in Sects. 8 and 9 respectively.

2 Related Works

A lot of researches in the area of CAD systems for breast cancer and developing intelligent techniques for improving classification accuracy have been conducted in last few decades [2, 18, 19]. Different studies have demonstrated that Computer Aided Detection (CAD) of breast cancer can improve the detection rate from 4.7 to 19.5 % compared to radiologists. Regarding classification of abnormalities in mammogram, a number of techniques have been presented using machine learning approaches to classify samples as normal and abnormal.

Anna et al. [20] investigated multi-scale texture properties of the tissue surrounding microcalcifications (MCs) for breast cancer diagnosis using probabilistic neural network. Azar and El-Said [21] used a probabilistic neural network for breast cancer classification. Gray-level first order statistics, gray-level co-occurrence matrices features, and Laws texture energy measures are extracted from original image. The classifying power of these features is analyzed using probabilistic neural network; achieving overall accuracy of 90 % when using laws texture features in the classification of 85 mammogram images.

Mudigonda et al. [22] proposed a segmentation method for finding the suspected mass regions in mammograms. Li et al. [23] presented a statistical model supported approach for enhanced segmentation and extraction of suspicious mass areas from mammographic images. Gao et al. [24] used a preprocessing technique to improve the mass detection. Mudigonda et al. [25] used a gradient and texture based features to detect malignant masses in mammograms. Suliga et al. [26], propose a Markov random field (MRF) based technique that is suitable for performing clustering in an environment which is described by limited data. Grim et al. [27] demonstrated a preprocessing model based on local statistical texture for screening mammograms. Preprocessing is done to emphasize diagnostically important details of suspicious regions in mammograms. A log-likelihood image is found to take information to identify the malignant tumor locations.

Yu and Huang [19] used a wavelet filter to detect all the suspicious regions using the mean pixel value. Heine et al. [28], proposed a multiresolution statistical method for identifying clinically normal tissue in digitized mammograms. Kupinski and Giger [29] presented a radial gradient index based algorithm and a probabilistic algorithm for detecting lesions in digital mammograms. Nakayama et al. [30] proposed a MC detection mechanism based on the shape features. Cheng et al. [31], proposed a novel approach for detecting the microcalcification clusters in arbitrary shape and in the mammograms of the breasts with various densities. Wang and Karayiannis [32] presented an approach based on the wavelet features for detection of microcalcification in mammograms.

Eltoukhy et al. [33], propose to construct and evaluate a supervised classifier for mammograms using a multiscale curvelet transform coefficients. Verma et al. [34] investigated a novel soft clustered based direct learning classifier which creates soft clusters within a class and learns using direct calculation of weights. Teo et al. [35], used the early time backscatter response obtained using an ultra wide band radar system is shown to have the potential for lesion classification. A rough lesion with multiple spicules has more significant scattering points than a lesion with compact shape. Peng et al. [36], presented a novel algorithm for the detection of microcalcifications using stochastic resonance (SR) noise. Tsui et al. [2], a novel method of 2-D analysis based on describing the contour using the B-mode image and the scatterer properties using the Nakagami image, which may provide useful clues for classifying benign and malignant tumors.

Among existing CAD techniques, the main problem of developing an acceptable CAD system is inconsistent and low classification accuracy. In order to improve the training process and accuracy, this chapter investigates novel intelligent classifiers that use texture information as input to classify the normal and abnormal tissues in mammograms. Moreover, the intelligent machine learning classifiers are optimized using heuristic algorithms for finding appropriate hidden neurons, learning rate and momentum constant during the training process.

3 Database Description

Mammograms were collected from the Mammographic Image Analysis Society (MIAS) is an organisation of UK research groups interested in the understanding of mammograms and has generated a database of digital mammograms developed by Suckling and Parker [37]. Films taken from the UK National Breast Screening Programme have been digitised to 50 micron pixel edge. The database contains 322 digitised films. It also includes radiologist's "truth" markings on the locations of any abnormalities that may be present. Each digitized mammograms was incorporated into a 1,024 × 1,024 pixel image. The database contains 208 normal, 63 benign and 51 malignant (abnormal) images. The database is concluding of four different kinds of abnormalities namely: architectural distortions, stellate lesions, circumscribed masses and calcifications.

A real time clinical study was taken to analyze screening mammograms of breast cancer patients. The goal of this study is to reduce the number of false positive rates which help to avoid unnecessary biopsies and emotional stress to many women. Women after the age of 40 are advised to take mammograms every year and hence the total number mammograms evaluated worldwide in one year may be in the order of millions. All clinical mammograms that were collected from screening clinics were positive for presence of cancer. Mammograms were collected from 54 patients and all these patients have agreed to have their mammograms to be used in research studies. For each patient 4 mammograms were taken in two different views, one is the Craniocaudal (CC) and the other is the Mediolateral Oblique (MLO) view. The two projections of each breast (right and left) were taken for every case. A total of 216 mammograms were taken, all the mammograms were digitized to a resolution of 290 × 290 Dots per Inch (DPI). The real clinical mammograms were digitized with a CADPRO *advantage* digitizer. Each digitized mammograms was incorporated into a 2,020 × 2,708 pixel image (5.47 Mpixels).

4 Methods

In radiology, Computer-Aided Detection system helps to assist doctors in the interpretation of medical images. Screening mammography is found to be the best tool for detection of early stage breast cancer. Mammography allows for efficient diagnosis of breast cancers at an earlier stage. Though they are an efficient tool, radiologists misdiagnose 10–30 % of the malignant cases. It's been analyzed that of all the cases sent for surgical biopsy, only 10–20 % are actually malignant. However, when mammography is combined with an intelligent CAD system it is found to be an efficient tool for detecting abnormalities in mammograms.

A CAD system will analyze the mammograms and identifies the patterns associated with the abnormalities for the radiologists to consider before making a final recommendation for biopsy. Studies have reported the benefit of having

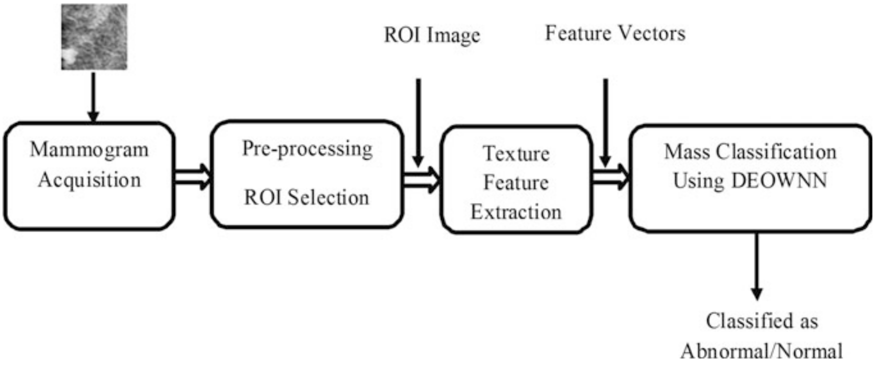

ROI Image
Feature Vectors
Mammogram Acquisition
Pre-processing
ROI Selection
Texture Feature Extraction
Mass Classification Using DEOWNN
Classified as Abnormal/Normal

Fig. 2 Block diagram of the DEOWNN based breast cancer detection system

double reading in screening mammography for increasing the detection rate [38, 39]. A computer system which is used for detecting breast cancer can be used as a second reader to increase the sensitivity rate.

The screen film mammograms are digitized and the digital mammogram image is then analyzed by the CAD system which marks the areas of concern like suspicious presence of calcification, masses, architectural distortions and bilateral symmetry. The CAD system evaluates the tissue textures in the mammogram and highlights the potential suspicious regions. This allows the radiologists to take an attention towards those regions and make the final conclusion regarding the pathology.

The proposed CAD system is based on a pattern recognition system which intelligently identifies the abnormal regions. CAD schemes using digital image processing techniques have the goal of improving the detection performance. Typically CAD systems are designed to provide a "second opinion" to aid rather than replacing the radiologist. Figure 2 shows the proposed approach for detection of abnormality in mammograms. The general approach of CAD for breast cancer detection in mammograms involves three stages:

1. Preprocessing.
2. Feature Extraction.
3. Classification.

Mammographic image analysis using a CAD system is an extremely challenging task. First, Since CAD systems are computer directed system, there is a need for a flawless system. Second, the large variability in the appearance of abnormalities makes this a very difficult image analysis task. Finally, abnormalities are often occluded or hidden in dense breast tissue, which makes detection difficult. Hence, there is a need for analysing the texture features of the mammogram and an intelligent classifier to classify the abnormalities (malignant tissues) in the mammograms.

The proposed detection system is composed of a segmentation method (ROI segmentation), a set of features for classification, and a classifier. For each of these main components researchers have developed a number of options from which it is nowadays possible to select particular components and use these in a modular fashion to build a complete detection system. However, even when breast cancer classification can be semi-automated, the involvement of experts in drawing accurate boundaries around abnormal regions is extremely time consuming and is not helpful in the endeavour to fully automate breast cancer diagnosis. In this context, the only way to make a significant advance in computer-aided breast cancer diagnosis may be to automate the abnormal boundary drawing as well as the classification. The proposed system has been designed in a framework of MATLAB 7.10, which aims at developing a CAD system for breast cancer detection.

4.1 Pre-processing

The goal of pre-processing the image is to simplify recognition of cancers (abnormalities) without throwing away any important information. Mammograms has breast region and is superimposed over background structures to which analysis is not necessary. One way would be to restrict the analysis to Region of Interest (ROI) that does not contain any background. The initial preprocessing is done in the digital mammogram to separate the region of interest (breast) and the dark background. The separation of ROI from the dark background is done using a global thresholding technique. Consider an input mammogram image $f(x, y)$, having light breast area on a dark background. The objects from the background are separated using a threshold value T and is defined as in Eq. (1). Then any point (x, y) for which $f(x, y) > T$ is called the breast area; otherwise, the point is called the background region. The threshold is chosen by visual inspection of the image histogram [40].

$$I(x, y) = \begin{cases} f, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \tag{1}$$

An intensity histogram is constructed and the local threshold value is chosen by statically examining the intensity values of the local neighborhood of each pixel. The mean of the local intensity distribution is calculated and used as a threshold value. The breast area in the mammogram only covers about 30 %, on average, of each mammogram. Based on this observation, the breast area is first segmented out in order to save processing and then further processing is restricted to the breast area.

The breast area in the database only covers about 30 %, on average, of each mammogram. Based on this observation, the breast area is first segmented out in order to save processing time and avoid false detections caused by markers and sharp edges near the chest side. Then further processing is restricted to the breast

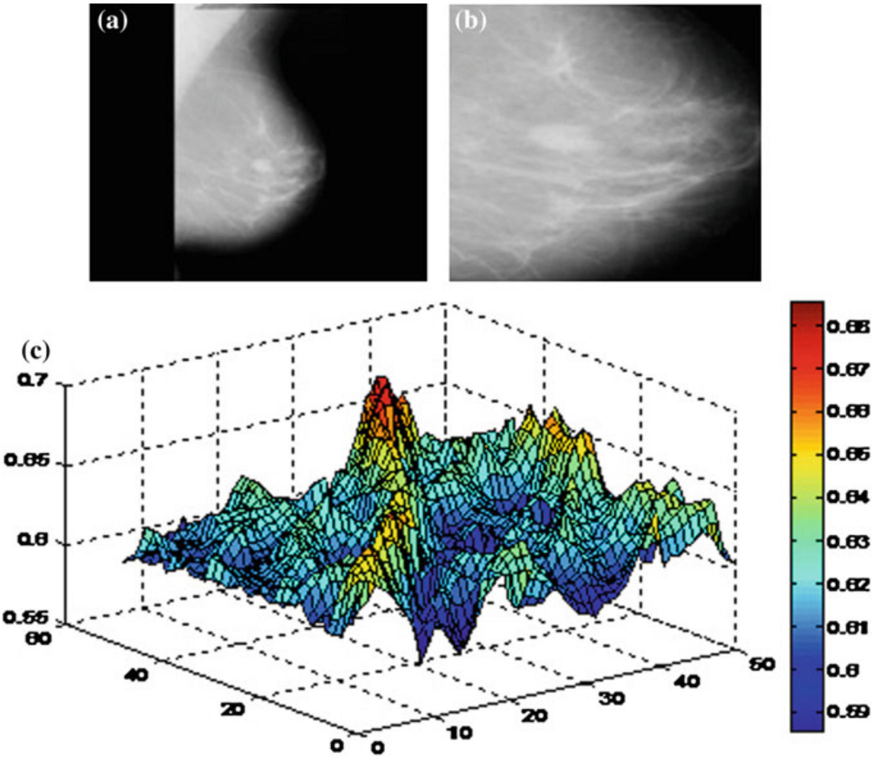


Fig. 3 Three dimensional representation of abnormality in mammogram **a** original image (*mdb* 010), **b** ROI image and **c** 3-D intensity distribution of the mass region

area. The mass intensity distribution of the ROI image is shown in Fig. 3b. The mass regions are visualized using 3-D intensity distribution in Fig. 3c and they are visualized with high intensity peaks.

4.2 Feature Extraction

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant then the input data will be transformed into a reduced representation set of features (features vector). Feature Extraction creates a set of features by decomposing the original data. A feature is a combination of attributes that is of special interest and captures important characteristics of the data. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. In Pattern recognition “relevant” information are extracted about an object via experiments and use these measurements (features) to classify an object [41].

Features such as shape, texture, color, etc. are used to describe the content of the image [42]. In most cases, medical images are based on carrying less information than color images. Medical images are usually low resolution and high noise images. They are difficult to automatically analyze for extracting features. Medical images acquired with different devices, even using the same modality, may have significantly varying properties. Moreover, color and intensity are not as important in medical images as in photographs; texture analysis becomes crucial in medical image. Texture also refers to visual patterns which have properties of homogeneity and cannot result from the presence of only a single color or intensity [43]. Texture perception plays an important role in the human visual system of recognition and interpretation.

Texture contains important information that is used by humans for the interpretation and the analysis of many types of images. Texture refers to the spatial interrelationships and arrangement of the basic elements of an image. Visually, these spatial interrelationships and arrangements of the image pixels are seen as variations in the intensity patterns or gray tones. Therefore, texture features have to be derived from the gray tones of the image. Texture has been one of the most important characteristic which has been used to classify and recognize objects and have been used in finding similarities between images in databases. These texture-analysis methods have been widely used in pattern recognition fields. The employment of texture features in medical imaging, especially in mammograms has been proved to be valuable. Texture features of an image to be classified are often used as inputs to a CAD system for discriminating between normal and abnormal tissues. The goal of texture classification then is to produce a classification map of the input image where each uniform textured region is identified with the texture class it belongs to.

The textural properties computed are closely related to the application domain to be used and becomes a vital property in medical imaging. Sutton and Hall [44] discuss the classification of pulmonary disease using texture features. Harms et al. [45] used image texture features to diagnose leukemic malignancy in samples of stained blood cells. Insana et al. [46] used textural features in ultrasound images to estimate tissue scattering parameters.

The three main approaches of pattern recognition for feature extraction and classification based on the type of features are as follows: (1) statistical approach, (2) syntactic or structural approach, and (3) spectral approach. In case of statistical approach, pattern/texture is defined by a set of statistically extracted features represented as vector in multidimensional feature space. The statistical features could be based on first-order, second-order, or higher-order statistics of gray level of an image. In case of syntactic approach, texture is defined by texture primitives, which are spatially organized according to placement rules to generate complete pattern. In syntactic pattern recognition, a formal analogy is drawn between the structural pattern and the syntax of language. In case of spectral method, textures are defined by spatial frequencies and are evaluated by autocorrelation function of a texture.

Feature based methods characterize a texture as a homogenous distribution of feature values such as gray level co-occurrence matrix (GLCM), Laws texture energy (LAWS) and Gabor features (GABOR). GLCM was introduced by Haralick [47], a co-occurrence matrix is used to describe how often one gray level appears in a specified spatial relationship to another gray level. The parameters used for constructing a GLCM are d and θ, where d is the distance between the two gray levels along a given direction θ. Haralick et al. [48] have applied textural features for image classification.

Normal Texture measures includes mean, variance, etc. which will be concatenated to a single feature vector. This will be fed to a classifier to perform classification. In this way, much of the important information contained in the whole distribution of the feature values might be lost. MC clusters usually appear as a few pixels with brighter intensity embedded in a textured background breast tissue [37]. By effectively extracting the texture information within any ROI of the mammogram, the region with MC and the region without MC can be differentiated. Laws Texture Energy Measures (LTEM) has proven to be a successful method to highlight high energy points in the image [49]. Anna et al. [20] suggests that LTEM has a best feature in analyzing texture of tissue for BC diagnosis. By considering the basic feature set the accuracy achieved using LTEM is 90 %.

The texture energy measures developed by Kenneth Ivan Laws at the University of Southern California have been used for many diverse applications [49, 50]. These texture features are used to extract Laws texture energy measures from the ROI containing abnormality and normal tissue patterns. These measures are computed by first applying small convolution kernels to the ROI and then performing a windowing operation.

A set of nine 5 × 5 convolution masks is used to compute texture energy, which is then represented by a vector of nine numbers for each pixel of the image being analyzed. The 2-D convolution kernels for texture discrimination are generated from the following set of 1-D convolution kernels of length five. The texture descriptions used are level, edge, spot, wave and ripple.

$$\begin{aligned}
\mathrm{L5} &= \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \\
\mathrm{E5} &= \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \\
\mathrm{S5} &= \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix} \\
\mathrm{W5} &= \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix} \\
\mathrm{R5} &= \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}
\end{aligned}$$

From this above 1-D convolution kernels 25 different two dimensional convolution kernels are generated by convoluting a vertical 1-D kernel with a horizontal 1-D kernel. Example for generating a 2-D mask from a 1-D is given below.

E5

L5

E5L5

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Similarly, 25 different two dimensional masks can be formed.

L5L5	E5L5	S5L5	W5L5	R5L5
L5E5	E5E5	S5E5	W5E5	R5E5
L5S5	E5S5	S5S5	W5S5	R5S5
L5W5	E5W5	S5W5	W5W5	R5W5
L5R5	E5R5	S5R5	W5R5	R5R5

The following steps will describe how texture energy measures are identified for each pixel in the ROI of a mammogram image.

Step 1: Apply the two dimensional mask to the preprocessed image i.e. the ROI to get $F(i, j)$, where $F(i, j)$ is a set of 25 $N \times M$ features.

Step 2: To generate the LTEM at the pixel, a non-linear filter is applied to $F(i, j)$. The local neighbourhood of each pixel is taken and the absolute values of the neighbourhood pixels are summed together. A 15 × 15 square matrix is taken for doing this operation to smooth over the gaps between the texture edges and other micro-features. The non linear filter applied is,

$$E(x,y) = \sum_{j=-7}^{7} \sum_{i=-7}^{7} |F(x+i, y+j)| \tag{2}$$

By applying the above Eq. (2) energy features per pixel are obtained. The TEM images are represented as,

L5L5T	E5L5T	S5L5T	W5L5T	R5L5T
L5E5T	E5E5T	S5E5T	W5E5T	R5E5T
L5S5T	E5S5T	S5S5T	W5S5T	R5S5T
L5W5T	E5W5T	S5W5T	W5W5T	R5W5T
L5R5T	E5R5T	S5R5T	W5R5T	R5R5T

Step 3: The texture features obtained from step 2 is normalized for zero-mean.

4.3 Classification

Following the feature extraction phase, classification of abnormal and normal regions in the mammograms is performed. The texture energy measures are collected for training and testing set. For abnormality detection, DEOWNN classifier is used, which is designed by encompassing WNN and DE [51].

The wavelet network uses wavelet activation function and preserves the universal approximation property. WNN is a feedforward neural network with an input layer, hidden layer and an output layer. The hidden layer is comprised normally of wavelets as activation function and the output layer is comprised of linear activation function. The output layer of WNN represents the weighted sum of the hidden layer units i.e. wavelet basis function. The backpropagation learning algorithm is used to update the network weights and to further minimize the standard Mean Square Error (MSE) of the networks approximation after network construction. Tuning a WNN is more important because of the following reasons,

Learning rate parameter, which, if not set properly, can either lead to oscillation or an indefinitely long training time.

Momentum Constant parameter is to accelerate the convergence of error propagation algorithm. The BP is just a gradient descent algorithm on the error space, which can be complex and contain many deceiving local minima. Therefore, the BPs are most likely gets trapped into a local minimum, making it entirely dependent on initial settings.

Number of hidden layers and hidden neurons, determination of the optimal number of hidden layers and hidden neurons is the most critical task. An ANN will not be capable of classifying a complex set of problems with no or few hidden neurons. In contrast, if the ANN has too many neurons/layers, it eventually leads to more complex networks and can also be highly time-consuming. The optimum number of hidden nodes/layers might depend on input/output vector sizes, training and test data sizes and more importantly the characteristics of the problem.

After the introduction of simplified neurons by McCulloch and Pitts [52], ANNs have been applied widely in many application areas, most of which use ANNs with the Back Propagation (BP) training algorithm. BP has the advantage of the directed search, in which weights are always updated to minimize the error. However, there are several aspects, which make the algorithm not guaranteed to be universally useful. Moreover, the Back-propagation (BP) algorithm is usually employed as the training algorithm of these networks. It turns out that the appropriate determination of a neural network including network structure and training plays an important role in the process of network design [53]. A too large structure can lead to unnecessary interconnected neurons, excessively computational load and over-fitting phenomenon of network; while a too small structure cannot provide desired network performance due to its limited information processing capability. Moreover, the BP algorithm based on gradient information usually results in slow convergent rate, gets easily trapped in a local minimum of the error function and is incapable of finding a global minimum in training phase [53, 54]. In an attempt to improve the classification accuracy of the WNN classifier, DE is used to tune the initial network parameters. Differential Evolution (DE) is a stochastic, population based optimization algorithm introduced by Storn and Prince [54].

5 Preliminaries

5.1 Wavelet Neural Network

Wavelet Neural Networks (WNN) are an efficient model for non linear pattern recognition [55, 56]. The wavelet transformation technique is used for obtaining information from signals that are aperiodic, noisy, intermittent or transient. Among many artificial intelligent methods, the feedforward ANN is the widely used statistical tool designed to diagnose pathological images especially of cancers and precancers. Thus, the study will strengthen the foundation of ANN in CAD application by combining the wavelet multiscale theory and neural network and obtain a novel high performance network—Wavelet Neural Networks. These networks are a new powerful tool for approximation and deals effectively with the problems of high dimensional model.

A Wavelet Neural Network was first introduced by Zhang and Benvniste [56] as a class of feedforward networks composed of wavelets. The discrete wavelet transform is used for analysing and synthesising feedforward neural network. The wavelet network uses wavelet activation function and preserves the universal approximation property.

WNN is a feedforward neural network with an input layer, hidden layer and an output layer. The hidden layer is comprised normally of wavelets as activation function and the output layer is comprised of linear activation function. The output layer of WNN represents the weighted sum of the hidden layer units i.e. wavelet basis function. The backpropagation learning algorithm is used to update the network weights and to further minimize the standard Mean Square Error (MSE) of the networks approximation after network construction.

Wavelet transforms a signal into components of different frequencies, allowing to study each component separately. The basic idea of wavelet transform is mapping the signals from one basis to another. Wavelets depend on two variables: scale (or frequency) and time and have two functions namely the scale function and the mother wavelet. Wavelets are powerful signal analysis tools. They can approximately realize the time-frequency analysis using a mother wavelet. The mother wavelet has a square window in the time- frequency space. The two well-known types of wavelets are Continuous Wavelet Transform (CWT), which deals with the function defined over the whole real axis and Discrete Wavelet Transform (DWT), which has a range of integers $(t = 0, 1..., N - 1)$, where N is the number of values in the time series.

A wavelet is a real or complex valued function $\psi(.)$ satisfying the following two conditions,

1.

$$\int_{-\infty}^{\infty} \psi(u)du = 0$$

2.

$$\int_{-\infty}^{\infty} |\psi^2(u)|du = 1$$

The function $\psi(.)$ is generally referred as mother wavelet. A family of wavelets can be created by translating and dilating this mother wavelet.

$$\psi_{\lambda,t}(u) = \frac{1}{\sqrt{\lambda}}\psi\left(\frac{u-t}{\lambda}\right) \tag{3}$$

where $\lambda > 0$, t is finite and $\left\|\psi_{\lambda,t}\right\| = \|\psi\|$ for all λ, t.

CWT takes more umber of dilations and translations of the mother wavelet and hence lot of redundancy in CWT. Discrete Wavelet Transform operates on a discretely sampled function or time series $x(.)$, time $t = 0, 1, \ldots, N-1$ to be finite. The dilations is denoted by λ and is of the form $2^{j-1}, j = 1, 2, 3, \ldots$, and the translation values are sampled at 2^j intervals when analysing within a dilation of 2^j $^{-1}$. The DWT samples at discrete times and scales, to reduce redundancy. DWT is a system of two filters one is the wavelet filter and the other is the scaling filter. The wavelet filter is a high pass filter and the scaling filter is the low pass filter.

The input signal X(z) is split by two filters H_0(z) and H_1(z) into a low pass component X_0 and a high pass component X_1, both of which are decimated (down-sampled) by 2:1. In order to reconstruct the signal, a pair of reconstruction filters G_0(z) and G_1(z) and usually the filters are designed such that output signal Y(z) is identical to the input X(z). A Haar wavelet is the simplest type of wavelet. In discrete form, Haar wavelets are related to a mathematical operation called the Haar transform. The Haar transform serves as a prototype for all other wavelet transforms. Like all wavelet transforms, the Haar transform decomposes a discrete signal into two subsignals of half its length. One subsignal is a running average or trend; the other subsignal is a running difference or fluctuation. A major problem in the development of wavelets during the 1980s was the search for scaling functions that are compactly supported, orthogonal, and continuous. These scaling functions were first constructed by Daubechies [57] that created great excitement in the wavelet research world.

The Daubechies wavelet transforms are defined in the same way as the Haar wavelet transform by computing the running averages and differences via scalar products with scaling signals and wavelets the only difference between them consists in how these scaling signals and wavelets are defined. The Daubechies wavelets are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. This wavelet type has balanced frequency responses but non-linear phase responses. Daubechies wavelets use overlapping windows, so the high frequency coefficient spectrum reflects all high frequency changes. Therefore Daubechies wavelets are useful in compression and noise removal of audio signal processing. Daubechies 4-tap wavelet has been chosen for this implementation.

Daubechies [57] discovered a class of wavelets, which are characterised by orthonormal basis functions. That is, the mother wavelet is orthonormal to each function obtained by shifting it by multiples of 2^j and dilating it by a factor of 2^j (where $j \in Z$). The Daubechies wavelet '*db4*' is a four-term member of the same class. The four scaling function coefficients, which solve the above simultaneous equations for N = 4, are specified as follows:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$$

$$h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$$

$$h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}$$

$$h_4 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

The scaling function for the *db4* wavelet can be built up recursively from these coefficients. The wavelet function can be built from the coefficients g_k which are found using the relation $g_k = (-1)^k h_{4-k-1}$. Wavelet Neural Networks combine the theory of wavelets and neural networks into one. A wavelet neural network generally consists of a feed-forward neural network, with one hidden layer, whose activation functions are drawn from an orthonormal wavelet family. The structure of a wavelet neural network is very similar to that of a (1 + 1/2) layer neural network. That is, a feed-forward neural network, taking one or more inputs, with one hidden layer and whose output layer consists of one or more linear combiners or summers as shown in Fig. 4. The hidden layer consists of neurons, whose activation functions are drawn from a wavelet basis. These wavelet neurons are usually referred to as wavelons.

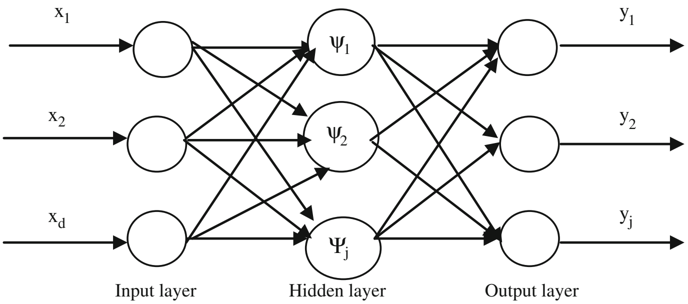


Fig. 4 Wavelet neural network structure

The WNN consists of three layers: input layer, hidden layer and output layer. All the units in each layer are fully connected to the nodes in the next layer. The input layer receives the input variable $X = [x_1, x_2, ..x_d]^T$ and send it to the hidden layer. The nodes in this layer are given as the product of the *j*th multi-dimensional wavelet with N input dimensions as

$$\psi_j(x) = \prod_{i=1}^{N} f_{\lambda,t}(x_i) \tag{4}$$

$f_{\lambda,t}$ is the activation function of the hidden layer and λ and t are the dilations and translations respectively.

The products of the hidden layer are then propagated to the output layer, where the output of the WNN will be the linear combination of the weighted sum of the hidden layer, which is represented in the form of,

$$y_j = \sum_i w_{ij}\psi_j(x) + b_j \tag{5}$$

where b_j the bias of node j between the hidden layer and the output layer $\psi_j(x)$ is taken as a Daubechies mother wavelet. The error is calculated by finding the difference between the target output (d_j) and the actual output (y_j). This error is then used mathematically to change the weights in such a way that the error will get smaller. The process is repeated again and again until the error is minimal. The localized wavelet activation function in the hidden layer of the WNN, the connection weight associated with the hidden nodes can be viewed as local piecewise constant models, which leads to learning efficiency and structure transparency. The pseudocode of training algorithm for WNN is outlined below.

```
for $i$ = 1,2,…, $N$ do
begin
select the number of neurons in the hidden layer with wavelet activation function
normalize the input ($x_i$) and the output ($y_i$) neurons.
initialize the dilation ($\lambda$) and translation ($t$) parameters
initialize learning rate = 0.01, momentum constant = 0.9
    initialize the weights to random values
    Assign random weights between ($x_i$) and hidden neurons
    Assign random weights between hidden neurons and ($y_i$)
end
repeat
    for each training pattern ($x_i$, $y_i$) do
    begin
        train the patterns
calculate the error by finding the difference between the network output and the desired output
        $E_j = \frac{1}{2}\Sigma_j(d_j - y_j)^2$
update the weights
    end
until error is acceptably low
```

5.2 *Differential Evolution*

Differential Evolution (DE) is a stochastic, population based optimization algorithm introduced by Storn and Prince [54]. DE has good convergence property, ability to handle non-linear and multimodal cost function and easy to use. Like other evolutionary algorithms, DE starts with a randomly selected initial population vector. In order to optimize the D-dimensional real parameter, the parameter vector has the form,

$$X_{i,G} = (x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}) \quad i = 1, 2, \ldots, N$$

where G, is the generation number and N is the population size. The initial population at 0th generation should cover the entire search space. The lower and upper bounds for each parameter is designed as, $X_j^L = (x_1^L, x_2^L, \ldots, x_D^L)$ and $X_j^U = (x_1^U, x_2^U, \ldots, x_D^U)$ respectively. The jth individual of the ith population is initialized as,

$$x_{j,i,0} = x_j^L + rand_{i,j}[0, 1] \times (x_j^U - x_j^L) \tag{6}$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1. Each of the N parameter vectors undergoes mutation, crossover and selection.

5.2.1 Mutation

For a given parameter vector $X_{i,G} = (x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G})$ $i = 1, 2, \ldots, N$ in generation G, randomly select three vectors $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ such that i, r_1, r_2, r_3 are different mutually exclusive random integers chosen from $\{1, 2, \ldots, N\}$. The mutant vector is generated according to,

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \tag{7}$$

In the above equation, the mutation factor F is a constant integer from [0, 2], which scales the differential variation of $(X_{r2,G} - X_{r3,G})$ and $V_{i,G}$is called the donor vector.

5.2.2 Crossover

Crossover operation is done to increase the potential diversity of the population. This operation incorporates successful solutions from the previous generations. The trial vector $U_{i,G} = [u_{1,i,G}, u_{2,i,G} \ldots u_{D,i,G}]$ is obtained from the target vector $X_{i,G}$ and donor vector $V_{i,G}$. The trial vector is formed from,

$$U_{j,i,G} = \begin{cases} V_{j,i,G} & \mathit{if}\ rand_{j,i} \leq CR \quad \mathit{or} \quad j = I_{rand} \\ X_{j,i,G} & \mathit{if}\ rand_{j,i} > CR \quad \mathit{and} \quad j \neq I_{rand} \end{cases} \tag{8}$$

where $rand_{j,i} \in U[0, 1]$ is a uniformly distributed random number, I_{rand} is a random number from $[1, 2, \ldots, D]$ and ensures that $V_{i,G} \neq X_{i,G}$ and CR is the crossover rate constant that appears like a control parameter for DE and $CR \in [0, 1]$.

5.2.3 Selection

The next step of the algorithm is the selection process which determines whether the target vector ($X_{i,G}$) or the trial vector ($U_{i,G}$) enter into the next generation. The target vector is compared with the trial vector and the one with the lowest fitness value will enter into the generation $G + 1$.

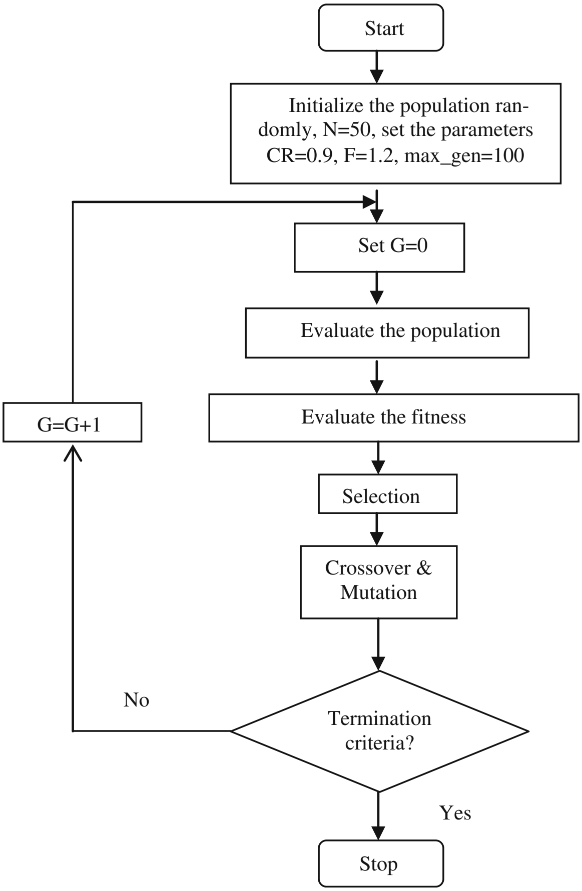
$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \mathit{if}\ f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & \mathit{if}\ f(U_{i,G+1}) > f(X_{i,G}) \end{cases} \quad i = 1, 2, \ldots, N \tag{9}$$

where $f(.)$ is the function to be minimized. If the new trial vector yields an equal or lower fitness value, then it replaces the corresponding target vector in the next generation. Otherwise, the target is retained in the population.

6 Differential Evolution Algorithm for Evolving WNN

In the proposed method, DEOWNN is applied for evolving fully connected Neural Network with wavelet activation function and is optimized with best network architecture by optimizing the number of neurons in the hidden layer, the learning rate and the momentum factor. Finding an optimal learning rate avoids major disruption of the direction of learning when very unusual pair of training patterns is presented. The main advantage of using optimal momentum factor is to accelerate the convergence of error propagation algorithm. The number of neurons in the input layer and output layer is fixed based on the problem defined. Let N_I represents the size of the neurons in the input layer and N_O represents the size of the neurons in the output layer. The number of neurons in the input and output layer are fixed and they are same for the entire configuration in the architecture spaces. The number of hidden layers in this problem is restricted and made as one. The range of the

Fig. 5 Flowchart for DEOWNN design



optimization process is defined by two range arrays $X_{\min} = \{Nh_{\min}, Lr_{\min}, Mc_{\min}\}$ and $X_{\max} = \{Nh_{\max}, Lr_{\max}, Mc_{\max}\}$where, *Nh* is the number of neurons in the hidden layer, *Lr* is the learning rate and *Mc* is the momentum factor. The fitness function sought for optimal training is the Mean Square Error (MSE) formulated as,

$$MSE_{DEOWNN} = \sum\nolimits_{p \in T} \sum\nolimits_{k=1}^{N_o} \left(t_k^p - y_k^{p,o}\right)^2 \tag{10}$$

where t_k^p is the target (desired) output, $y_k^{p,o}$is the actual output from the *k*th neuron in the output layer *o*, for the pattern *p* in the training set. With the framed fitness function the DEOWNN algorithm automatically evolve a best solution. The flowchart for the DEOWNN design is given in Fig. 5 and the pseudo code for DEOWNN design is given below.

```
Let G be the Generation, N be the Population size, CR be the Crossover rate constant

Initialize
        $G = 0, N = 50, CR \in [0,1]$
define
        upper and lower bounds for each parameter
        $X_j^L \le X_{j,i,1} < X_j^U$
        Initial parameter values of each individual are randomly selected on the intervals [$X_j^L, X_j^U$]
        for each individual belonging to $P_G$ at $G = 0$
        $x_{j,i,0} = x_j^L + rand_{i,j}[0,1] \times (x_j^U - x_j^L)$
end
while stopping criteria is not reached
do
for $i = 1,2,.....,N$
        select random numbers from $i = 1,2,.....,N$
            such that $i, r_1, r_2, r_3 \in (1,2,...,N)$ & $i, r_1, r_2, r_3$ are distinct numbers
        select $I_{rand} \in (1,2,...,N)$
    Generate donor vector, $\forall\ i \le D$
            $V_{i,G} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G})$
    Generate trial vector
            $U_{j,i,G} = \begin{cases} V_{j,i,G} \ if\ rand_{j,i} \le CR \ or \ j = I_{rand} \\ X_{j,i,G} \ if\ rand_{j,i} > CR \ and\ j \ne I_{rand} \end{cases}$
end
        Evaluate trial vector
            $X_{i,G+1} = \begin{cases} U_{i,G+1} \ if\ f(U_{i,G+1}) \le f(X_{i,G}) \\ X_{i,G} \ if\ f(U_{i,G+1}) > f(X_{i,G}) \end{cases}$
        Increment $G = G + 1$
end while
```

7 Experimental Work and Analysis

In the experimental level the main problem covered is to distinguish between different types of malignancy based on the texture properties and to classify the mammographic image as normal and abnormal. The proposed detection tool are designed to detect microcalcification clusters, speculated masses, circumscribed masses ill defined masses, architectural distortions. Two databases is used in this study MIAS database and mammograms collected from real clinical database, these database are collected because they have different types of mammographic abnormal cases. Only 30 % of the mammogram contains the breast region and the remaining are composed of noisy background. Hence a threshold is applied based method to cut off the noisy background region thereby separating the breast region. The objective of this process is to remove irrelevant and unwanted background of the mammogram. The region of interest (ROI) is segmented from the breast area by applying a threshold for representing the colour map range. The threshold range contains a low value and a high value. The output obtained is an image with zeros representing the segmented ROI. In order to preserve all true abnormal regions and to reduce the number of false positives, the thresholds must be carefully determined.

To verify the effectiveness of the proposed approach two databases were used: Database generated by Mammographic Image Analysis Society (MIAS) and Mammograms collected from real clinical database (Real clinical Database). Table 1 shows the distribution of cases taken for experimental analysis from real clinical database.

The collected mammograms are randomly chosen for training and the final performance is evaluated using the receiver operating characteristic curve. For the purpose of pattern (abnormal and normal) classification, features were extracted from the segmented ROI. Texture contains important information that is used by the human for the interpretation and analysis of many types of images. The objective is to investigate the significance of texture information present in the abnormal regions as compared to the normal region in terms of discriminating capabilities. The presence of microcalcification and masses causes architectural distortions in the surrounding tissues. As a result, mammographic images possess textural information that could bear discriminating features. The following features were extracted using Laws texture descriptors like level (L), edge (E), spot (S), wave (W) and ripple (R).

Table 1 Summary of the cases used for experimental analysis from clinical database

Class of malignancy	# of Images
Circumscribed masses	44
Ill-defined	76
Obscured masses	10
Spiculated masses	62
Microcalcification	24
Total	216

1. LL—texture energy from the LL kernel
2. EE—texture energy from the EE kernel
3. SS—texture energy from the SS kernel
4. LE—average texture energy from the LE and El kernels
5. ES—average texture energy from the ES and SE kernels
6. LS—average texture energy from the LS and SL kernels

The normal patterns include all the benign cases like tissues having cysts, fibroadenomas, ducts, ligaments and parenchymal patterns. For each ROI a window size of 15 × 15 pixels were taken for Laws feature extraction. Based on the different types of anatomical differences evidenced by the mammographic appearances, 25 Laws texture features were extracted based on the descriptors level, edge, spot, wave and ripple. This comprises a total of 25 input features and is given as input to the classifier. DEOWNN classifier is thus designed with 25 input neurons and one neuron at the output layer.

7.1 Detection of Abnormality in MIAS Database Using DEOWNN

The training patterns were taken from the MIAS database and a total of 2,050 patterns are used for training. Training of DEOWNN is done in such a way that the desired outputs are assigned a value 1 for cancers (abnormality) and zero for non cancer (normal breast tissue). The optimization of DEOWNN classifier is performed with the learning rate and the momentum factor varied from 0 to 1 and the hidden neurons varied from 31 to 200. For this training a maximum of 100 generations are performed with a population size N = 50 and with 500 training epochs. The value of the mutation factor F is set to 1.2 and the crossover constant CR is set to 0.9. During each generation, the best fitness score (minimum MSE) achieved at the optimum dimension is stored. Using the proposed DEOWNN algorithm, an optimized WNN is achieved with $Nh = 132$, $Lr = 0.00127$ and $Mc = 0.9264$.

Figure 6 shows the classification results of abnormalities in various mammogram images from MIAS database. The results demonstrate the strength of the proposed methodology. The results demonstrate the usefulness of the DEOWNN classifier in identifying cancerous and non cancerous regions. Due to the fact that masses and microcalcifications are the two types of objects that are the best indicators of a possible early stage of breast cancer, identifying the abnormal cells are more important to increase the survival.

The DEOWNN classifier achieves a classification accuracy of 96.203 % and AUC of 0.97843 for MIAS database, which is found to be higher than the other optimally tuned classifier models. The evaluation results show that the DEOWNN classifier is capable of achieving a sensitivity of 96.923 % with a specificity of 92.857 % for MIAS.

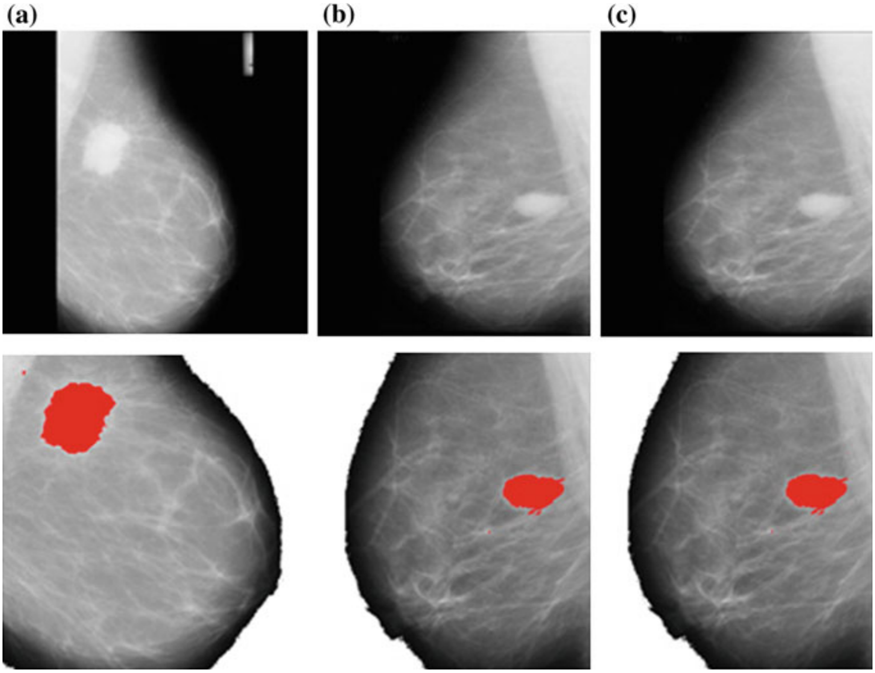


Fig. 6 Detection results of abnormalities in mammograms for MIAS database. The *top images* show the original mammograms of mdb184, mdb025, mdb083, mdb248 respectively. Detected masses are shown below **a** spiculated mass, **b** circumscribed mass and **c** asymmetry mass

7.2 Detection of Abnormalities in Real Clinical Database

The DEOWNN methodology has been evaluated on real clinical database collected form mammogram screening centres. The mammograms in the training subset were found to have a total of 1,064 patterns containing both abnormal and normal patterns. The DEOWNN algorithm was trained with the same parameters used for analyzing the MIAS database. An optimized WNN is achieved with $Nh = 114$, $Lr = 0.00112$ and $Mc = 0.9275$. Testing is done for all the 216 real time clinical images.

The normal breast tissues of woman below 40 years of age are much denser, which may be predicted as an abnormal mass region leading to high miss classification rate [58]. Different kinds of mammograms encountered in clinical applications are considered and the experimental results reveal that masses are detected effectively even in very dense breast mammograms in most of the cases. Masses are characterized by the margin of the mass. The mammographic border between the mass and the normal tissue is useful for predicting benign and malignant masses. Margins of a mass is described as circumscribed, obscured, ill defined and spiculated. The circumscribed masses have a well defined margin. Figure 7 shows

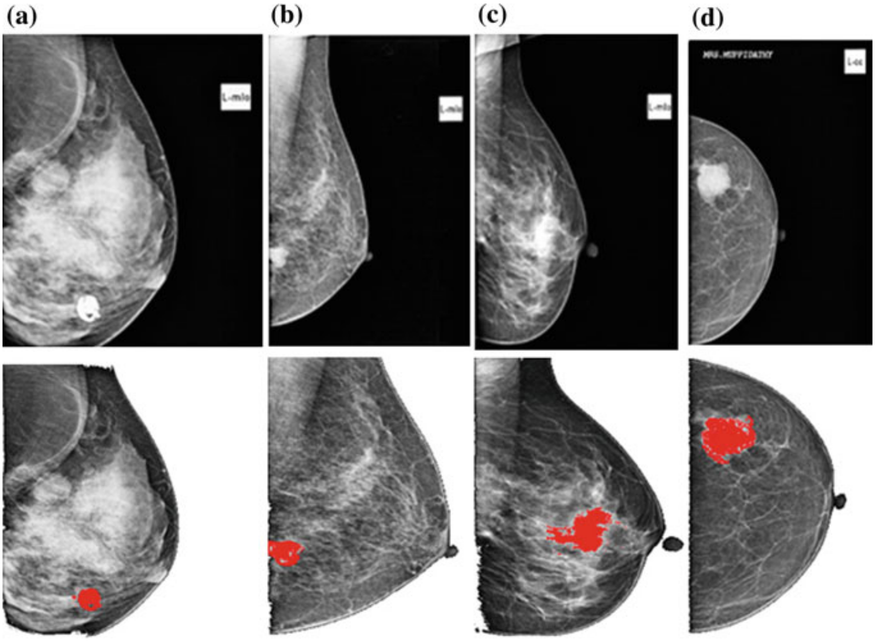


Fig. 7 Detection results of abnormalities in mammograms. The *top images* show the original mammograms. Detected masses are shown below **a** circumscribed mass, **b** obscured mass, **c** spiculated mass and **d** ill defined mass

detection of masses using DEOWNN classifier and shows the power of Laws texture features in discriminating the abnormal and normal tissue patterns. The mammograms illustrated in Fig. 7 are denser and the margins are obscured and not clearly seen. Hence it is difficult for a radiologist to identify the mass in a denser tissue. The most difficult to diagnose mass in mammograms are ill defined and spiculated masses. The shapes of these masses are irregular and have ill defined and spiculated margins.

As observed, the proposed DEOWNN scheme has achieved a sensitivity of 93.333 % at specificity level 89.474 % when applied to real clinical database. The area under the ROC curve is analysed and found to be 0.9573 and a classification accuracy of 92.405 % is achieved for DEOWNN classifier.

8 Performance Analysis

To evaluate the performance of the proposed system in detecting the microcalcification clusters Receiver Operating Characteristic (ROC) curve is used. ROC curve is a plot of the True positive Rate (TPR) versus False Positive Rate (FPR) Metz [59]. The TPR denote the fraction of patients actually having the abnormality and that are diagnosed as positive and the FPR is the fraction of patients actually without the abnormality and that are diagnosed as positive. The detection performance is analyzed using the area under the ROC curve (A_Z). Several metrics is determined for quantitative evaluations of the intelligent classifiers. The ROC curve is the fundamental tool for diagnostic test evaluation [60]. It has become very popular in biomedical applications, particularly in radiology and imaging. It is also used in machine learning applications to assess classifiers performance. The true positive rate also known as sensitivity is the ratio of the malignant cases correctly classified to the total number of malignant cases in the test cases. The false positive rate is the ratio of the number of normal cases incorrectly classified to the total number of normal cases in the test case. Sensitivity measures the proportion of actual positives which are correctly identified when the mammogram contains cancers tissues in it. Specificity measures the proportion of negatives which are correctly identified when cancer is not present in the mammogram. The following statistics can be defined,

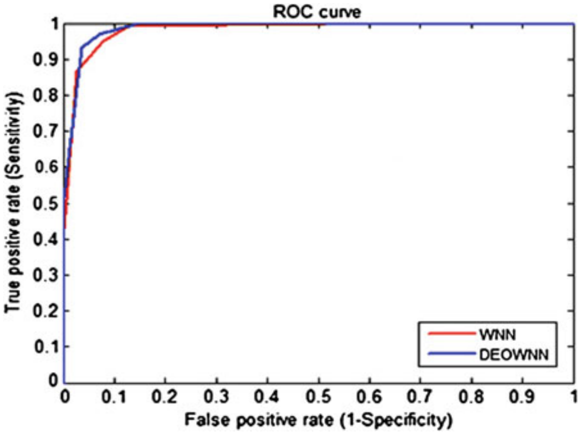
$$sensitivity = \frac{TP}{(TP + FN)}$$

$$specificity = \frac{TN}{(TN + FP)}$$

$$TPR = sensitivity \quad FPR = 1 - specificity$$

The Area under the ROC curve (AUC or A_Z) is a measure of how well a parameter can distinguish between two diagnostic groups (abnormal/normal tissues). AUC can be interpreted as the probability that the test results from a randomly chosen diseased individual is more indicative of disease than that from a randomly chosen non-diseased individual. The overall performance of diagnostic systems has been measured and reported in terms of classification accuracy which is the percentage of diagnostic decisions that proved to be correct. Figures 8 and 9. shows the ROC curve for classifiers using MIAS Database and real clinical database respectively.

Fig. 8 ROC curve for MIAS database



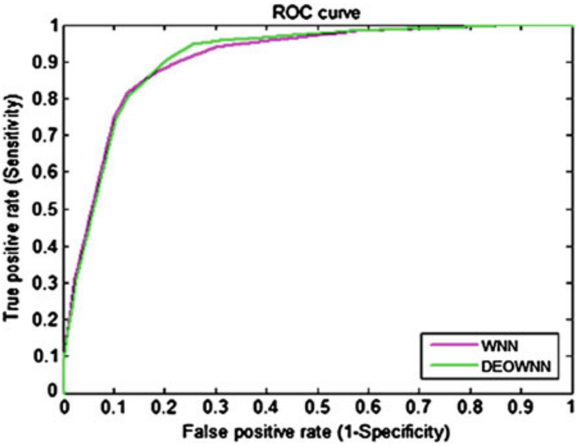


Fig. 9 ROC curve for real clinical database

In order to evaluate the laws texture features on DEOWNN classifier additional metrics like sensitivity, specificity, classification accuracy, Youden's index and misclassification rate is considered. Youden's index (J) developed by Youden [61] is a single statistic that captures the performance of a diagnostic test is given as J and is calculated as,

$$J = sensitivity + specificity - 1$$

The result emphasizes the potential of DEOWNN algorithm to be used as a classifier for abnormality detection in mammograms. Table 2 demonstrates the performance of the classifier using various summary measures. It was of more interest to focus the experimentation on trying to improve the classification rate by focusing on optimizing the parameters of WNN. Consequently, by using optimized parameter for WNN the classification accuracy is improved drastically.

Summarizing the results for MIAS database and for real clinical mammogram the classification accuracy of DEOWNN is higher than that of the other well known classifier models. This is because of the fact that the DEOWNN incorporates the

Table 2 Classification results for WNN and optimized WNN

Classifiers/Metrics	Performance measures for MIAS database		Performance measures for real clinical database	
	WNN	DEOWNN	WNN	DEOWNN
Sensitivity (%)	90.244	96.923	86.441	93.333
Specificity (%)	88.571	92.857	82.5	89.474
Accuracy (%)	89.873	96.203	85.44	92.405
AUC	0.94413	0.97843	0.8920	0.95735
Youden's index	0.78815	0.8978	0.6894	0.82807
Misclassification rate	0.10127	0.03797	0.1455	0.07594

wavelet neural network and optimally designs the neural network using differential evolution algorithm. This superior performance makes DEOWNN suitable for efficiently detecting abnormalities in mammograms. The proposed classifier designed using DE algorithm applied to WNN are investigated for detecting breast cancer in mammograms and the results gave better classification accuracy than the traditional classifiers. The optimized wavelet neural network accelerates the convergence of the back propagation algorithm and also it avoids major disruptions in the direction of learning.

9 Conclusion

In this chapter, a novel method is investigated for the detection of suspicious cancerous regions in mammograms using an intelligent breast cancer detection system. Laws texture features are used in the proposed methods and DEOWNN classifier is used to classify the abnormal regions in the digital mammograms. The proposed intelligent breast cancer detection system can be used to achieve the detection of subtle signs of breast cancer at early stage. The performance evaluation demonstrates that the result of the optimized wavelet neural network classifier is better than that of the non-optimized neural network. This is consistent with the fact that optimization is useful in initial parameter setting of the network. The DE-OWNN classifier designed using DE algorithm applied to WNN for classifying abnormal and normal patterns in mammograms. The resulting DEOWNN classifier achieves the main design objective thereby maintaining a robust and generic architecture with superior performance. The proposed DEOWNN classifier performs training and testing with good performance compared to other classifiers under study and hence it is more suitable for real time medical image analysis. As a result, the DEOWNN classifier can be conveniently applied to classifying abnormalities in mammograms, thus providing a second reader for the radiologists.

References

1. Freer, T.W., Ulissey, M.J.: Screening mammography with computer-aided detection: prospective study of 12,860 patients in a community breast center. Radiology **220**, 781–786 (2001)
2. Tsui, P.-H., Liao, Y.-Y., Chang, C.-C., Kuo, W.-H., Chang, K.-J., Yeh, C.-K.: Classification of benign and malignant breast tumors by 2-d analysis based on contour description and scatterer characterization. IEEE Trans. Med. Imaging **29**(2), 513–522 (2010)
3. Giger, M.L., Karssemeijer, N., Armato, S.G.: Computer aided diagnosis in medical imaging. IEEE Trans. Med. Imaging **20**, 1205–1208 (2001)
4. Sheppard, L.M.: Not your mother's mammography. IEEE Spectr **39**, 56–57 (2002)
5. Zheng, B., Chang, Y.H., Gur, D.: Computerized Detection of masses in digitized mammograms using single-image segmentation and a multilayer topographic feature analysis. Acad. Radiol. **2**, 959–966 (1995)

6. Baker, J.A., Rosen, E.L., Lo, J.Y., Gimenez, E.I., Walsh, R., Soo, M.S.: Computer-aided detection (CAD) in screening mammography: sensitivity of commercial CAD systems for detecting architectural distortion. Am. J. Roentgenol. **81**, 1083–1088 (2003)
7. Tourassi, G.D., Floyd Jr, C.E.: Performance evaluation of an information-theoretic CAD scheme for the detection of mammographic architectural distortion. In: Proceedings of SPIE—The International Society for Optical Engineering, vol. 5370, pp. 59–66 (2004)
8. Lau, T.K., Bischof, W.F.: Automated detection of breast tumors using the asymmetry approach. Comput. Biomed. Res. **24**, 273–295 (1991)
9. Yin, F.F., Giger, M.L., Doi, K., Vyborny, C.J., Schmidt, R.A.: Computerized detection of masses in digital mammograms: automated alignment of breast images and its effect on bilateral-subtraction technique. Med. Phys. **21**, 445–452 (1994)
10. Chan, H.P., Doi, K., Vyborny, C.J., Schmidt, R.A., Metz, C.E., Lam, K.L., Ogura, T., Wu, Y., MacMahon, H.: Improvement in radiologists detection of clustered microcalcifications on mammograms. The potential of computer-aided diagnosis. Invest. Radiol. **25**, 102–1110 (1990)
11. Cheng, H.D., Cai, X., Chen, X., Hu, L., Lou, X.: Computer-aided detection and classification of microcalcifications in mammograms: a survey. Pattern Recogn. **36**, 2967–2991 (2003)
12. Motakis, E., Ivshina, A.V., Kuznetsov, V.A.: Data-driven approach to predict survival of cancer patients. IEEE Eng. Med. Biol. Mag. **28**, 58–66 (2009)
13. Bird, R.E., Wallace, T.W., Yankaskas, B.C.: Analysis of cancers missed at screening mammography. Radiology **184**, 613–617 (1992)
14. Azar, A.T., El-Metwally, S.M.: Decision tree classifiers for automated medical diagnosis. Neural Comput. Appl. **23**(7–8), 2387–2403 (2013). doi:10.1007/s00521-012-1196-7
15. Azar, A.T.: Statistical analysis for radiologists' interpretations variability in mammograms. Int. J. Syst. Biol. Biomed. Technol. (IJSBBT) **1**(4), 28–46 (2012)
16. Azar, A.T., El-Said, S.A.: Performance analysis of support vector machines classifiers in breast cancer mammography recognition. Neural Comput. Appl. **24**(5), 1163–1177 (2014). doi:10.1007/s00521-012-1324-4
17. Moftah, H.M., Azar, A.T., Al-Shammari, E.T., Ghali, N.I., Hassanien, A.E., Shoman, M.: Adaptive k-means clustering algorithm for MR breast image segmentation. Neural Comput. Appl. **24**(7–8), 1917–1928 (2014). doi:10.1007/s00521-013-1437-4
18. Tiedeu, A., Daul, C., Kentsop, A., Graebling, P., Wolf, D.: Texture-based analysis of clustered microcalcifications detected on mammograms. Digit. Signal Proc. **22**, 124–132 (2012)
19. Yu, S.-N., Huang, Y.-K.: Detection of microcalcifications in digital mammograms using combined model-based and statistical textural features. Expert Syst. Appl. **37**(7), 5461–5469 (2010)
20. Anna, N., Ioannis, S., Spyros, G., Filippos, N., Nikolaos, S., Eleni, A., George, S., Lena, I.: Breast cancer diagnosis: analyzing texture of tissue surrounding microcalcifications. IEEE Trans. Inf Technol. Biomed. **12**, 731–738 (2008)
21. Azar, A.T., El-Said, S.A.: Probabilistic neural network for breast cancer classification. Neural Comput. Appl. **23**(6), 1737–1751 (2013). doi:10.1007/s00521-012-1134-8
22. Mudigonda, N.R., Rangayyan, R.M., Leo Desautels, J.E.: Detection of breast masses in mammograms by density slicing and texture flow-field analysis. IEEE Trans. Med. Imaging **20**, 1215–1227 (2001)
23. Li, H., Wang, Y., Liu, K.J.R., Lo, S.C.B., Matthew, T.: Computerized radiographic mass detection part I: lesion site selection by morphological enhancement and contextual segmentation. IEEE Trans. Med. Imaging **20**, 289–301 (2001)
24. Gao, X., Wang, Y., Li, X., Tao, D.: On combining morphological component analysis and concentric morphology model for mammographic mass detection. IEEE Trans. Inf Technol. Biomed. **14**, 266–273 (2010)
25. Mudigonda, N.R., Rangayyan, R.M., Leo Desautels, J.E.: Gradient and texture analysis for the classification of mammographic masses. IEEE Trans. Med. Imaging **19**(10), 1032–1043 (2000)

26. Suliga, M., Deklerck, R., Nyssen, E.: Markov random field-based clustering applied to the segmentation of masses in digital mammograms. Comput. Med. Imaging Graph. **32**, 502–512 (2008)
27. Grim, J., Somol, P., Haindl, M., Danes, J.: Computer aided evaluation of screening mammograms based on local texture model. IEEE Trans. Image Process. **18**, 765–773 (2009)
28. Heine, J.J., Deans, S.R., Cullers, D.K., Stauduhar, R., Laurence, P.: Multiresolution statistical analysis of high-resolution digital mammograms. IEEE Trans. Med. Imaging **16**, 503–515 (1997)
29. Kupinski, M.A., Giger, M.L.: Automated seeded lesion segmentation on digital mammograms. IEEE Trans. Med. Imaging **17**(4), 510–517 (1998)
30. Nakayama, R., Uchiyama, Y., Yamamoto, K., Watanabe, R., Namba, K. Computer aided diagnosis scheme using a filter bank for detection of microcalcification clusters in mammograms. IEEE Trans. Biomed. Eng. **53**(2), 273–283 (2006)
31. Cheng, H.-D., Lui, Y.M., Freimanis, R.I.: A novel approach to microcalcification detection using fuzzy logic technique. IEEE Trans. Med. Imaging **17**, 442–450 (1998)
32. Wang, T.C., Karayiannis, N.B.: Detection of microcalcifications in digital mammograms using wavelets. IEEE Trans. Med. Imaging **17**, 498–509 (1998)
33. Eltoukhy, M.M., Faye, I., Samir, B.B.: Breast cancer diagnosis in digital mammogram using multiscale curvelet transform. Comput. Med. Imaging Graph. **34**, 269–276 (2010)
34. Verma, B., McLeod, P., Klevansky, A.: Classification of benign and malignant patterns in digital mammograms for the diagnosis of breast cancer. Expert Syst. Appl. **37**, 3344–3351 (2010)
35. Teo, J., Chen, Y., Soh, C.B., Gunawan, E., Low, K.S., Putti, T.C., Wang, S.-C.: Breast lesion classification using ultra wideband early time breast lesion response. IEEE Trans. Antennas Propag. **58**, 2604–2613 (2010)
36. Peng, R., Hao, C., Varshney, P.K.: Noise-enhanced detection of micro-calcifications in digital mammograms. IEEE J. Sel. Top. Sign. Process. **3**, 62–73 (2009)
37. Suckling, J., Parker, J.: The Mammographic Images Analysis Society Digital Mammogram Database. In: Proceedings of 2nd International Workshop on Digital Mammography, UK, pp. 375–378 (1994)
38. Anderson, E.D., Muir, B.B., Walsh, J.S., Kirkpatrick, A.E.: The efficacy of double reading mammograms in breast screening. Clin. Radiol. **49**, 248–251 (1994)
39. Thurfjell, E.L., Lernevall, K.A., Taube, A.A.: Benefit of Independent Double Reading in a Population based Mammography Screening Program. Radiology **191**, 241–244 (1994)
40. Gonzales, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Upper Saddle River, NJ (2002)
41. Rafael, C., Gonzalez, R.E., Woods S.L.: Digital Image Processing Using MATLAB. Pearson Education India (2005)
42. Tsai, D.-Y., Kojima, K.: Measurement of texture features of medical images and its application to computer aided diagnosis in cardiomyopathy. Measurement **37**, 284–292 (2005)
43. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based feature distributions. Pattern Recogn. **29**, 51–59 (1996)
44. Sutton, R.N., Hall, E.L.: Texture measures for automatic classification of pulmonary disease. IEEE Trans. Comput. **C-21**, 667–676 (1972)
45. Harms, H., Gunzer, U., Aus, H.M.: Combined local color and texture analysis of stained cells. Comput. Vis. Graphics Image Process. **33**, 364–376 (1986)
46. Insana, M.F., Wagner, R.F., Garra, B.S., Brown, D.G., Shawker, T.H.: Analysis of ultrasound image texture via generalized Rician statistics. Opt. Eng. **25**, 743–748 (1986)
47. Haralick, R.M.: Statistical and structural approaches to texture. Proc. IEEE **67**, 786–804 (1979)
48. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **SMC-3**, 610–621 (1973)
49. Laws, K.J.: Texture energy measures. Proceeding DARPA Image Understanding Workshop, pp. 47–51 (1979)

50. Christodoulou, C.I., Pattichis, C.S., Pantziaris, M., Nicolaides, A.: Texture-based classification of atherosclerotic carotid plaques. IEEE Trans. Med. Imaging **22**, 902–912 (2003)
51. Chauhan, N., Ravi, V., Karthik Chandra, D.: Differential evolution trained wavelet neural network application to bankruptcy prediction in banks. Expert Syst. Appl. **36**, 7659–7665 (2009)
52. McCulloch, W.S., Pitts, W.: A logical study of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**, 115–133 (1943)
53. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, New York (1995)
54. Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
55. Zhang, J., Walter, G.G., Miao, Y., Lee, W.N.W.: Wavelet neural networks for function learning. IEEE Trans. Signal Process. **43**, 1485–1497 (1995)
56. Zhang, Q., Benvniste, A.: Wavelet networks. IEEE Trans. Neural Netw. **3**, 889–898 (1992)
57. Daubechies, I.: Time-frequency localization operators: a geometric phase space approach. IEEE Trans. Inf. Theory **34**, 605–612 (1988)
58. Zheng, B., Qian, W., Clarke, L.P.: Digital mammography mixed feature neural network with spectral entropy decision for detection of microcalcifications. IEEE Trans. Med. Imaging **15**, 589–597 (1996)
59. Metz, C.E.: ROC methodology in radiologic imaging, Invest Radiol **21**(9), 720–733 (1986)
60. Zweig, M.H., Campbell, G.: Receiver operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clin. Chem. **39**, 561–577 (1993)
61. Youden, W.J.: An index for rating diagnostic tests. Cancer **3**, 32–35 (1950)