Advances in Intelligent Systems and Computing 509

Sudip Kumar Sahana
Sujan Kumar Saha *Editors*

Advances in Computational Intelligence

Proceedings of International Conference on Computational Intelligence 2015

Springer

Advances in Intelligent Systems and Computing

Volume 509

Series editor

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

About this Series

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Sudip Kumar Sahana · Sujan Kumar Saha
Editors

Advances in Computational Intelligence

Proceedings of International Conference on Computational Intelligence 2015

Springer

Editors

Sudip Kumar Sahana
Department of Computer Science
and Engineering
Birla Institute of Technology Mesra
Ranchi, Jharkhand
India

Sujan Kumar Saha
Department of Computer Science
and Engineering
Birla Institute of Technology Mesra
Ranchi, Jharkhand
India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-10-2524-2 ISBN 978-981-10-2525-9 (eBook)
DOI 10.1007/978-981-10-2525-9

Library of Congress Control Number: 2016950378

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

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #22-06/08 Gateway East, Singapore 189721, Singapore

Preface

This book comprises the proceedings of the First International Conference on Computational Intelligence (ICCI 2015) held during December 10–11, 2015 at Birla Institute of Technology, Mesra, Ranchi, India. ICCI 2015 was an international gathering for the researchers working on all aspects of computational intelligence and provided a high-level academic forum for the participants to disseminate their new research findings in the emerging areas of research. It also created a stimulating environment for the participants to interact and exchange information on future challenges and opportunities in the field of computational intelligence.

The definition of computational intelligence was introduced by Bezdek (1994) as: a system is called computationally intelligent if it deals with low-level data such as numerical data, has a pattern-recognition component and does not use knowledge in the artificial intelligence sense. Additionally, Bezdek and Marks (1993) clearly mentioned that computational intelligence should be based on soft computing methods. And the principal constituents of soft computing (SC) are fuzzy logic (FL), feural computing (NC), evolutionary computation (EC) machine learning (ML), and probabilistic reasoning (PR). In this book we targeted to cover the recent trends, developments and future possibilities of these soft computing techniques. Additionally, we also cover the bio-inspired computing techniques, which have become widely popular in recent times, in this book.

Computational intelligence has a rational theoretical part as well as an experimental wing. In these proceedings we try to balance theory and experiments by selecting papers from both the new theoretical findings in different concepts of computational intelligence as well as the application papers which are full of experiments for exploring new possibilities where the computational intelligence technique are superior.

The book is divided into seven parts, namely, Fuzzy Logic, Artificial Neural Network, Genetic Algorithm and Other Bio-inspired Computing, Applications in Cloud Computing, Applications in Image Processing, Applications in Security, and Other Applications of Computational Intelligence.

ICCI 2015 received 121 submissions from different countries in the world. Each submission was reviewed by at least two reviewers, and on average 2.4 reviewers per paper. Based on the rigorous reviews by the reviewers, 41 high-quality papers were selected for publication with an acceptance rate of 33.88 %.

Ranchi, India

Sudip Kumar Sahana
Sujan Kumar Saha

Acknowledgements

International Conference on Computational Intelligence (ICCI 2015) was a watershed event enriched by contributions from researchers all over the country and also from other countries. We were privileged to provide them with a platform to express their scientific intellect. It was these contributions that made ICCI a really successful endeavor. We are grateful to every single author who sent their submissions to the conference.

ICCI 2015 would definitely not have been possible without the constant and untiring efforts of every single person in the organizing committee. We are thankful to them for making the conference the success that it was.

Our special gratitude is towards the wonderful members of the program committee and reviewers who were our guiding light and led us in the right direction whenever we stood at crossroads during the organization of the conference. The high standard set by the papers is a reflection of the efforts that had been put by the reviewers.

We would also like to thank the members editorial team at Springer for giving a final shape to these proceedings and fructifying the efforts put in by all involved in ICCI 2015. It is only through their zeal and encouragement that we have been able to assimilate these proceedings in a timely manner.

ICCI 2015 expresses its sincere gratitude to one and all who contributed in making the event the grand success that it was.

Sudip Kumar Sahana
Sujan Kumar Saha

Contents

Part I Fuzzy Logic

Part II Artificial Neural Network

Part III Genetic Algorithm and Bio-inspired Computing

Part IV Applications in Cloud Computing

Part V Applications in Image Processing

Part VI Applications in Security

Part VII Other Applications of Computational Intelligence

About the Editors

Dr. Sudip Kumar Sahana was born in Purulia, West Bengal, India on October 8, 1976. He received B.E. degree in Computer Technology from Nagpur University, India in 2001, M.Tech. degree in Computer Science in 2006, and Ph.D. in Engineering in 2013 from the B.I.T. (Mesra), Ranchi, India. His major field of study is computer science. He is currently working as Assistant Professor at the Department of Computer Science and Engineering, B.I.T. (Mesra), Ranchi, India. His research and teaching interests include soft computing, grid computing, network traffic management, and artificial intelligence. He is author of number of research papers in the field of computer science. He is a lifetime member of Indian Society for Technical Education (ISTE).

Dr. Sujan Kumar Saha is currently a faculty member at the Department of Computer Science and Engineering of Birla Institute of Technology Mesra, Ranchi, India. He has done his Ph.D. at the Computer Science and Engineering Department of IIT Kharagpur (2010) and M.Tech. from IIT Delhi (2005). His main research interests include natural language processing, machine learning, web mining and educational technology. He has authored around 25 research articles in various reputed international journals and conferences. Total citation count of his research papers is 156. He has also served as a reviewer of various journals and conferences.

Part I
Fuzzy Logic

A New Approach to Interval-Valued Fuzzy Soft Sets and Its Application in Decision-Making

B.K. Tripathy, T.R. Sooraj and R.K. Mohanty

Abstract Soft set (SS) theory was introduced by Molodtsov to handle uncertainty. It uses a family of subsets associated with each parameter. Hybrid models have been found to be more useful than the individual components. Earlier interval-valued fuzzy set (IVFS) was introduced as an extension of fuzzy set (FS) by Zadeh. Yang introduced the concept of IVFSS by combining and soft set models. Here, we define IVFSS through the membership function approach to define soft set by Tripathy et al. very recently. Several concepts, such as complement of an IVFSS, null IVFSS, absolute IVFSS, intersection, and union of two IVFSSs, are redefined. To illustrate the application of IVFSSs, a decision-making (DM) algorithm using this notion is proposed and illustrated through an example.

Keywords SS · FS · FSS · IVFS · IVFSS · DM

1 Introduction

Fuzzy set introduced by Zadeh [1] in 1965 has been found to be a better model of uncertainty and has been extensively used in real-life applications. To bring topological flavor into the models of uncertainty and associate family of subsets of a universe to parameters, SS model was proposed in 1999 [2]. The study on SS was carried out by Maji et al. [3, 4]. As mentioned in the abstract, hybrid models obtained by suitably combining individual models of uncertainty have been found to be more efficient than their components. Following this trend Maji et al. [5] put forward the concept of FSS as a hybrid model from FS and SS. Tripathy et al. [6] defined soft sets through their

B.K. Tripathy (✉) · T.R. Sooraj · R.K. Mohanty
SCOPE, VIT University, Vellore, Tamilnadu, India
e-mail: tripathybk@vit.ac.in

T.R. Sooraj
e-mail: soorajtr19@gmail.com

R.K. Mohanty
e-mail: rknmohanty@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_1

characteristic functions. This approach has been highly authentic and helpful in defining the basic operations, such as the union, intersection, and complement of soft sets. Similarly, it is expected that defining membership function for fuzzy soft sets will systematize many operations defined upon them as done in [7, 8]. Extending this approach further, we introduce the membership functions for IVFSS in this paper. In [2], some applications of SS were discussed. In [3], an application to decision-making is proposed. This study was further extended to the context of FSSs by Tripathy et al. [8], where they identified some drawbacks in [3] and took care of these drawbacks while introducing an algorithm for decision-making. In this paper, we have carried this study further using IVFSS in handling the problem of multi-criteria decision-making. This notion further extended in [9–14].

It is well known that IVFSS introduced by Yang [15] is a more realistic model of uncertainty than the fuzzy set. In [16], an application of IVFSS is given. This concept is extended in [17] by taking parameters as fuzzy entities. Here, we follow the definition of soft set proposed in [6] in defining IVFSS and redefine the basic operations on them. The highlight of this work is the introduction of a decision-making algorithm that uses IVFSS, and we illustrate the suitability of this algorithm in real-life situations. In addition, it generalizes the algorithm introduced in [8] while keeping the authenticity intact.

2 Definitions and Notions

By *P*(*U*) and *I*(*U*), we denote the power set and the fuzzy power set of *U*, respectively.

Definition 2.1 (*Soft Set*) A pair (*F, E*) is called as a soft set over the universal set *U*, where

$$F: E \to P(U) \tag{2.1}$$

The pair (*U*, *E*), which is a combination of a universal set *U* and a parameter set *E*, is called a soft universe.

Definition 2.2 We denote a FSS over (*U, E*) by (*F, E*), where

$$F: E \to I(U) \tag{2.2}$$

Let *I*([0, 1]) denote the set of all closed subintervals of [0, 1].

Definition 2.3 (IVFS) An IVFS *X* on a universe *U* is a mapping, such that

$$\mu_X: U \to Int([0,1]) \tag{2.3}$$

Moreover, $\forall x \in U, \mu_X(x) = [\mu_X^-(x), \mu_X^+(x)] \subseteq [0, 1]$. Here, $\mu_X^-(x)$ and $\mu_X^+(x)$ represent as the lower and upper degrees of membership of x to X.

3 Interval-Valued FSS

In this section, we follow the membership function approach introduced in [7] to define IVFSS. The basic operations on IVFSS are also redefined. Let (*F, E*) be an IVFSS. We associate with (*F, E*) a family of parameterized membership functions $\mu_{(F,E)} = \left\{ \mu^a_{(F,A)} | a \in E \right\}$ as in (3.1).

Definition 3.1 Given $a \in E$ and $x \in X$, we define

$$\mu^a_{(F,E)}\ (\mathrm{x}) = [\alpha, \beta] \in \mathrm{I}([0,1]) \tag{3.1}$$

For any two IVFSS (*F*, *E*) and (*G*, *E*), we define the following operations.

Definition 3.2 The union of (*F, E*) and (*G, E*) is the IVFSS (*H, E*), and $\forall a \in E$ and $\forall x \in U$, we have

$$\begin{aligned}(F,E) \cup (G,E)(x) &= \max[\mu^a_{(F,E)}(x), \mu^a_{(G,E)}(x)] \\ &= [\max(\mu^{a-}_{(F,E)}(x), \mu^{a-}_{(G,E)}(x)), \max(\mu^{a+}_{(F,E)}(x), \mu^{a+}_{(G,E)}(x))]\end{aligned} \tag{3.2}$$

where $\mu^{a-}_{(F,E)}$ and $\mu^{a+}_{(F,E)}$ denotes the lower and upper membership value of the IVFSS.

Definition 3.3 The intersection of (*F, E*) and (*G, E*) is the IVFSS (*H, E*), and $\forall a \in E$ and $\forall x \in U$, we have

$$\begin{aligned}(F,E) \cap (G,E)(x) &= \min[\mu^a_{(F,E)}(x), \mu^a_{(G,E)}(x)] \\ &= [\min(\mu^{a-}_{(F,E)}(x), \mu^{a-}_{(G,E)}(x)), \min(\mu^{a+}_{(F,E)}(x), \ \mu^{a+}_{(G,E)}(x))]\end{aligned} \tag{3.3}$$

Definition 3.4 (*F, E*) is said to be interval valued fuzzy soft subset of (*G, E*), $(F,E) \subseteq (G,E)$. Then, $\forall a \in E$, $\forall x \in U$,

$$\mu^{a+}_{(F,E)}(x) \le \mu^{a+}_{(G,E)}(x) \text{ and } \mu^{a-}_{(F,E)}(x) \le \mu^{a-}_{(G,E)}(x) \tag{3.4}$$

Definition 3.5 We say that (*F, E*) is equal to (*G, E*) written as (*F, E*) = (*G, E*) if $\forall x \in U$,

$$\mu^{a+}_{(F,E)}(x) = \mu^{a+}_{(G,E)}(x) \text{ and } \mu^{a-}_{(F,E)}(x) = \mu^{a-}_{(G,E)}(x) \tag{3.5}$$

Definition 3.6 For any two IVFSSs (*F, E*) and (*G, E*) over a common soft universe (*U, E*), we define the complement (*H, E*) of (*G, E*) in (*F, E*) as $\forall a \in E$ and $\forall x \in U$.

$$\mu^{a+}_{(H,E)}(x) = \max\left\{0, \mu^{a+}_{(F,E)}(x) - \mu^{a+}_{(G,E)}(x)\right\} \text{ and } \mu^{a-}_{(H,E)}(x) = \max\left\{0, \mu^{a-}_{(F,E)}(x) - \mu^{a-}_{(G,E)}(x)\right\} \tag{3.6}$$

Definition 3.7 The complement of an IVFSS over a soft universe (*U, E*) can be derived from the above Definition 3.6 by taking (*F, E*) as *U* and (*G, E*) as (*F, E*). We denote it by $(F, E)^c$ and clearly $\forall x \in U$ and $\forall e \in E$,

$$\begin{aligned} &\mu^{e+}_{(F,E)^c}(x) = \max(0, \mu^{e+}_{U}(x) - \mu^{e+}_{(F,E)}(x)) \text{ and} \\ &\mu^{e-}_{(F,E)^c}(x) = \max(0, \mu^{e-}_{U}(x) - \mu^{e-}_{(F,E)}(x)) \end{aligned} \tag{3.7}$$

It can be seen easily that

$$\mu^{e+}_{(F,E)^c}(x) = 1 - \mu^{e+}_{(F,E)}(x)) \text{ and } \mu^{e-}_{(F,E)^c}(x) = 1 - \mu^{e-}_{(F,E)}(x)) \tag{3.8}$$

4 Application of IVFSS in Decision-Making

Tripathy et al. [8] rectified some of the issues in [3] and provided suitable solutions for the problems in that paper. In addition, there is the concept of negative and positive parameters that was introduced.

Consider the case of an interview conducted by an organization, where interview performance of each candidate is analyzed by a panel. Here, we assign some parameters to evaluate the performance of each candidate. Some parameters are communication skills, personality, reactivity, etc.

We denote a set of candidates as U = {c_1, c_2, c_3, c_4, c_5, c_6} and *E* be the parameter set given by E = {knowledge, communication, presentation, reaction, other activities}. We denote the parameters as e_1, e_2, e_3, e_4, and e_5 for further calculations, where e_1 denotes knowledge, e_2 denotes communication, e_3 denotes presentation, e_4 denotes reaction, and e_5 denotes other activities. Consider an IVFSS (*U, E*) which describes the 'performance of a candidate'.

Table 1 shows the IVFSS of performance of candidates in a selection process. In the case of IVFSSs, we need to consider three cases.

Table 1 Tabular representation of IVFSS

U	e_1	e_2	e_3	e_4	e_5
c_1	0.2–0.4	0.3–0.5	0.8–0.9	0.4–0.7	0.6–0.9
c_2	0.4–0.8	0.6–0.9	0.2–0.5	0.7–1	0.5–0.6
c_3	0.5–0.8	0.7–0.9	0.7–0.8	0.8–1	0.5–0.7
c_4	0.6–0.8	0.5–0.9	0.8–1	0.5–0.9	0.7–0.8
c_5	0.1–0.4	0.9–1	0.3–0.6	0.1–0.5	0.8–1
c_6	0.9–1	0.5–0.7	0.1–0.3	0.2–0.4	0.3–0.7

(i) Pessimistic
(ii) Optimistic
(iii) Neutral

Neutral values are obtained by taking the average of pessimistic values and optimistic values.

$$neutral\,value = \frac{pesimistic + optimistic}{2} \tag{4.1}$$

4.1 Algorithm

1. Input the IVFSS.
2. Get the priority of the parameters from the user which lies in [−1, 1]. The default priority value for a parameter is 0 (Zero), which means that the parameter has no impact on decision-making and can be opted out from further computation.
3. Extract the pessimistic, optimistic, and neutral values from the interval valued fuzzy sets.
4. Do the following steps for pessimistic, optimistic, and neutral values of IVFSSs.
    a. Multiply the priority values with the corresponding parameter values to get the priority table.
    b. Compute the row sum of each row in the priority table (PT).
    c. Construct the comparison table (CT) by finding the entries as differences of each row sum with those of all other rows.
    d. Compute the row sum for each row in the CT to get the score.
    e. Assign rank to each candidate based on the CT values.
5. Construct the decision table based on the results we got in the above step, and final decision can be taken by the sum of all the ranks.
6. The object having highest value in the final decision column is to be selected. If more than one candidate is having the same rank-sum, then the candidate having higher value under the highest absolute priority column is selected and will continue like this.

In the case of pessimistic decision-making, we consider the lower membership value of each parameter, in the case of optimistic decision-making, we need to take the highest membership value of each parameter, and in the case of neutral decision–making, we need to take average of both pessimistic and optimistic values. First, we are considering the pessimistic case. The values of the pessimistic case are shown in Table 2.

Table 2 Pessimistic values

U	e_1	e_2	e_3	e_4	e_5
c_1	0.2	0.3	0.8	0.4	0.6
c_2	0.4	0.6	0.2	0.7	0.5
c_3	0.5	0.7	0.7	0.8	0.5
c_4	0.6	0.5	0.8	0.5	0.7
c_5	0.1	0.9	0.3	0.1	0.8
c_6	0.9	0.5	0.1	0.2	0.3

Table 3 Priority table for pessimistic case

U	e_1	e_2	e_3	e_4	e_5
c_1	0.14	0.09	0.16	–0.2	0.24
c_2	0.28	0.18	0.04	–0.35	0.2
c_3	0.35	0.21	0.14	–0.4	0.2
c_4	0.42	0.15	0.16	–0.25	0.28
c_5	0.07	0.27	0.06	–0.05	0.32
c_6	0.63	0.15	0.02	–0.1	0.12

Table 4 CT for pessimistic case

c_i	c_j							
	c_1	c_2	c_3	c_4	c_5	c_6	Row sum	Rank
c_1	0	0.08	–0.07	–0.33	-0.24	–0.39	–0.95	5
c_2	–0.08	0	–0.15	–0.41	-0.32	–0.47	–1.43	6
c_3	0.07	0.15	0	–0.26	-0.17	–0.32	–0.53	4
c_4	0.33	0.41	0.26	0	0.09	–0.06	1.03	2
c_5	0.24	0.32	0.17	–0.09	0	–0.15	0.49	3
c_6	0.39	0.47	0.32	0.06	0.15	0	1.39	1

Here, the panel is assigning some priority to the parameters. The priorities for the parameters e_1, e_2, e_3, e_4, and e_5 are 0.7, 0.3, 0.2, –0.5, and 0.4, respectively. With the help of this priority values, we create a priority table, as shown in Table 3.

The CT is formed as in step 4c of the algorithm, which is shown in Tables 4, 5, and 6.

Similarly, we need to find the comparison table for optimistic and neutral cases. The comparison table for optimistic decision-making is given in Table 5.

The CT for neutral decision-making is shown in Table 6.

The final decision can be taken as the average of pessimistic, optimistic, and neutral decision-making. It shown in Table 7.

From this table, we can see that candidate c_6 is the best choice. The next choices are in the order of c_4, c_5, c_3, c_2, and c_1.

Table 5 CT optimistic decision-making

	c_1	c_2	c_3	c_4	c_5	c_6	Row sum	Rank
c_1	0	−0.05	−0.15	−0.28	−0.23	−0.36	−1.07	6
c_2	0.05	0	−0.1	−0.23	−0.18	−0.31	−0.77	5
c_3	0.15	0.1	0	−0.13	−0.08	−0.21	−0.17	4
c_4	0.28	0.23	0.13	0	0.05	−0.08	0.61	2
c_5	0.23	0.18	0.08	−0.05	0	−0.13	0.31	3
c_6	0.36	0.31	0.21	0.08	0.13	0	1.09	1

Table 6 CT for neutral decision-making

	c_1	c_2	c_3	c_4	c_5	c_6	Row sum	Rank
c_1	0	0.015	−0.11	−0.305	−0.235	−0.375	−1.01	6
c_2	−0.015	0	−0.125	−0.32	−0.25	−0.39	−1.1	5
c_3	0.11	0.125	0	−0.195	−0.125	−0.265	−0.35	4
c_4	0.305	0.32	0.195	0	0.07	−0.07	0.82	2
c_5	0.235	0.25	0.125	−0.07	0	−0.14	0.4	3
c_6	0.375	0.39	0.265	0.07	0.14	0	1.24	1

Table 7 Decision table

	Pessimistic	Optimistic	Neutral	Row sum	Final decision
c_1	5	6	6	17	6
c_2	6	5	5	16	5
c_3	4	4	4	12	4
c_4	2	2	2	6	2
c_5	3	3	3	9	3
c_6	1	1	1	3	1

5 Conclusions

In [6], the notion of soft set was defined using the characteristic function approach, which was further extended in [8] to take care of FSS. Here, we further extended the approach to define IVFSS and redefined all basic operations on them. These are elegant and authentic. In addition, an algorithm to handle decision-making where the input data is in the form of IVFSS is proposed. A suitable example illustrates the application of the algorithm in real-life situations.

References

1. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
2. Molodtsov, D.: Soft set theory—first results. Comput. Math Appl. **37**, 19–31 (1999)
3. Maji, P.K., Biswas, R., Roy, A.R: An application of soft sets in a decision making problem. Comput. Math. Appl. **44**, 1007−1083 (2002)
4. Maji, P.K., Biswas, R., Roy, A.R.: Soft set theory. Comput. Math Appl. **45**, 555–562 (2003)
5. Maji, P.K., Biswas, R., Roy, A.R.: Fuzzy Soft Sets. J. Fuzzy Math. **9**(3), 589–602 (2001)
6. Tripathy, B.K., Arun, K.R.: A new approach to soft sets soft multisets and their properties. Int. J. Reason.-based Intell. Syst. **7**(3), 244–253 (2015)
7. Sooraj, T.R., Mohanty, R.K., Tripathy, B.K.: Fuzzy soft set theory and its application in group decision making. In: Advanced Computing and Communication Technologies. Advances in Intelligent Systems and Computing, vol. 452, pp.171−178 (2015)
8. Tripathy, B.K., Sooraj, T.R., Mohanty, R.K.: A new approach to fuzzy soft set theory and its application in decision making. In: Computational Intelligence in Data Mining—Volume 2, pp. 305–313 (2016)
9. Mohanty, R.K., Sooraj, T.R., Tripathy, B.K.: An Application of IVIFSS in medical diagnosis decision making. Int. J. Appl. Eng. Res. (IJAER) **10**(92), 85–93 (2015)
10. Mohanty, R.K., Sooraj, T.R., Tripathy, B.K.: IVIFSS and decision making. In: Proceedings of ICDECT-2016. Springer (2016)
11. Tripathy, B.K., Mohanty, R.K., Sooraj, T.R., Arun, K.R.: A new approach to intuitionistic fuzzy soft sets and their application in decision making. Adv. Intell. Syst. Comput. **439**, 93–100 Springer (2016)
12. Tripathy, B.K., Mohanty, R.K., Sooraj, T.R., Tripathy, A.: A modified representation of IFSS and its usage in GDM. In: Smart Innovation, Systems and Technologies, vol. 50, pp. 365−375 (2016)
13. Tripathy, B.K., Mohanty, R.K., Sooraj, T.R.: On intuitionistic fuzzy soft sets and their application in decision making. In: Proceedings of ICSNCS-2016, NewDelhi (2016)
14. Tripathy, B.K., Mohanty, R.K., Sooraj, T.R.: On intuitionistic fuzzy soft set and its application in group decision making. In: Proceedings of ICETETS-2016, Thanjavur (2016)
15. Yang, X.B., Lin, T.Y., Yang, J.Y., Li, Y., Yu, D.: Combination of interval-valued fuzzy set and soft set. Comput. Math Appl. **58**(3), 521–527 (2009)
16. Tripathy, B.K., Sooraj, T.R., Mohanty, R.K.: A new approach to interval-valued fuzzy soft sets and its application in group decision making. In: proceedings of CDCS-2015, Kochi (2015)
17. Tripathy, B.K., Panigrahi, A.: Interval-valued intuitionistic fuzzy parameterized soft set theory and its application in decision making. In: Proceedings of 10th international conference on intelligent systems and control (ISCO 2016), vol. 2, pp. 385–390 (2016)

A Grid-Based Approach to Prolong Lifetime of WSNs Using Fuzzy Logic

Ajai Kumar Mishra, Rakesh Kumar, Vimal Kumar and Jitendra Singh

Abstract Wireless sensor networks (WSNs) are autonomous, self-configured and consist of distributed sensors for monitoring any physical or environmental conditions. Sensor nodes cooperatively disseminate their data through the network to a base station. In recent years, such networks have shown its wide applicability in various areas. Generally, sensor nodes are small, cost-effective, memory constrained and having limited processing capabilities for sensing data in any particular region from the environment. Energy is one of the significant factors in such network. Whole network lifetime depends on how efficiently consumption of energy takes place. Sensor nodes are combined into groups which is called cluster. The purpose of clustering approach is to make the consumption of energy in more effective way. A cluster head node is used for collecting sensed data from cluster nodes for transmitting to the base station. An efficient election of cluster head minimizes energy consumption, thereby increasing network lifetime. One major drawback in dynamic clustering approach is that in every round, cluster head selection is done locally and decides the cluster region. This process has extra communication cost in massage exchange to select the appropriate cluster head. Transmission of message from one node to another node consumes energy that leads to inefficient use of energy resource. In this paper, a non-probabilistic grid-based approach to prolong the WSNs lifetime using fuzzy logic has been proposed. In this, whole network is divided into predefined grid area and selecting a node as grid head (GH) using two fuzzy variables, viz., base station distance and residual energy of sensor nodes. This approach uses a multi-hop communication

A.K. Mishra · R. Kumar · V. Kumar (✉) · J. Singh
Department of Computer Science & Engineering,
Madan Mohan Malaviya University of Technology, Gorakhpur 273010, India
e-mail: vimalmnnit16@gmail.com

A.K. Mishra
e-mail: akm.r.mishra@gmail.com

R. Kumar
e-mail: rkiitr@gmail.com

J. Singh
e-mail: jitendra6890@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_2

approach. GH nodes are authorized to communicate with other GH nodes and base station. Simulation results show that the proposed approach prolongs WSNs network lifetime than existing ones.

Keywords Grid Head • Fuzzy logic • Non-probabilistic • Wireless sensor networks

1 Introduction

Recent advancement in microelectromechanical system (MEMS) has made availability of cheaper small wireless sensor nodes feasible nowadays. Normally, WSNs are deployed for specific purposes. Such networks consist of wireless nodes, which have scarce resources in terms of energy, memory, and computational power. Sensor nodes use radio frequency (RF) for communication purpose. A sensor node consists of five units, viz., transceiver, sensor, processor, memory, and a power unit. Figure 1 shows architecture of a wireless sensor node.

Most important unit in all five units defined above is power unit, because wireless sensor network lifetime completely depends on power source. Sensor nodes used in wireless sensor network are battery operated which is non-rechargeable in nature and replacement of batteries is not feasible [1]. Whole network lifetime depends mostly on sensor nodes power source. An efficient use of clustering technique minimizes sensor nodes energy consumption. In this technique, only some nodes allowed to communicate with a base station [2–4]. Nodes have this characteristic called cluster head (CH). Controlling and managing energy consumption in an efficient way are a major challenge in WSNs. Data communication process is more energy consuming with respect to data processing at nodes. Energy consumption may be minimized by making efficient communication between nodes. To execute three thousands instructions, energy consumption is equivalent to one bit data transmission at one hundred meters [5]. A cluster head is accountable for gathering data from other member nodes of a cluster and sending

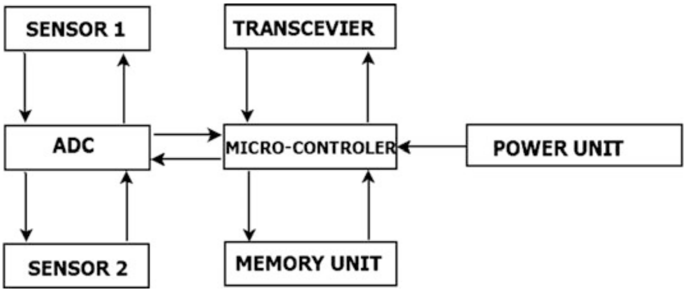


Fig. 1 Architecture of sensor node

processed data to a base station using aggregation mechanism [6]. An efficient election of cluster head (CH) can significantly minimize energy consumption, thereby increasing network lifetime. Uncertainties in WSNs can be handled by the use of fuzzy logic. Zedah et al. [3] in their study demonstrated that algorithms based on probabilistic theory are less appropriate than fuzzy-based algorithms.

From soft computing domain, fuzzy logic is one of the most widely used problem-solving methodologies used in control system. It provides a quicker approach to reach at a definite conclusion in a scenario with incomplete and non-numerical noisy information. It exhibits human intelligence reasoning behavior to handle incomplete data and unexpected situation. For wireless sensor network protocol design, fuzzy logic exhibits advantages in terms of transmission media characteristics and protocol performance making easy fuzzy representation and realistic. It can handle various wireless sensor networks uncertainties in an efficient manner. LEACH is the main protocol for clustering algorithms in wireless sensor networks. The cluster head selection is based on probabilistic approaches [3, 7, 8], and local information is used to make decision for cluster formation. Research activities are being carried out to overcome WSNs constraints and solve application and design issues.

Various centralized and distributed have developed to select cluster head and to form clusters. When network size increases, centralized schemes are not suitable. For large-scale networks, distributed schemes are more suitable. In distributed schemes, nodes within a cluster locally decide its cluster head and its cluster members on the basis of various parameters. Clustering protocol may be dynamic and static. In dynamic clustering [2, 7, 8, 9, 10, 11] protocol, operation has various rounds, and cluster is formed in each round. Because of this, energy depletion rate is high. Many clustering algorithms [2, 12, 13, 14, 15, 16] based on fuzzy logic that is useful for taking real-time decisions without exact information of system.

This paper proposes a grid-based approach to prolong WSNs lifetime. Grid head selection approach using fuzzy logic is non-probabilistic and fully distributed. Distributed grid head selection fixed routing scheme reduces extra communication cost with base station. No randomized function is used to generate a number to make decision to select grid head like other probabilistic approaches. Fuzzy logic is used to calculate fitness value for nodes for selecting as grid head. Residual energy of sensor node, distance from base station, is taken as a parameter to compute grid head probability value.

The rest of this paper is organized as follows. Related work, different clustering algorithms for WSN, along with their advantages and disadvantages has been given in Sect. 2. Our proposed scheme is described in Sect. 3. Simulation results and discussions have been presented in Sect. 4. Finally, the paper concludes by giving future scope in Sect. 5.

2 Related Works

LEACH [12] gave a hierarchical protocol for WSN which is one of the most widely used protocols by most of the researchers. In this, sensor nodes transmit to their cluster heads. Each cluster head aggregates and compresses the data, and finally forward this data to a remotely located centralized base station (BS). A stochastic algorithm in each round is used by a sensor node to determine whether it will be elected as a cluster head in each round. LEACH operation is comprised of a number of rounds. In each round, there is a set-up phase where clusters are formed. This is followed by a steady-state phase. In this, data are sent from sensor nodes to the cluster head and finally to the base station (BS). Clusters formation in LEACH is done by the use of a distributed algorithm. In this algorithm, sensor nodes make self-decisions without any centralized controlling authority. At the beginning of round r + 1, which starts at time t with probability P_i (t), each sensor node i elects itself as a cluster head. Selection of P_i (t) is made in such a manner that the expected number of cluster heads for this particular round is k. If there are N sensor nodes in a network, then the probability of becoming a cluster head by each node at the end of round r is given by the following equation:

$$P_i(t) = \begin{cases} \frac{k}{N - k*(r mod N/k)} : & C_i(t) = 1 \\ 0 & C_i(t) = 0 \end{cases} \tag{1}$$

where

$C_i(t)$ ← indicator function to determining a sensor node i has been a cluster node within the most recent (r mod N/k) rounds.

$C_i(t) = 0$ implies sensor node i as a cluster head. In this way, only nodes that have not already been elected as cluster head recently (i.e., C_i (t) = 1), and may be elected as cluster head in the next round, i.e., $r + 1$. If p represents probability factor, then by replacing k/N in Eq. (1) as p, we obtain a threshold value. This can be further used as a threshold value to select a node as a cluster head. Each sensor node chooses a random number between 0 and 1. If this number is less than a threshold value $T(n)$, then node is elected as a cluster head for the present round. The threshold value can be computed by the following equation:

$$T(n) = \begin{cases} \frac{P}{1 - p*(r mod 1/p)} & if\ n \in G \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where

G ← set of sensor nodes not been cluster head in the last $1/p$ rounds.

The major drawback of LEACH protocol is poor clustering formation. In this, some cluster heads are very close to the base station, while some others are very far away. Thereby, depletion of residual energy of cluster heads far away leads at a faster pace than the closer ones.

Gupta et al. [17] introduced fuzzy logic in wireless sensor networks. Fuzzy LEACH is derived from the LEACH base protocol. As per Eq. 2, cluster heads selection is done using a threshold value in the LEACH protocol which is fixed. To overcome the issue of poor clustering in LEACH, fuzzy logic plays a vital role in cluster head election mechanism, and it overcomes the problem encountered by pure probabilistic models. In Fuzzy–LEACH, three fuzzy descriptors are used, viz.,

- energy of node
- concentration and
- node centrality

They enhance process of cluster head election. Node centrality value reflects how central the node is to the cluster. This is computed using sum of the squared distance of other sensor nodes from a predefined sensor node. A base station performs the election of cluster head in each round for every sensor node to become a cluster head by examining three input fuzzy variable in F-LEACH. Twenty-seven fuzzy IF–THEN rules are predefined at the base station. F-LEACH assumes that the base station generates an appropriate cluster head as the base station has full information about the entire network. Each node of a cluster sends k bit message to the cluster head. The cluster head receives and processes it into *cnk* bit messages. The value $c \leq 1$ is known as compression coefficient. F-LEACH working model is almost similar as LEACH. In F-LEACH, two phases are required for the cluster head selection. Each one consists of a setup phase and followed by a steady state. Cluster heads are elected during setup phase using fuzzy logic knowledge processing, whereas various data processing is performed during steady-state phase.

Kim et al. [18] first gave a mechanism for the election of cluster head using two fuzzy variables as input, viz., energy and proximity distance. The election of cluster head by the use of fuzzy logic (CHEF) is given by proximity distance. It is defined as the summation of distances between the cluster head and sensor nodes inside radius (r) distance. In Eq. (3), r refers to the average radius of cluster. It is given as:

$$r = \sqrt{\frac{\text{area}}{\pi.n.P}} \tag{3}$$

where

$n \leftarrow$ number of sensor nodes in WSN.

$P \leftarrow$ battery level.

This approach elects a sensor node with highest energy level as a cluster head. In CHEF [18], two factors, viz., energy and local distance, are used to choose the suitable cluster heads thereby maximizing WSN lifetime. CHEF uses candidate method to take care of more cluster heads formation within r. This approach based on totally probabilistic model for selection of cluster head. Therefore, there is a chance that distribution of CH is not perfect and some node may not have any cluster head.

After making some enhancement in CHEF [18], Sharma et al. [14] proposed a novel fuzzy-based master cluster head election leach which is called F-MCHEL.

The election of cluster head is done in some different ways. Only one cluster head elects as a master cluster head having highest residual energy, instead of directly transmitting from a number of cluster heads to the base station. The approach makes use of two input parameters for FIS, i.e., fuzzy inference system. These are proximity distance and energy to the election of cluster head out of all selected cluster head.

3 Proposed Work

The idea of proposed scheme is initiated by merit and demerit of above discussed approaches. In all previous dynamic clustering schemes, we have two major demerits.

- Decide the cluster area by chosen cluster head in each round.

This process consumes valuable energy resource in processing and communicating among node to fix cluster head and cluster region.

- Overlapping of cluster head range.

It may possible that two nodes chosen as cluster head, close with each other. In our proposed approach, we have made an effort to eliminate these two shortcomings of existing schemes by proposing a novel grid-based dynamic cluster head selection scheme in WSN.

In our proposed approach, as shown in Fig. 2, network area is divided into grids where wireless sensor nodes are randomly deployed. The advantage of doing such kind of arrangement is that all sensor nodes have unique grid, which means that there will be no sensor nodes which belong to more than one grid. It reduces the energy consumption taken place at the time of dynamic clustering approach. In proposed approach, each grid contains almost equal number of homogeneous wireless sensor nodes, but we may also use this approach where each grid contains different numbers of wireless sensor nodes with heterogeneous nature of nodes. Fuzzy system rules used in the proposed model is given in Table 1.

3.1 Steps in Proposed Approach

Pseudocode of the proposed approach is given below.

```
Step 1: Consider network area $n \times n$ meters
Step 2: Divide network area into small size area ($kxl$ in meters) called as grid
Step 3: Randomly deployed sensor nodes each predefined grid
Step 4: For each round $r = 1$ to $r\_max$
```

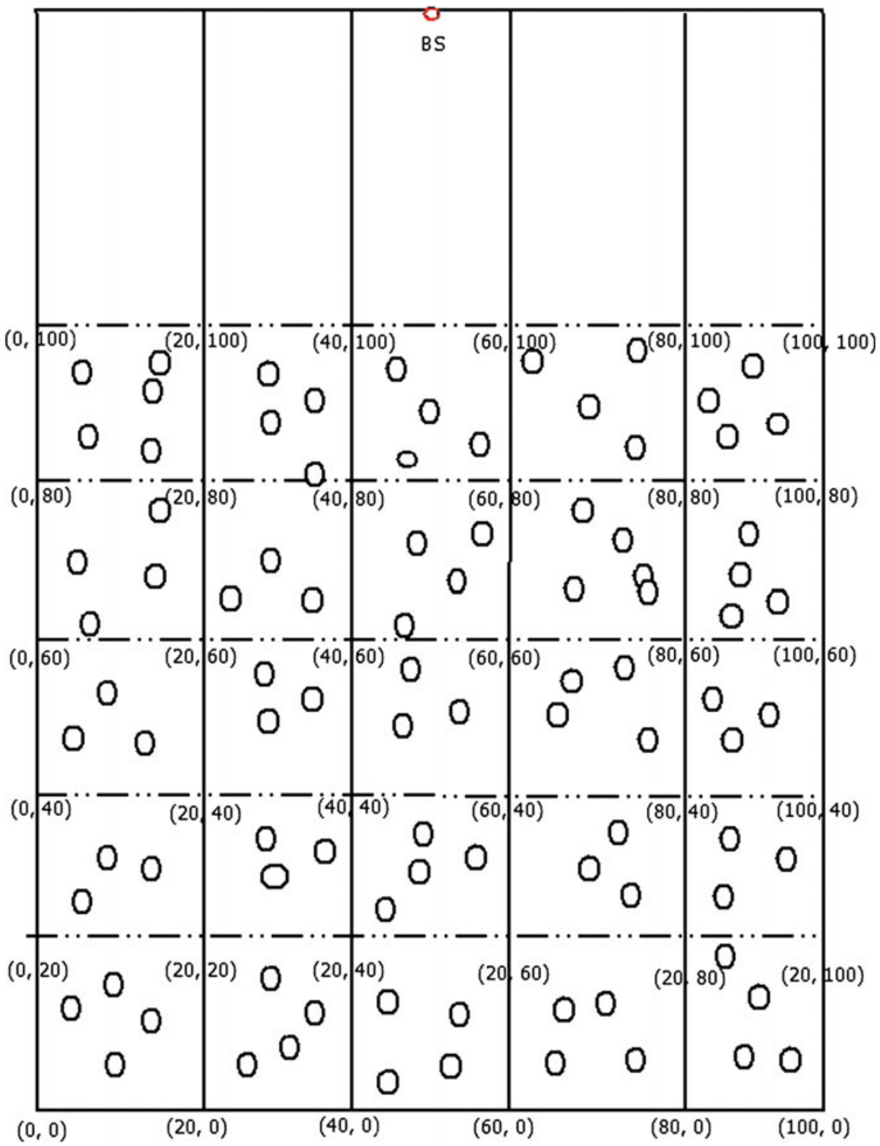


Fig. 2 Model of deployed nodes and base station

Calculate grid head chance of each node using fuzzy variables node residual energy and base station distance.

Step 5: Node having the maximum grid head chance selected as grid head
Step 6: Grid head receives data and forwards to next appropriate grid head or base station
Step 7: Go to step 4 until last node died

Table 1 Fuzzy system rule base

S. No	Base station distance	Residual energy	Grid head chance
1.	Close	Low	Low high
2.	Close	Medium	High
3.	Close	High	Very high
4.	Medium	Low	Rather medium
5.	Medium	Medium	Medium
6.	Medium	High	Low medium
7.	High	Low	Very low
8.	High	Medium	Low
9.	High	High	Rather low

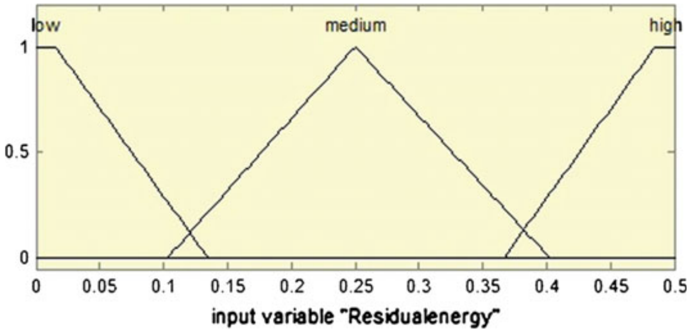


Fig. 3 Membership function for residual energy

3.2 *Input and Output for Fuzzy System*

Grid head chance calculation of each node of network is based on two fuzzy variables, and output is calculated based on the defuzzification method.

3.2.1 Input Variables

Residual Energy: Residual energy is calculated as remaining energy of sensor nodes till current round. In self-organization scheme, cluster head consumes much power than member nodes, because it has responsibility of data aggregation, processing of data, routing of data, etc. Membership function for residual energy is shown in Fig. 3.

Residual Energy of sensor node = Initial Energy of sensor node – consumed energy till current round.

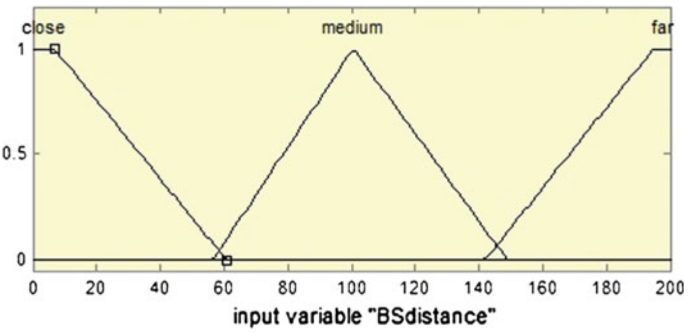


Fig. 4 Membership function for base station distance

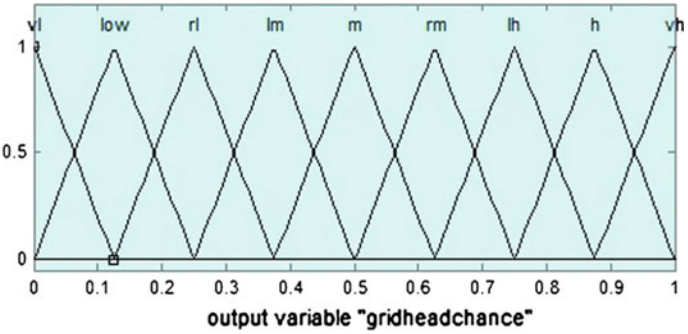


Fig. 5 Membership function for grid head chance vl = Very low, rl = Rather low, lm = Low medium, m = Medium, rm = Rather medium, lh = Low high, vh = Very high

Base Station Distance: Distance of grid head from base station. It is calculated using the distance square method. Membership function for base station distance is shown in Fig. 4.

3.2.2 Output

Output produced by fuzzy system must be crisp for real time implementation. In fuzzy inference system, we apply fuzzified value as input but output must be defuzzified value. In this approach, we use centroid method of defuzzifiation to get crisp output for selection of grid head. Membership function for grid head chance is shown in Fig. 5.

4 Simulation and Result Analysis

Proposed approach is simulated using the MATLAB simulation environment. The performance of this protocol is compared with the low energy adaptive clustering hierarchy, i.e., LEACH. The results show that the proposed approach, i.e., GAPFL extends the network lifetime, reduces the energy consumption requirement, and optimizes the number of cluster heads. Table 2 contains the simulation parameters.

Table 2 Simulation parameters

Parameter	Value
Network size	100 m * 100 m
Base station location	50 m, 140 m
Sensor nodes	200
Initial energy of nodes (E_0)	0.5 J
Packet size	4000 bit
Transmission energy (E_{TX})	50 nJ/bit
Receiving energy (E_{RX})	50 nJ/bit
E_{fs}	10 pJ/bit/m^2
$\mathrm{E}_{\mathrm{amp}}$	0.0013 pJ/bit/m^4

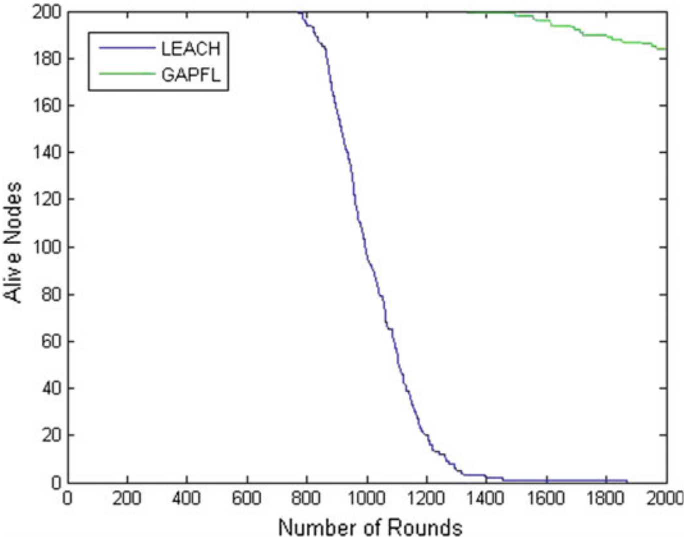


Fig. 6 Number of alive nodes in GAPFL versus LEACH

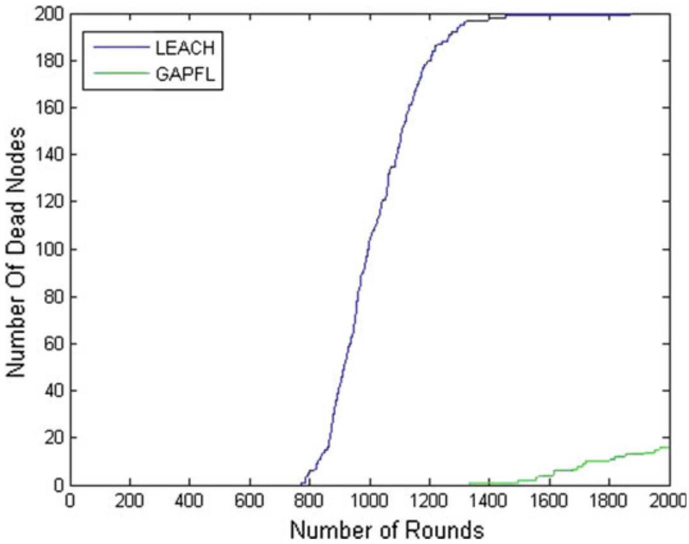


Fig. 7 Number of dead nodes in GAPFL versus LEACH

4.1 Simulation Results

MATLAB [17] tool is used to get the simulation results, as shown in Figs. 6 and 7. GAPFL protocol prolongs WSNs lifetime in terms on number of rounds. We simulated and compare GAPFL with LEACH in various number of rounds. Simulation result is shown below.

The above graph represents number of alive nodes in LEACH. In the above graph, till 2000 rounds, no nodes alive in LEACH protocol but using proposed, i.e., GAPFL approach near about 182 nodes is still alive.

5 Conclusion and Future Scope

In this paper, we proposed a grid-based approach to prolong wireless sensor networks lifetime using fuzzy logic which resolves the issues related with probabilistic approaches. Presented scheme is fully non-probabilistic, no random () function is used like CHEF [18]. In our approach, grid head selection is completely deterministic and decided by two variables, viz., base station distance and node residual energy using fuzzy logic. Grid head selection approach is based on nine if–then rules. The use of grid-based approach and fixed routing scheme results in reduction of power consumption in extra transmission and processing of data. Our main contribution in this paper is to reduce number of unnecessary communication and processing of data to prolong wireless sensor networks lifetime. Simulation results show that proposed scheme is helpful to maximize the lifetime of such networks.

Future research issues in our proposed approach will be dynamic grid formation and on demand grid head selection instead of grid head selection in each round. There are also scope for applying artificial intelligence concepts to find grid head and routing in wireless sensor networks.

References

1. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy- efficient communication protocol for wireless micro sensor networks. Published in the Proceedings of the Hawaii International Conference on System Sciences (2000)
2. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wireless Commun. **1**(4), 660–670 (2002)
3. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
4. Gajjar, S., Sarkar, M., Dasgupta, K.: Cluster head selection protocol using fuzzy logic for wireless sensor networks. Int. J. Comput. Appl. **97**, 38–43 (2014)
5. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**(3), 325–349 (2005)
6. Lindsey, N.S., Raghvendra, C., Shivlingam, K.M.: Data gathering algorithms in sensor networks using energy metrics. IEEE Trans. Parall. Distrib. Syst. **13**(9), 924–935 (2002)
7. Soro, S., Heinzelman, W.: Cluster head election techniques for coverage preservation in wireless sensor networks. Ad Hoc Netw. **5**, 955–972 (2009)
8. Thein, M.C.M., Thein, T.: An energy efficient cluster-head selection for wireless sensor networks. In: International Conference on Intelligent Systems, Modeling and Simulation, pp. 287–291 (2010)
9. Dastagheib, J., Oulia, H.: An efficient approach for clustering in wireless sensor network using fuzzy logic. Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT) **3**, 1481–1486 (2011)
10. Bandyopadhyay, S., Coyle, E.J.: An energy efficient hierarchical clustering algorithm for wireless sensor networks. Proc. IEEE INFOCOM **3**, 1713–1723 (2003)
11. Younis, O., Krunz, M., Ramasubramanian, S.: Node clustering in wireless sensor networks: recent developments and deployment challenges. IEEE Netw. **20**(3), 20–25 (2006)
12. ALMomani, I.M., Saadeh, M.K.: FEAR: fuzzy-based energy aware routing protocol for wireless sensor networks. Int. J. Commun. Netw. Syst. Sci. **4**, 403−415 (2011)
13. Natarajan, H., Selvaraj, S.: A fuzzy based predictive cluster head selection scheme for wireless sensor networks. In: International Conference on Sensing Technology, pp. 560–566 (2014)
14. Sharma, T., Kumar, B.: F-MCHEL: Fuzzy based master cluster head election leach protocol in wireless sensor network. Int. J. Comput. Sci. Telecommun. **3**(10), 8–13 (2012)
15. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**, 191–203 (1984)
16. Pal, N.R., Keller, J.M., Bezdek, J.C.: A possiibilistic fuzzy c-means clustering algorithms. IEEE Trans. Fuzzy Syst. **13**, 517–530 (2005)
17. Gupta, I., Riordan, D., Sampalli, S.: Cluster-head election using fuzzy logic for wireless sensor networks. In: Annual Conference on Communication Networks Services, pp. 255–260 (2005)
18. Kim, J.M., Park, S.H., Han, Y., Chung, T.M.: CHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks. In: International Conference on Advance Communications Technology, pp. 654–659 (2008)

Harmonics Minimization in Inverter Using Fuzzy Controller-Based Photovoltaic Cell

Subha Darsini Misra, Asish Ku. Nanda and Sudhansu Kumar Mishra

Abstract This paper represents a novel method to find total harmonic reduction (THD) for photovoltaic cell with fuzzy logic. The proposed method is based on fuzzy controller method. In contrast to previous method, it gives more THD, whereas fuzzy controller method gives less THD compared to P&O and I&C method. The proposed method involves some processes of fuzzy logic, such as fuzzification and defuzzification. In addition, some membership functions were involved. The simulation and programs are implemented in MATLAB. From the result, it has shown that fuzzy control technique achieves less THD under suitable temperature.

Keywords PV model · Fuzzy controller · THD · FFT analysis

1 Introduction

The energy from sun is essential to us, because it has the facility to the world directly or indirectly all most all its energy that can handle the World [1]. Photovoltaics (PVs) are arrays (combination of cells), which contain a photovoltaic material that is used to converts solar energy into electrical energy. There is a method for power control mechanism called maximum power point tracking (MPPT) by means of which we are able to increase the efficiency of operation of solar modules; hence, in the field of renewable energy, it plays a very effective role [2, 3]. Some conventional

S.D. Misra (✉) · A.Ku. Nanda
Department of Electrical Engineering, Rajdhaani Engineering College,
Bhubaneswar, Odisha, India
e-mail: tikinaiter@gmail.com

A.Ku. Nanda
e-mail: asishoec@gmail.com

S.K. Mishra
Department of Electrical and Electronics Engineering, BIT Mesra, Ranchi, Jharkhand, India
e-mail: sudhansu.nit@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_3

methods are used by researchers to track the maximum power point (MPP), such as perturbation and observation method, incremental conductance method [1], constant voltage method, and short-circuit method. The above methods are notable for tracking MPP under good conditions; hence, the fuzzy logic-controlled method is implemented. The fuzzy logic controller (FLC) is an effective method for improving MPP tracking and gives better total harmonic distortion (THD) as compared to other methods [4]. There is an optimal operating point in the solar panel that used to maximum power to the load [5–7]. Generally, we called that particular point as maximum power point tracking. As current verses voltage characteristics of the PV module are strongly influenced by solar irradiance and cell temperature, there is a non-linear variation of the locus of maximum power point tracking [8]. For tracking of MPP, there are several methods, such as perturb and observation (P&O), incremental conductance (IC), and fuzzy controlled method [4, 9].

2 PV Cell

Sunlight converts into electricity by photovoltaic system. Basic device of the photovoltaic system is known as photovoltaic cell which is the most basic generation part of the PV system. A silicon photovoltaic cell can be simulated using the mathematical model of a single diode. This model consists of a non-linear diode, a photo current source, and an internal resistance. Solar cell has a very important role that is a building block of solar panel. A photo voltaic module is formed by connecting many solar cells in series and parallel combinations. If only one single diode is considered for modeling, then the model is formed, as shown in Fig. 1, that consists of one diode, one current source having an internal resistance.

2.1 *Effect of Variation of Solar Irradiation*

The power verses voltage (P–V), current verses voltage (I–V), and power verses current (P–I) curves of the solar cell are depend on the values of solar irradiation. This solar irradiation causes the environmental changes, and keeps on fluctuating, which can be tracked by available control mechanism, and also for fulfil the

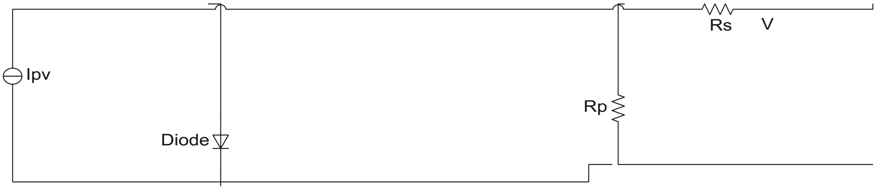


Fig. 1 Circuit diagram of PV cell

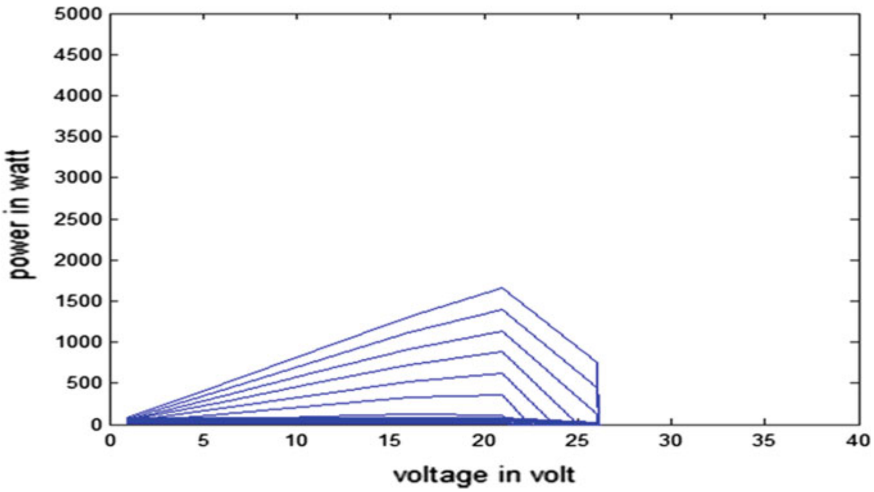


Fig. 2 P–V curve with solar irradiation

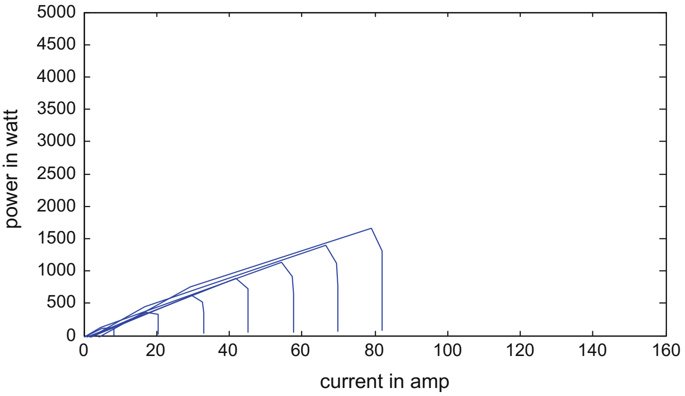


Fig. 3 P–I curve with solar irradiation

required demands, it can alter the working of solar cell. Variations of solar irradiation are shown in Figs. 2, 3, and 4, between P–V, P–I, and I–V, respectively.

3 Boost Converter

Boost converter is a DC–DC converter that is an electronic device, in which DC electrical power changes from one level to other level efficiently by the process of command of duty cycle signal. As like AC, DC simply cannot be stepped up or

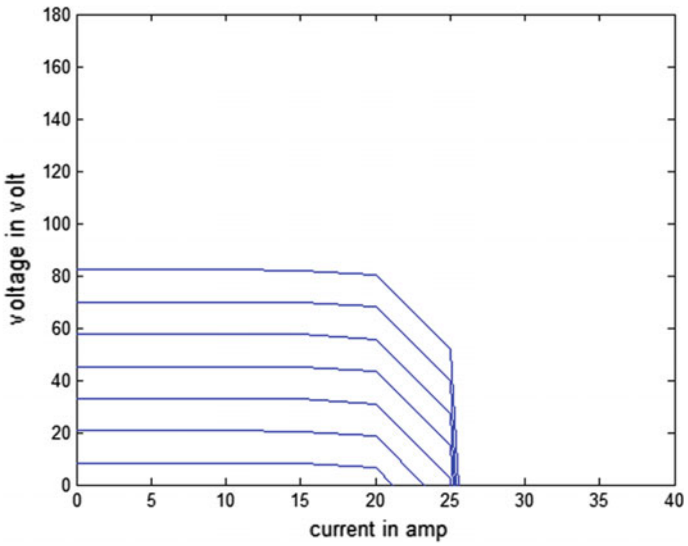


Fig. 4 V–I curve with solar irradiation

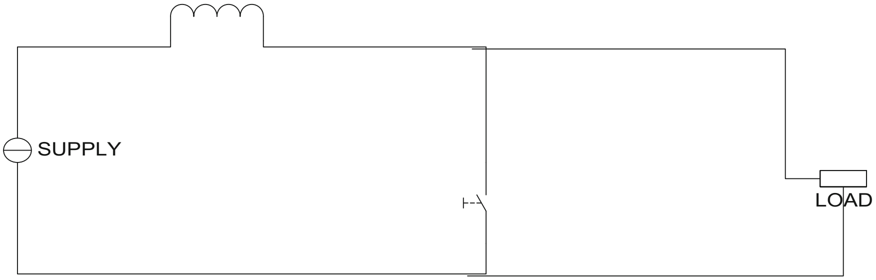


Fig. 5 Boost converter

down using a transformer; hence, above steps are required. Through a regulated converter DC–DC converter can be given the load to hold its maximum. Figure 5 shows the schematic diagram of a boost converter which is used for the purpose of DC–DC converter.

4 Fuzzy Logic Controller Implementation in PV

Fuzzy logic controller is implemented in this novel work. It has advantages that, it does not require any exact awareness about the mathematical model of the system, and also it is being robust technology [1]. Again fuzzy logic controller is more suitable for the non-linear system; hence, fuzzy logic has been implemented in this navel work, and we got a better performance in MPPT applications [5]. The methodology of fuzzy logic is well mimicking the capability of human being of inexact and uncertain judgment. There are three modules that are fuzzification, inference, and defuzzification in a typical fuzzy logic-based system [2, 3]. In the very first step, the fact base of the fuzzy system is established that is called fuzzification; here, the system input is first identified and then output and after it the appropriate if–then rule is identified, and finally, it derives a membership function (MF) using raw data. After fuzzification, an evaluation of all if–then rules is obtained in inference step, and finally, there is a determination of truth value is done. After completion of inference, defuzzification process is carried on. In this novel work, we use only a single input fuzzy logic controller.

5 Simulink Model and Simulation Results

See Figs. 6, 7, 8, 9, 10, 11, 12 and 13.

Comparison of THD

Perturbation and observation method	Incremental conductance method	Fuzzy controller method
THD = 117.82	THD = 76.53	THD = 62.56

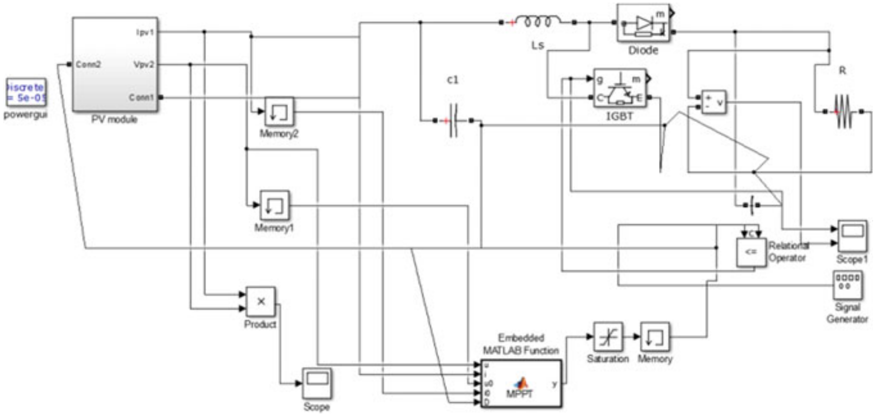


Fig. 6 Simulation diagram of I&C method

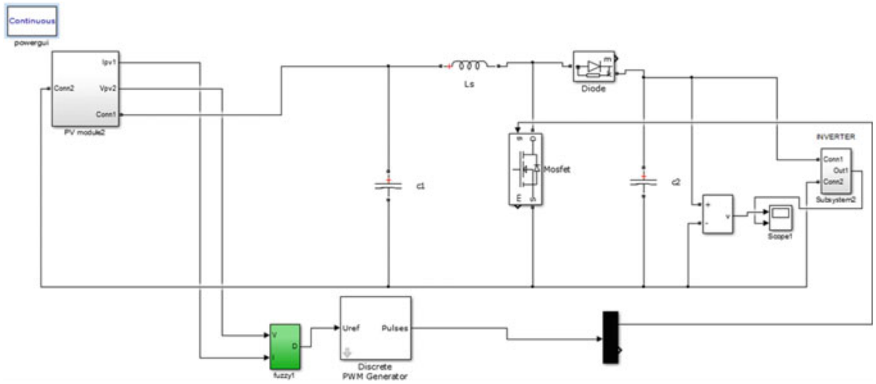


Fig. 7 Simulation diagram of fuzzy control method

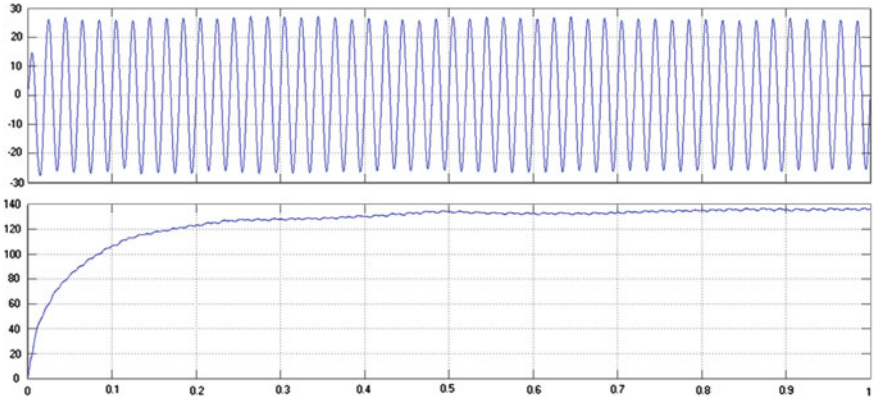


Fig. 8 Simulation result of P&O method

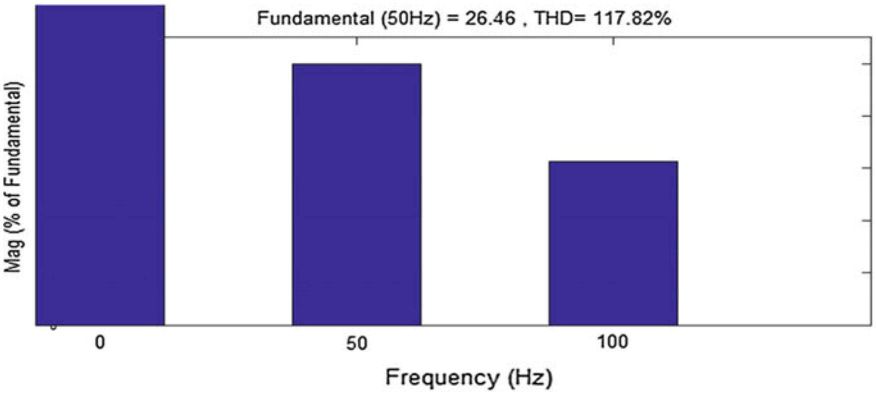


Fig. 9 FFT analysis of P&O method

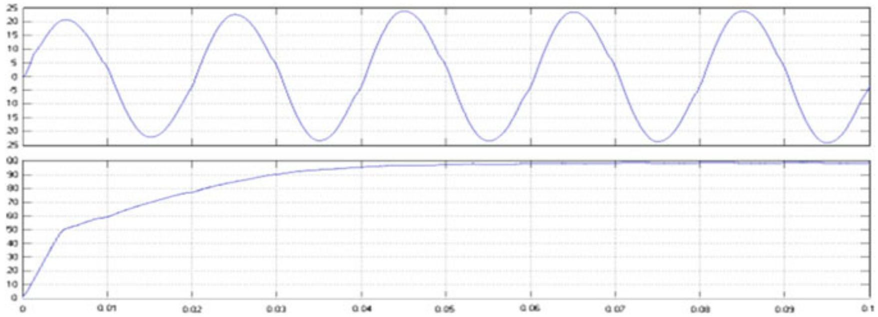


Fig. 10 Simulation result of I&C

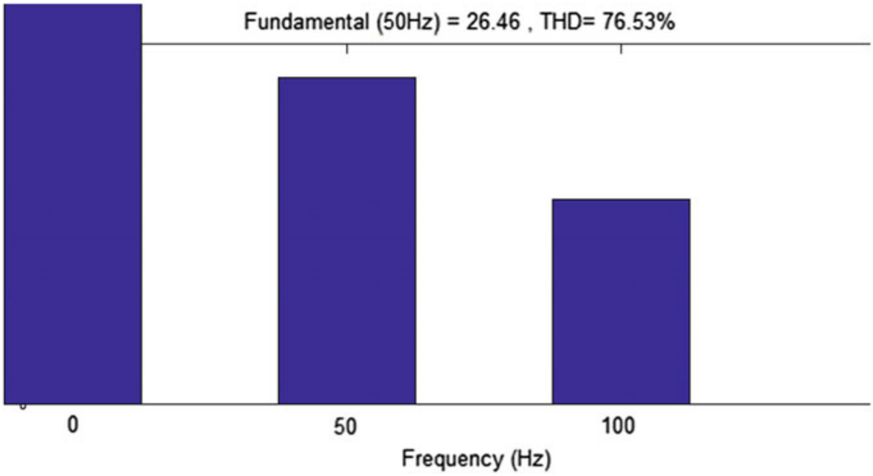


Fig. 11 FFT analysis of I&C method

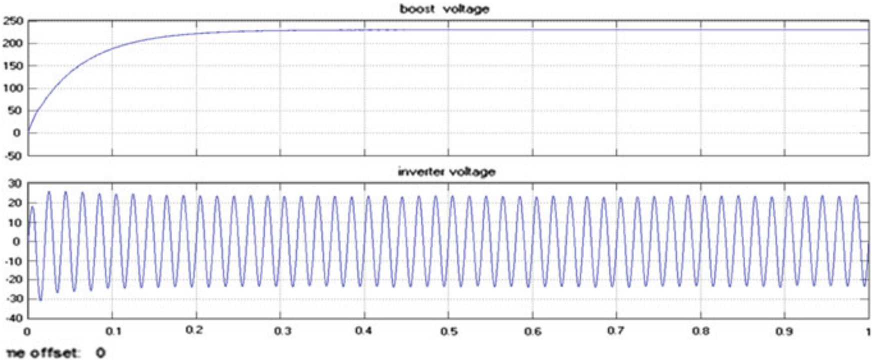


Fig. 12 Simulation result of fuzzy control method

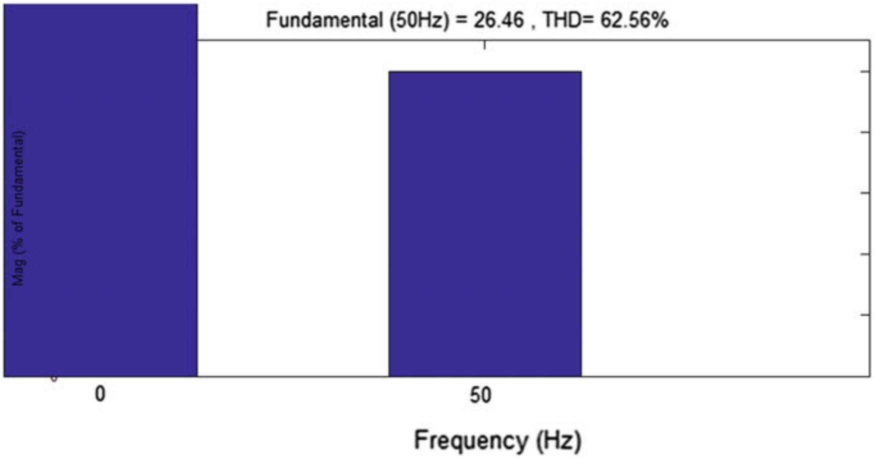


Fig. 13 FFT analysis of fuzzy control

5.1 The Simulation has been Done on the Basis of Data as

(a) In P&O method and incremental conductance method, R = 30 Ω, C = 500 μF, and L = 20 μH. The simulation result is shown in Fig. 8. In I&C method and incremental conductance method, R = 30 Ω, C = 500 μF, and L = 20 μH. The simulation block diagram of I&C method is shown in Fig. 6, and the result of simulation is shown in Fig. 10. The only difference is programmer written in the embedded system block.

(b) In fuzzy control method L, C is taken as 20 μH and 500 μF. In place of R load of boost converter, a single-phase inverter is connected in parallel with the LC filter, where L = 4 mH and C = 25 μF. The simulation block diagram of fuzzy logic control method is shown in Fig. 7, and the simulation result of this method is shown in Fig. 12. In the fuzzy control method in place of DC source, PV is taken, then it is connected to boost converter.

(c) In the boost converter in place of R-load, a single-phase inverter is taken, and it is connected to LC filter. In addition, the MATLAB version 2013 is used in the fuzzy control method.

(d) We have found that the FFT analysis of the above methods and fuzzy controller gives better THD Compare to P&O and I&C method, i.e., 62.56 %. Figure 9 shows the FFT analysis using P&O method, Fig. 11 shows the FFT analysis using I&C method, and Fig. 13 shows the FFT analysis using fuzzy control method.

6 Conclusion

This paper gives a brief idea about a single-phase photovoltaic (PV) inverter concept with fuzzy-based MPPT control. It presents the fuzzy logic control for controlling the MPPT of the PV system. The proposed algorithm is fuzzy logic control which is able to simulate. The simulation results shows that this system accept fuzzy parameter for faster response and good performance in FFT analysis. In addition, the result of simulation and experiment has shown that MPPT controller using fuzzy logic gives more power than the conventional method. The given system provides energy to offer improved output waveforms in steady state and gives better THD (total harmonic distortion). Therefore, the design and simulation of the fuzzy logic-based MPPT controller have been implemented in MATLAB 2013 version. Hence, the given method gives well response, such as fast responses, better THD for rapid irradiance, and temperature variations of photovoltaic cell. Thus, fuzzy controller gives better THD (total harmonic distortion), i.e., 62.56 %.

References

1. DKM, Deepa, L.: Implementation of fuzzy logic MPPT control and modified H-bridge inverter in photovoltaic system. In: IEEE Conference Publications on 13–14 Feb 2014
2. Altin, N., Ozdemir, S.: Three-phase three-level grid interactive inverter with fuzzy logic based maximum power point tracking controller. Energy Conv. Manage. **69**, 17–26 (2013)
3. Subramanian, V., Murugesan, S.: An artificial intelligent controller for a three phase inverter based solar PV system using boost converter. In: 2012 Fourth International Conference on Advanced Computing (ICoAC), pp. 1–7, 13–15 Dec 2012 (2012)
4. Misra, S.D., Mohapatra, T., Nanda, A.K.: Harmonic reduction using PV fuzzy with inverter. In: International Conference on CIDM, pp 99−108, 20−21 Dec 2014. doi:10.1007/978-81-322-2202-6_9
5. Salam, A., Salim, A., Zaharah, S., Hamidon, F., Adam, I.: Fuzzy logic controller for photovoltaic generation system in islanding mode operation. In: International Conference on Artificial Intelligence in Computer Science and ICT (AICS 2013), pp. 81–88. Langkawi, Malaysia, 25–26 Nov 2013
6. Sivagamasundari, M.S., Mary, P.M.: Fuzzy logic based cascaded H-bridge eleven level inverter for photovoltaic system using sinusoidal pulse width modulation technique. World Acad. Sci. Eng. Technol. Int. J. Electr. Robot. Electron. Commun. Eng. **8**(2), 443–447 (2014)
7. Sathy, S., karthikeyan, C.: Fuzzy Logic based Z-source inverter for hybrid energy resources. Int. J. Eng. Sci. Innov. Technol. **2**, 57–63 (2013). ISSN: 2319-59
8. Misra, S.D., Mohapatra, T., Nanda, A.K.: Harmonic reduction using PV fuzzy with inverter. In: International Conference on CIDM. doi:10.1007/978-81-322-2202-6_9 20-21 Dec 2014, pp. 99-108
9. Amjad, A.M., Salam, Z.: A review of soft computing methods for harmonics elimination PWM for inverters in renewable energy conversion systems. Renew. Sustain. Energy Rev. **33** (2014)

Fuzzy Logic-Based Unequal Clustering with On-Demand-Based Clustering Approach for a Better Lifetime of Wireless Sensor Network

D.R. Das Adhikary and Dheeresh K. Mallick

Abstract Clustering is a mechanism by which the network is partitioned into disjoint sets of groups to achieve energy efficiency and facilitate data aggregation in wireless sensor network. The clustering algorithm proposed earlier performs clustering on every round basis. On-demand clustering is a recent approach which eradicates the every round based clustering by performing clustering when it is required. But the on-demand clustering approaches proposed so far use equal clustering where clusters are of almost equal size thus, it suffers from the hot spot problem. Therefore, to solve the hot spot problem, a fuzzy logic-based unequal clustering approach is proposed along with an on-demand-based clustering. The proposed approach is implemented and compared with ECPF. Simulation results demonstrate that the proposed approach performs better, in terms of lifetime and other metrics.

Keywords Wireless sensor networks · Clustering · On-demand clustering · Fuzzy logic · Unequal clustering

1 Introduction

A group of sensors with wireless communication facility when deployed to observe an area of interest and facilitate report to an observatory system forms a network like structure, popularly known as wireless sensor network (WSN). Typically, a sensor is a tiny device with built-in sensing and wireless communication facility [1]. The role of a sensor is confined to sense its vicinity, process the sensed data, and send the data to the observatory system commonly known as the base station (BS) or sink [2].

D.R. Das Adhikary (✉) · D.K. Mallick
Department of Computer Science & Engineering, BIT Mesra, Ranchi 835215, India
e-mail: dibya@bitmesra.ac.in

D.K. Mallick
e-mail: dkmallick@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_4

The tininess of sensor comes with a cost of many limitations. One such limitation is battery, most of the sensors are battery powered, and due to the harsh condition of its deployment, recharge or replacement of the battery is not feasible [3]. Clustering is a mechanism employed to achieve energy efficiency in WSN [4]. It is also helpful in data aggregation and network management [5].

The operation of a typical clustering approach carried out in rounds. Most of the clustering algorithm proposed earlier select CH in every round. The action is justified, as it selects the CHs with best resources [6]. But the problem is a clustering process requires a lot of control information exchanges. To resolve the above-stated problem with clustering a few authors proposed on-demand based clustering. In on-demand-based clustering, the clustering process is triggered only when it is required. Usually, a selected CH set a threshold value for clustering process. When the remaining energy of the CH goes beyond the predefined threshold, it triggered the clustering process. But the problem arises, when the load of the CHs is unbalanced. Unbalanced load of CHs means unequal energy consumption rate. Therefore, the CHs have more energy consumption per round trigger the clustering process, whereas the energy level of the CHs with less energy consumption per round is far from the threshold. Unequal clustering is known for its load distribution properties [7]. In unequal clustering, clusters close to the sink are smaller in size to preserve energy for inter-cluster communication [8], and fuzzy logic has been used in WSN [9, 10] for resolving uncertainties in many scenarios.

Therefore, in this paper, we propose an on-demand clustering with unequal cluster range for CHs. In the proposed approach, the CHs are selected based on their energy status and then assigned a cluster range using a fuzzy inference system (FIS). The cluster range of the proposed approach is a factor of nodes remaining energy, distance to sink and centrality to neighbor.

The remaining of this paper is organized as follows. Section 2 briefly summaries some of the well-known clustering approach. In Sect. 3, the proposed methodology is described. Section 4 presents the detailed simulation and analysis of the proposed work. Last, conclusion is given in Sect. 5 of the paper.

2 Literature Review

Over the last decade, many clustering algorithm have been proposed. Low energy adaptive cluster hierarchy (LEACH) [11] is the first protocol to address the energy issue in WSN by applying clustering. In LEACH, a few nodes are randomly selected as CH, and the role is rotated among nodes to ensure fair distribution of workload in the network. Although LEACH is able to extend the network lifetime, but there are a few problems exist with it, which are addressed by many follow-up approach. In LEACH, the placement of CH is random, and the selection of CH is also random, so in LEACH-C or LEACH-centralized [12], the author addresses these issues. Multi-hop LEACH [13] tries to eradicate the direct communication between CHs and the base station by selecting an optimal path that adapts to multiple hops, whereas in energy LEACH [13], it considers the residual energy of the node, while choosing the CHs thus eliminates the randomized selection of CH of LEACH. Hybrid energy efficient distributed (HEED) protocol [14, 15] uses residual energy as a parameter while selecting CH.

To eradicate the unbalanced energy consumption problem, a few authors proposed unequal clustering [7, 8, 16] as an effective way of balancing the energy consumption. The principle of unequality in clustering was first discussed by Soro and Heinzelman [16]. They proposed a scheme called unequal clustering size (UCS). The main idea of UCS is to form adaptive clusters based on their distance to the sink. The authors in [7] proposed another unequal clustering mechanism called energy-efficient unequal clustering (EEUC) which is unlike UCS, and select CHs based on a competition. Unequal cluster-based routing (UCR) [8] is an extension to the EEUC. Like EEUC, in this approach as one move closer to the sink, the size of the cluster gradually decreases. By being so, it saves energy for the inter-cluster communication.

Fuzzy logic is used to handle vagueness or uncertainties in a system. Some clustering algorithms utilize fuzzy logic to resolve uncertainties in WSNs. First, Gupta et al. [9] used fuzzy logic to deal with uncertainties in the selection of CHs. Subsequently, Kim et al. [17] and Anno et al. [18] make use of fuzzy logic in the selection of CHs. Apart from this, Bagci et al. [10, 19] and the authors [20] proposed an unequal clustering approach, which uses fuzzy logic to assign distinct clustering range for CHs.

On-demand clustering is a recent trend in clustering WSN where the cluster is formed when required. Taheri et al. proposed energy-aware distributed clustering protocol using fuzzy logic (ECPF) [6] which introduced on-demand-based clustering, and it uses fuzzy logic to select the CHs.

3 The Proposed Approach

The on-demand clustering algorithms proposed earlier shows excellent improvement over the every round based clustering approach. But the on-demand clustering approach proposed so far uses equal cluster range for CHs. In a multi-hop inter-cluster communication scenario, the CHs near the sink are over burdened with relay traffic and lose significant amount of energy in relaying data, and thus, the CHs near sink triggered the clustering process prematurely.

Hence, based on the above observation, the proposed algorithm constructs a more balanced clustering scheme by considering the remaining energy. The clustering process of the proposed approach is on a demand basis. The CHs are selected based on local information. A node waits a certain amount of time before declare itself as CH where the wait time is inversely proportional to nodes remaining energy. An FIS is used for calculation of cluster range of CHs.

The operation of the proposed methodology is divided into rounds. For each round, the proposed approach consists of two stages: cluster setup phase and steady-state phase. In the proposed approach, the cluster setup phase is followed my multiple steady-state phase. A neighborhood discovery phase is occurred once at the time of network deployment before the actual operation of the proposed approach begins.

3.1 Neighborhood Discovery Phase

The algorithm starts with the neighborhood discovery phase, in which the sink broadcasts a Hello message. On receiving this Hello message, a node can calculate its distance from the sink. Receiving node of the Hello message, broadcast a Hello Reply message, consists of sender id, within a range $R_{\max}$, where $R_{\max}$ is the maximum cluster range. Receiving nodes of the Hello_Reply message add the sender as its neighbor. Whenever any node has remaining energy below a given threshold, it will broadcast itself as dead by sending a dead message. The receiving nodes of dead message update their neighborhood information.

3.2 Cluster Setup Phase

The cluster setup phase of the proposed approach consists of:

- Selection of candidate CHs.
- Assigning appropriate cluster range to candidate CHs.
- Selection of the CHs from the set of candidate CHs.
- Assigning non-CH nodes to clusters.

After the end of the neighborhood discovery phase, each node waits for a Wait_Time before broadcast the Candidate_CH message. The Wait_Time has been calculated as follows:

$$\text{Wait\_Time} = \frac{1}{\text{Remaining Energy(ERE)}} \tag{1}$$

When the Wait_Time is over, the candidate CH broadcast a Candidate_CH message in its cluster range. The cluster range calculation of the proposed approach is outlined in the following section.

3.3 Fuzzy Logic-Based Cluster Range Calculation

To handle ambiguity in cluster range selection, in this paper, we used fuzzy inference systems (FIS). In our proposed approach, we have used the Mamdani method of fuzzy inference technique [21]. The input variables of the FIS are remaining energy, distance to sink, and centrality. The variables are defined as follows.

- Remaining energy (ERE)—Energy level remaining of the node.
- Distance to sink (D2S)—Nodes distance (Euclidean distance) to sink.
- Centrality (Cen)—A value which classifies the nodes based on the distance from the neighbors with proportion to network dimension.

The first input fuzzy set is remaining energy; Fig. 1 shows membership functions of input variable remaining energy. The fuzzy sets in the form of linguistic variables include low, medium, and high. The second input fuzzy set is distance to sink of the node; Fig. 2 shows membership functions of input variable distance to sink. The fuzzy sets in the form of linguistic variables include near, not too far and far. The third input fuzzy set is node centrality; Fig. 3 shows membership functions of input variable centrality. The fuzzy sets in the form of linguistic variables include close, acceptable, and far. The only fuzzy output variable of the FIS is Range Factor; Fig. 4 shows membership functions of output variable range factor. There

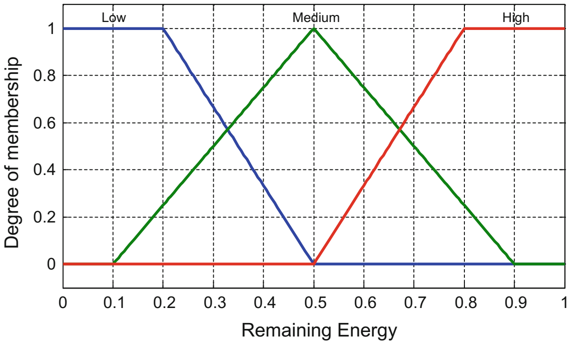


Fig. 1 Membership functions of input variable remaining energy

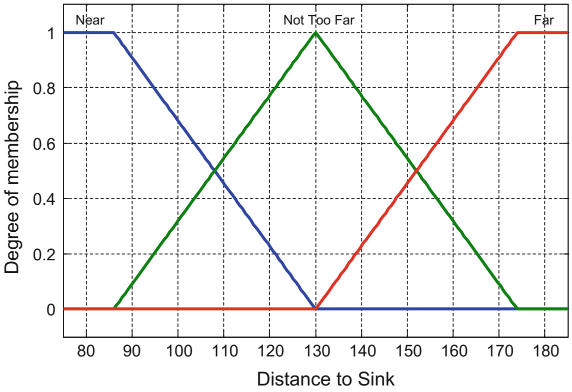


Fig. 2 Membership functions for input variable distance to sink

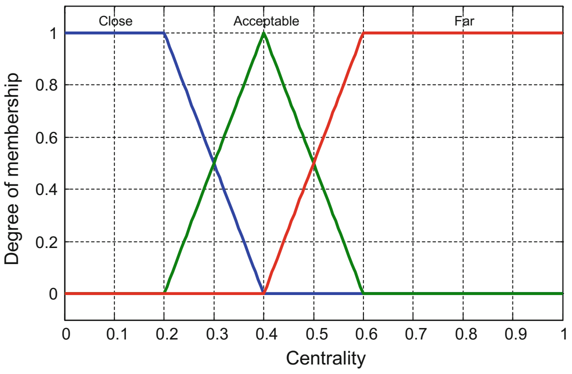


Fig. 3 Membership functions for input variable centrality

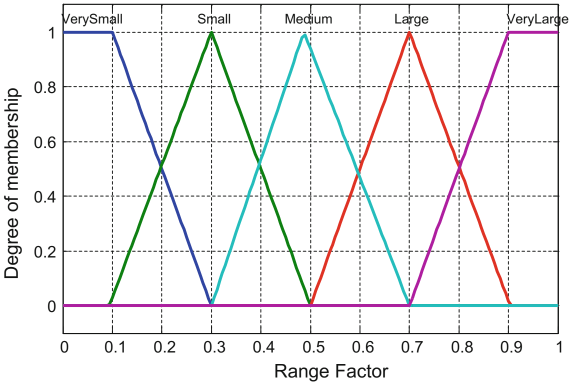


Fig. 4 Membership functions for output variable range factor

are five linguistic variables for the output fuzzy set fitness; they are very small, small, medium, large, and very large.

Fuzzy inference unit uses the fuzzy “if–then” rules for the calculation of output. There are twenty-seven fuzzy mapping rules defined in Table 1. By applying these fuzzy “if–then” mapping rules on the inputs, the fuzzy output range factor has been generated. The output fuzzy variable has no use unless defuzzified to a single crisp value. In this paper, the centre of gravity (COG) method [22] has been used for the defuzzification.

The output of the FIS is Range Factor. Range factor is the variable that influences the cluster range of the candidate CHs. The effective cluster range of the candidate CHs is calculated as follows:

$$\text{The Effective Cluster Range (ECR)} = \text{Range Factor} \times R_{\max} \tag{2}$$

The clustering process of the proposed approach is on a demand basis. When a CH has left with energy less than a given threshold, it reports to the sink to conduct a cluster setup.

Table 1 The fuzzy rule base

SI No	ERE	D2S	Cen	Range factor
1	Low	Near	Far	Very small
2	Low	Near	Acceptable	Very small
3	Low	Near	Close	Small
4	Low	Not too far	Far	Very small
5	Low	Not foo far	Acceptable	Small
6	Low	Not too far	Close	Small
7	Low	Far	Far	Small
8	Low	Far	Acceptable	Medium
9	Low	Far	Close	Medium
10	Medium	Near	Far	Small
11	Medium	Near	Acceptable	Medium
12	Medium	Near	Close	Medium
13	Medium	Not too far	Far	Large
14	Medium	Not too far	Acceptable	Medium
15	Medium	Not too far	Close	Small
16	Medium	Far	Far	Large
17	Medium	Far	Acceptable	Medium
18	Medium	Far	Close	Large
19	High	Near	Far	Medium
20	High	Near	Acceptable	Large
21	High	Near	Close	Large
22	High	Not too far	Far	Medium
23	High	Not too far	Acceptable	Large
24	High	Not too far	Close	Large
25	High	Far	Far	Large
26	High	Far	Acceptable	Very large
27	High	Far	Close	Very large

3.4 On-Demand Clustering

Most of the clustering algorithm proposed earlier select CH in every round. Their action is justified, as it selects the CHs with the best resources. But the problem is a clustering process required a lot of control information exchanges. Therefore, in this paper, an on-demand clustering process similar to [6] is proposed, and the clustering phase is triggered by the sink when the remaining energy of a CH goes beyond a given threshold.

After the end of each cluster setup phase, each selected CH save its energy status in a variable ETH and stored it in the memory. At the end of every steady-state phase if the remaining energy of any of the CHs, i.e., ERE falls below $\alpha \times$ ETH (α floating point number with range $0 \leq \alpha \leq 1$), it set the election_bit_field in the packet as true and send it to the sink. On receiving a packet if the sink found that election_bit_field set as true it broadcast a pulse signal, that inform the nodes to conduct a cluster setup. The on-demand clustering process mitigates the control packet overhead required by consecutive cluster setup phase and, consequently, decreases the overall energy consumption.

The rest of the algorithm is exactly identical to ECPF [6], so we exclude the explanation in this section.

4 Result and Discussion

In this section, the results of the simulation experiments are presented to illustrate the effectiveness of the proposed method. For simplicity, in this work, we used an ideal Mac layer and error free communication link, and the energy is consumed whenever a sensor sends or receives data or performs data aggregation. We compare the proposed algorithm with ECPF. The simulations are performed using MATLAB software. The details of the parameters used in the simulation and their values are given in Table 2. Apart from these parameters, a few assumptions are made about the system as stated in [6]. In this paper, we use the first-order radio model as stated in [12] to model the energy dissipation.

The sensor nodes are randomly deployed over a 100 m × 100 m area, with just one base station located at position (50 m, 175 m). The simulation is performed with 100, 200, 300, and 400 set of nodes. The metrics used for comparison are the lifetime of the network in terms of rounds, first node dies (FND), half of nodes alive (HNA), last node dies (LND), and the energy spent per round. The value of α is set to 0.8 as described in [6], and the value of R_{max} is set to 80 m, and we also found it suitable for our configuration.

In WSN, energy is the key factor that decides the effectiveness of an approach. A node spent its energy for various kinds of operations. Figure 5 represent the

Table 2 Parameter used for simulation

Parameter	Value
Initial energy per node	2 J
ϵ_{fs}	10 pJ/bit/m2
ϵ_{mp}	0.0013 pJ/bit/m4
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
d_0	87 m
Idle power	13.5mW
Sleep power	15 μW
Round time	20 s
Data packet size	100 byte
Control packet size	25 byte
Aggregation ratio	10 %

Fig. 5 Total energy dissipated per round

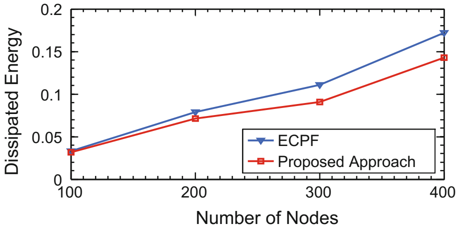


Fig. 6 Network lifetime (FND)

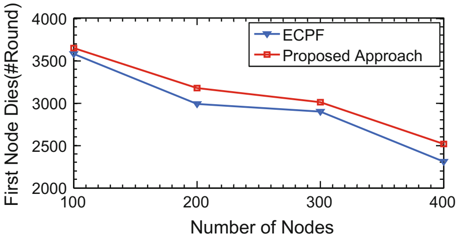
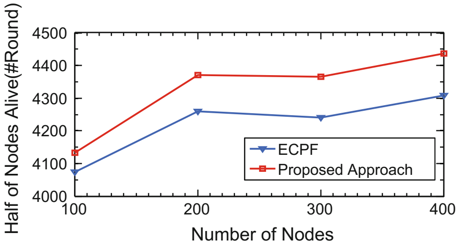


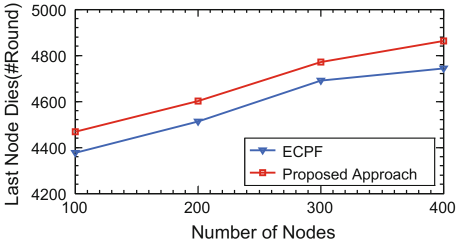
Fig. 7 Network lifetime (HNA)



average energy consumption per round. It can be observed that the performance of the proposed algorithm is better than ECPF. The fuzzy logic-based unequal clustering of the proposed approach distributes the network load optimally over the selected CHs. Thus, it solves the immature triggering of the clustering process of ECPF. As the number of clustering process decreased, consequently, the average energy consumption per round decreased.

Achieving a better network lifetime is the ultimate goal of clustering in WSN. The comparison of network lifetime of the proposed approach with ECPF is shown in Figs. 6, 7 and 8. The comparison of network lifetime is based on three metrics

Fig. 8 Network lifetime (LND)



namely FND, HNA, and LND. It can be observed from the figures that the performance of the proposed approach is well ahead of ECPF. The unequal clustering approach creates stable clusters, which in turn minimizes the number of election, and as election contribute to the energy consumption, the performance of the proposed approach improved significantly.

It can be noted that the performance of the proposed approach is better than ECPF irrespective of number of nodes and network dimension; moreover, it gives a better result with a large number of nodes. Therefore, it proves that the proposed approach is scalable irrespective of network dimension and node size.

5 Conclusion

Better lifetime is a design goal that every clustering approach tried to accomplish. This paper presents a fuzzy logic-based unequal clustering approach along with an on-demand-based clustering to efficiently improve the lifetime of the WSN. The fuzzy logic-based unequal cluster range approach efficiently assigns different cluster range to CHs based on its remaining energy, distance to sink, and centrality. We have examined the algorithm thoroughly by doing extensive simulation. According to the simulation result, the proposed method gets a better performance compared to the CHEF. The result is shown that the proposed method is scalable and improves the network lifetime significantly irrespective of network dimension. There are several assumptions made about the system that provides the scope for further studies.

References

1. Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. Commun. Mag. IEEE **40**(8), 102–114 (2002)
2. Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Comput. Netw. **38**(4), 393–422 (2002)
3. Abbasi, A.A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Comput. Commun. **30**(14), 2826–2841 (2007)
4. Pal, V., Singh, G., Yadav, R. P.: SCHS: Smart cluster head selection scheme for clustering algorithms in wireless sensor networks (2012)
5. Wei, C., Yang, J., Gao, Y., Zhang, Z.: Cluster-based routing protocols in wireless sensor networks: a survey. In: 2011 International Conference on Computer Science and Network Technology (ICCSNT), vol. 3, pp. 1659–1663. IEEE (2011)
6. Taheri, H., Neamatollahi, P., Younis, O.M., Naghibzadeh, S., Yaghmaee, M.: An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy-logic. Ad Hoc Netw. **10**(7), 1469–1481 (2012)
7. Li, C., Ye, M., Chen, G., Wu, J.: An energy-efficient unequal clustering mechanism for wireless sensor networks. In: 2005. IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, pp. 8. IEEE (2005)

8. Chen, G., Li, C., Ye, M., Jie, W.: An unequal cluster-based routing protocol in wireless sensor networks. Wireless Netw. **15**(2), 193–207 (2009)
9. IGupta, I., Riordan, D., Sampalli, S.: Cluster-head election using fuzzy-logic for wireless sensor networks. In: Proceedings of the 3rd Annual Communication Networks and Services Research Conference, pp. 255–260, May 2005
10. Bagci, H., Yazici, A.: An energy aware fuzzy approach to unequal clustering in wireless sensor networks. Appl. Soft Comput. **13**(4), 1741–1749 (2013)
11. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii international conference on System sciences, pp. 1–10, January 2000
12. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wireless Commun. **1**(4), 660–670 (2002)
13. Xiangning, F., Yulin, S.: Improvement on LEACH protocol of wireless sensor network. In: International Conference on Sensor Technologies and Applications, SensorComm 2007, pp. 260−264, October 2007
14. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. **3**(4), 366–379 (2004)
15. Younis, O., Fahmy, S.: Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, vol. 1, March 2004
16. Soro, S., Heinzelman, W.B.: Prolonging the lifetime of wireless sensor networks via unequal clustering. In: Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, 2005. 8pp. IEEE (2005)
17. Kim, J.M., Park, S.H., Han, Y.J., Chung, T.M.: CHEF: cluster head election mechanism using fuzzy-logic in wireless sensor networks. In: 10th international conference on Advanced communication technology, ICACT 2008, vol. 1, pp. 654–659, February 2008
18. Anno, J., Barolli, L., Xhafa, F., Durresi, A.: A cluster head selection method for wireless sensor networks based on fuzzy-logic. In: IEEE Region 10 Conference TENCON 2007, pp. 1–4, Oct 2007
19. Bagci, H., Yazici, A.: An energy aware fuzzy unequal clustering algorithm for wireless sensor networks. IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1–8, July 2010
20. Seyyit Alper, S., Bagci,H., Yazici, A.: MOFCA: multi-objective fuzzy clustering algorithm for wireless sensor networks. Appl. Soft Comput. **30**, 151–165 (2015)
21. Wang, L.-X.: A course in fuzzy systems. Prentice-Hall press, USA (1999)
22. Runkler, T.A.: Selection of appropriate defuzzification methods using application specific properties. Fuzzy Syst. IEEE Trans. **5**(1), 72–79 (1997)

Intuitionistic Fuzzy-Based Multi-Attribute Decision-Making Approach for Selection of Inventory Policy

Mahuya Deb and Prabjot Kaur

Abstract Selection of inventory control policies is of great concern in the dynamic business environment as they are the drivers of success towards achieving a competitive advantage in terms of cost, quality and service. Inventory policy selection is affected by a number of criterions some of which may be cost, demand and lead time which are quite conflicting in nature. Therefore, inventory control policy selection can be categorised as a Multi-Criteria Decision-Making technique involved in evaluating a set of alternatives through which the enterprises need to identify optimal inventory policy. This research develops a decision model which is focused towards evaluation, ranking and selection of inventory policies based on these conflicting criteria using intuitionistic fuzzy numbers.

Keywords Inventory policies · MCDM · Intuitionistic fuzzy numbers

1 Introduction

The intense competition in today's operating environment has made it increasingly important for industrial enterprises to continuously seek the best practices towards managing their operations and, eventually, differentiate themselves from their competitors. In particular, the optimization of production- and inventory-related decisions provides a vital step towards better fulfilment of customers' needs at a minimum cost. Therefore, in this situation inventory policy selection constitutes an important purchasing function for the survival and sustenance of the manufacturing sector which constitutes an important element of the modern supply chain and is, therefore, vital for the progress and development of any industry. With the

M. Deb (✉) · P. Kaur
Department of Mathematics, Birla Institute of Technology, Mesra,
Ranchi 835215, Jharkhand, India
e-mail: mahuya8@gmail.com

P. Kaur
e-mail: tinderbox_fuzzy@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_5

inventory model involving various types of cost, selection of an optimum model could address the relevant aspects of the system or reality which is of concern to the decision maker. Therefore, it is imperative to opt for an inventory policy which aims at minimising the total cost, lead time and maximise customer satisfaction. Through an extensive study of the literature, it is revealed that quite a few number of models related to inventory have been proposed over the years. To name a few, viz. economic order quantity (EOQ)(IP1), just in time (JIT)(IP2), vendor managed inventory (VMI)(IP3) and monthly policy (MP)(IP4) finds mention in this paper based on the five selection criteria, namely 'unit cost', 'holding cost', 'procurement cost', 'shortage cost', demand and 'lead time'. Thus, the inventory policy selection falls under the multi-criteria group decision-making problem which deals with vague and imprecise criteria quite conflicting in nature and these data could be obtained either by simulation technique or through analysis of linguistic subjective judgments defined by fuzzy sets.

Brooking [4] discussed an inventory wherein the system could be improved through costs in an environment which was automatic. Arcelusand and Srinivasan [1] looked into the cost structures of an inventory system and their implication on inventory policy. Chen and Hwang [6] tried to minimise the average cost with the help of stochastic lead time. Cetinkaya and Parlar [5] considered an inventory model with backorder considering the fixed and proportional backorder cost, stochastic demand, single product and periodic review. An algorithm was developed by Zhou et al. [22] which computed the optimal values for the parameters in infinite time horizon. Thereafter, researchers from time to time have presented widespread MCDM methods which could provide viable and effective solution to various real-life problems [6, 14–18, 21, 22]. The concept of fuzzy set introduced by Zadeh [19] has a membership function with the non-membership function being one minus the membership function. However, in reality when a person is asked to express his/her preference degree to an object, it is possible that he/she is not so sure about it, that is, there usually exists a hesitation or uncertainty about the degree, and there is no means to incorporate the hesitation or uncertainty in a fuzzy set. Later, Brooking [4] gave a generalised form of fuzzy set, called Intuitionistic Fuzzy Set (IFS), which is characterised by a membership function and a non-membership function. The intuitionistic fuzzy set has received more and more attention since its appearance. A ranking method with the help of intuitionistic fuzzy number was developed by Mitchell [12] and Nayagam et al. [13]. A new methodology for ranking TIFNs using multi-attribute decision-making methods (MADM) was developed by Zhang and Nan [20]. Expert opinion and fuzzy Distance-Based Approach (DBA) methodology was proposed by Gupta et al. [8] for ranking inventory policies. The concept of value and ambiguity was used by Li [9] for ranking of inventory policies.

In this research paper, a new methodology is proposed using Intuitionistic Fuzzy Number for evaluation and ranking of inventory ordering policies. Based on opinion from experts in accordance with selection criterion, inventory policies are evaluated and ranked using value and ambiguity indices of Intuitionistic Fuzzy Number as proposed by Li [9]. This methodology employs simple mathematical

Table 1 Selection Criteria for Inventory Policies

Criterions	Characteristics
Unit cost (C1)	This cost refers to the purchase price of items and if it remains constant does not affect the inventory control decisions
Holding cost (C2)	Also known as the inventory carrying cost, they generally range between 15 and 30 percent of the price of the concerned product and vary from plant to plant. These are incurred to achieve reduction in costs of material, ordering and control systems
Procurement cost (C3)	Termed as preparation costs, these are incurred in connection with ordering and procurement
Shortage cost (C4)	This cost is incurred when the demand goes unfulfilled and provides the most formidable estimation problem. This cost could be ten times the holding cost. It depends on the quantity of demand unsatisfied but not on the duration
Demand (C5)	It refers to the consumption of the item and are considered to be either known quantities or unknown
Lead time (C6)	It is the average actual time gap between the customers placing an order and till it is received by the customer

computation and is applicable to decision-making problem. A numerical example is examined to demonstrate the implementation process and applicability of the method proposed in this paper.

The organisation of the paper is as follows: In Sect. 1, the various criterions for evaluations are defined. Thereafter, the concept of TIFNs is introduced in Sect. 2 where the values and ambiguities of the membership degree and the non-membership degree for TIFNs as well as the value index and ambiguity index are defined. Section 3 deals with the value and ambiguity-based ranking method to solve multi-attribute decision-making problems in which the ratings of alternatives on attributes are expressed using TIFNs. Section 4 deals with a numerical example and lastly the conclusion is provided in Sect. 5.

Inventory policy selection and evaluation involves a process which can be compared, monitored over time, and hopefully improved through some measures. This section defines a list of various attributes relevant to inventory policies, which could be utilised for ranking. However, it is not exhaustive and may require additional inputs depending upon the purpose and choice of the decision maker (Table 1).

2 Preliminaries on Intuitionistic Fuzzy Sets

We quote a few definitions of triangular intuitionistic fuzzy numbers (TIFNs) as defined by Li [10] relevant to the present work.

A triangular intuitionistic fuzzy number $\tilde{A} = (a_1, a_2, a_3; w_{\tilde{a}}, u_{\tilde{a}})$ is a special intuitionistic fuzzy set on the real number set R whose membership and non-membership are defined as follows:

$$\mu_{\tilde{A}}(x) = \left\{ \begin{array}{l} \dfrac{(x - a_1)w_{\tilde{a}}}{a_2 - a_1}, a_1 < x \leq a_2 \\ w_{\tilde{a}},\ x = a_2 \\ \dfrac{(a_3 - x)w_{\tilde{a}}}{a_3 - a_2}, a_2 < x \leq a_3 \end{array} \right\}$$

&

$$\nu_{\tilde{A}}(x) = \left\{ \begin{array}{l} \dfrac{a_2 - x + u_{\tilde{a}}(x - a_1)}{a_1 - a_2}, a_1 < x \leq a_2 \\ u_{\tilde{a}},\ x = a_2 \\ \dfrac{x - a_2 + u_{\tilde{a}}(a_3 - x)}{a_3 - a_2}, a_3 < x \leq a_2 \end{array} \right\}$$

$$0 \leq w_{\tilde{a}} + u_{\tilde{a}} \leq 1$$

A triangular intuitionistic fuzzy number $\tilde{A} = (a_1, a_2, a_3; w_{\tilde{a}}, u_{\tilde{a}})$ where the values $w_{\tilde{a}}\, and\, u_{\tilde{a}}$, respectively, represent the membership degree which is maximum and minimum non-membership degree so that they satisfy the condition $w_{\tilde{a}} \in [0, 1]\ and\, u_{\tilde{a}} \in [0, 1]$ and $0 \leq w_{\tilde{a}} + u_{\tilde{a}} \leq 1$.

Analogously it is assumed that $\pi_{\tilde{A}}(x) = 1 - \mu_{\tilde{A}}(x) - \nu_{\tilde{A}}(x)$ which is known as the indeterminacy degree of the TIFN $\tilde{A}$. It is commonly referred to as the hesitation index. A TIFN describes some quantity "about a_2" which is expressed with values a_1 and a_3 specified with certain degrees of membership and non-membership. a_2 is the mean around which there is a left spread and right spread.

Therefore,

(i) The value of the TIFN $\tilde{A}$ for membership function is defined as follows:

$$V_\mu(\tilde{A}) = \frac{(a_1 + 4a_2 + a_3)w_a^2}{6} \tag{1}$$

Likewise, the value of the TIFN $\tilde{A}$ for non-membership function is as follows:

$$V_\nu(\tilde{A}) = \frac{(a_1 + 4a_2 + a_3)(1 - u_a)^2}{6} \tag{2}$$

Also

(ii) The ambiguity of the TIFN $\tilde{A}$ for membership function is as follows:

$$A_{\mu}(\tilde{A}) = \frac{(a_3 - a_1)w_a^2}{6} \tag{3}$$

The ambiguity of the TIFN $\tilde{A}$ for non-membership function is as follows:

$$A_{\nu}(\tilde{A}) = \frac{(a_3 - a_1)(1 - u_a)^2}{6} \tag{4}$$

For any given weight $\theta \in [0, 1]$, the θ weighted value and θ weighted ambiguity of the TIFN $\tilde{A}$ are defined as follows:

$$\begin{aligned} V_{\theta}(\tilde{A}) &= \theta\, V_{\mu}(\tilde{A}) + (1 - \theta)\, V_{\nu}(\tilde{A})\ and \\ A_{\theta}(\tilde{A}) &= \theta A_{\mu}(\tilde{A}) + (1 - \theta) A_{\nu}(\tilde{A}) \end{aligned}$$

The parameter $\theta \in [0, 1]$ is the importance or preference that the decision maker may attach to his choice. $\theta \in [0, 1/2]$ reflects that the decision maker has negative feeling or uncertainty; when $\theta \in [1/2, 1]$, the decision maker is quite positive and certain; for θ = 1/2 the decision maker remains neutral and he does not entertain any positive or negative feeling. Thus, the weighted value and weighted ambiguity reflects how the decision maker bestows his subjective approach on uncertainty of a quantity which lacks a proper definition such as a fuzzy data such as an intuitionistic fuzzy number.

Thus, the weighted value $V_{\theta}(\tilde{A})$ and weighted ambiguity $A_{\theta}(\tilde{A})$ of any triangular intuitionistic fuzzy number is defined as follows:

$$\begin{aligned} V_{\theta}(\tilde{A}) &= \frac{a_1 + 4a_2 + a_3 \left[\theta w_{\tilde{a}}^2 + (1 - \theta)(1 - u_{\tilde{a}})^2\right]}{6} \\ A_{\theta}(\tilde{A}) &= \frac{a_3 - a_1 \left[\theta w_{\tilde{a}} + (1 - \theta)(1 - u_a)^2\right]}{6} \end{aligned} \tag{5}$$

3 Working Procedure for Ranking [11]

The algorithm and process of the weighted value and ambiguity-based multi-attribute decision making with triangular intuitionistic fuzzy numbers can be summarised as follows:

Step 1 Identify and determine the attributes and alternatives, denote the sets of attributes and alternatives by $X = \{x_1, x_2, \ldots x_n\}$ and $O = \{o_1, o_2, o_3 \ldots o_m\}$, respectively.

Step 2 The decision maker's view is obtained to get ratings of the alternatives on the attributes, i.e. construct the triangular intuitionistic fuzzy number decision matrix $\tilde{A} = (a_{ij})_{mn}$.

Step 3 The weight vector $W = (w_1, w_2, w_3 \ldots w_n)^T$ is constructed taking into account the decision maker's judgement and attitude.

Step 4 The normalised TIFN decision matrix $D = (\tilde{d}_{ij})_{mn}$ of $\tilde{A}$ is constructed whose elements can be calculated through equation

$$\tilde{d}_{ij} = \left[\left(\frac{a_{1ij}}{a_{3i}^{+}}, \frac{a_{2ij}}{a_{3i}^{+}}, \frac{a_{3ij}}{a_{3i}^{+}}; w_{\tilde{\alpha}ij}, u_{\tilde{\alpha}ij} \right) \right] \quad (\mathrm{j} = 1, 2, 3 \ldots n,\ \mathrm{i} \in \psi_b)$$

and

$$\tilde{d}_{ij} = \left[\left(\frac{a_{1i}^{-}}{a_{1ij}}, \frac{a_{1i}^{-}}{a_{2ij}}, \frac{a_{1i}^{-}}{a_{3ij}}; w_{\tilde{\alpha}ij}, u_{\tilde{\alpha}ij} \right) \right] \quad (\mathrm{j} = 1, 2, 3 \ldots n,\ \mathrm{i} \in \psi_c)$$

where ψ_b and ψ_c denote, respectively, the sets of the inventory policies and policy attributes, $\psi_b \cap \psi_{\mathrm{c}} = \phi$ and $\psi_b \cup \psi_{\mathrm{c}} = \{1, 2, 3, \ldots, m\}$; $a_{3i}^{+} = \max \{a_{3ij} | j = 1, 2, 3, \ldots, n\}; i \in \psi_b, a_{1i}^{-} = \min\{a_{1ij} | j = 1, 2, 3, \ldots, n\}; i \in \psi_c$

Step 5 The weighted normalised triangular intuitionistic fuzzy number decision matrix $F = (\tilde{f}_{ij})_{mxn}$ of D is obtained. Its elements are calculated through the equation as follows: $\tilde{f}_{ij} = \omega_i \tilde{d}_{ij}\ (i = 1, 2, 3, \ldots, m; j = 1, 2, 3, \ldots, n)$.

Step 6 The aggregate values (for evaluations) $\tilde{T}_j$ of the alternatives x_j (j = 1, 2,3,…,n) are obtained as $\tilde{T}_j = \sum_{i=1}^{m} \tilde{f}_{ij}\ (j = 1, 2, 3, \ldots, n)$

Step 7 The non-increasing order of $\tilde{T}_j\ (j = 1, 2, 3, \ldots, n)$ is determined according to some ranking method of intuitionistic fuzzy numbers wherein they are ranked and the best alternative is obtained.

4 Numerical Example

An example is considered here wherein the four inventory policies IP1, IP2, IP3 and IP4 as per the Sect. 1 consider the attributes C1, C2, C3, C4, C5 and C6 as mentioned in Sect. 2. The various inventory policies are rated on the basis of the aforesaid criterions to obtain the decision matrix given below. Then the weight vector **W** = [0.18, 0.14, 0.12, 0.07, 0.14, 0.12] is obtained. Thereafter, the normalised triangular intuitionistic fuzzy number decision matrix is formulated and the comprehensive evaluations are done using the formula mentioned in the algorithm (Tables 2 and 3).

Table 2 The TIFN decision matrix

Inventory policies	Criterions					
	C1	C2	C3	C4	C5	C6
IP1	⟨(0.33, 0.55, 0.66); 0.7, 0.2⟩	⟨(0.71, 0.85, 1); 0.7, 0.2⟩	⟨(0.38, 0.5, 0.625); 0.6, 0.3⟩	⟨(0.16, 0.22, 0.33); 0.2, 0.3⟩	⟨(0.7, 0.7, 0.87); 6, 0.3⟩	⟨(0.49, 0.54, 0.62); 0.6, 0.3⟩
IP2	⟨(0.66, 0.07, 0.88); 0.6, 0.3⟩	⟨(0.28, 0.42, 0.57); 0.6, 0.3⟩	⟨(0.63, 0.75, 0.88); 0.5, 0.4⟩	⟨(0.7, 0.8, 1); 0.8, 0.2⟩	⟨(0.8, 0.8, 1); 0.8, 0.2⟩	⟨(0.87, 0.98, 1); 0.8, 0.2⟩
IP3	⟨(0.09, 0.88, 1); 0.8, 0.1⟩	⟨(0.29, 0.43, 0.57); 0.7, 0.2⟩	⟨(0.38, 0.5, 0.56); 0.7, 0.2⟩	⟨(0.72, 0.78, 0.88); 0.6, 0.3⟩	⟨(0.58, 0.67, 0.78); 0.6, 0.3⟩	⟨(0.51 ,0.54, 0.68); 0.5, 0.4⟩
IP4	⟨(0.22, 0.33, 0.55); 0.6, 0.4⟩	⟨(0.57, 0.71, 0.85); 0.5, 0.2⟩	⟨(0.75, 0.88, 1); 3, 0.3⟩	⟨(0.67, 0.78, 0.89); 0.5, 0.3⟩	⟨(0.48, 0.56, 0.69); 0.5, 0.4⟩	⟨(0.27, 0.33, 0.37); 0.6, 0.2⟩

Table 3 The weighted normalised TIFN decision matrix

Inventory policies	Criterions					
	C1	C2	C3	C4	C5	C6
IP1	⟨(0.059, 0.099, 0.119); 0.7, 0.2⟩	⟨(0.099, 0.119, 0.14); 0.7, 0.2)⟩	⟨(0.046, 0.66, 0.625); 0.6, 0.3⟩	(0.011, 0.0154, 0.023); 0.2, 0.3	(0.098, 0.098, 0.122); 0.6, 0.3	(0.059, 0.065, 0.074); 0.6, 0.3
IP2	⟨(0.119, 0.013, 0.158); 0.6, 0.3⟩	(0.039, 0.059, 0.080); 0.6, 0.3	(0.076, 0.09, 0.106); 0.5, 0.4	(0.049, 0.056, 0.07); 0.8, 0.2	(0.112, 0.112, 0.14); 0.8, 0.2	(0.104, 0.098, 0.12); 0.8, 0.2
IP3	⟨(0.016, 0.158, 0.18); 0.8, 0.1⟩	(0.041, 0.060, 0.080); 0.7, 0.2	(0.046, 0.06, 0.067); 0.7, 0.2	(0.050, 0.055, 0.062); 0.6, 0.3	(0.081, 0.094, 0.109); 0.6, 0.3	(0.032, 0.040, 0.044); 0.5, 0.4
IP4	⟨(0.040, 0.059, 0.099); 0.6, 0.4⟩	(0.080, 0.099, 0.119); 0.5, 0.2	(0.09, 0.106, 0.1);0.3, 0.3	(0.047, 0.055, 0.062); 0.5, 0.3	(0.067, 0.078, 0.097); 0.5, 0.4	(0.032, 0.040, 0.044); 0.6, 0.2

The weighted comprehensive values of inventory policies can be obtained as follows: $\tilde{T}_1 = \langle(0.372, 0.456, 0.8005); 0.2, 0.3\rangle$ $\tilde{T}_2 = \langle(0.499, 0.428, 0.674); 0.5, 0.4\rangle$ $\tilde{T}_3 = \langle(0.266, 0.467, 0.542); 0.3, 0.3\rangle$ $\tilde{T}_4 = \langle(0.356, 0.437, 0.541); 0.3, 0.4\rangle$

The values and ambiguities of the triangular intuitionistic fuzzy numbers for membership and non-membership functions can be calculated using Eqs. (1–4)

Table 4 Inventory policies ranked as per the criterion

Inventory policy	Rank
Economic order quantity	1
Just in time	3
Vendor-managed inventory	2
Monthly policy	4

$$
\begin{aligned}
V_\mu(\tilde{T}_1) &= \frac{0.372 + 4 * 0.456 + 0.8005[\theta(0.2)^2 + (1-\theta)(1-0.3)^2]}{6} \\
&= 0.50(0.2)^2 = 0.02 \\
V_\mu(\tilde{T}_2) &= 0.48(0.5)^2 = 0.12 \\
V_\mu(\tilde{T}_3) &= 0.446(0.3)^2 = 0.040 \\
V_\mu(\tilde{T}_4) &= 0.440(0.3)^2 = 0.040 \\
V_\nu(\tilde{T}_1) &= 0.50(0.8)^2 = 0.32 \\
V_\nu(\tilde{T}_2) &= 0.48(0.6)^2 = 0.173 \\
V_\nu(\tilde{T}_3) &= 0.446(0.7)^2 = 0.219 \\
V_\nu(\tilde{T}_4) &= 0.440(0.6)^2 = 0.1584
\end{aligned}
$$

For $\theta =$ 0.5, the decision maker remains neutral and, therefore, the value index and ambiguity index for the different alternatives are calculated at this point.

$$V_\theta(\tilde{T}_1) = 0.3575 \quad A_\theta(\tilde{T}_1) = 0.051 \; V_\theta(\tilde{T}_2) = 0.1992 \, A_\theta(\tilde{T}_2) = 0.01725 \; V_\theta(\tilde{T}_3) = 0.3082 \, A_\theta(\tilde{T}_3) = 0.03174 \; V_\theta(\tilde{T}_4) = 0.2178 \, A_\theta(\tilde{T}_4) = 0.015$$

The definition mentioned in Li [11] leads us to conclusion that (Table 4)

$$\mathrm{IP1} > \mathrm{IP3} > \mathrm{IP4} > \mathrm{IP2}$$

5 Conclusion

To stand out in the face of uncertainty, more and more companies begin to select the inventory policies as one of their development strategies. The selection of inventory policy plays a decisive role for achieving success in organisations thereby maintaining the correct balance of stock. This paper is an attempt to evaluate and rank the various inventory policies using the multi-attribute decision-making approach. Uncertainty and imprecision which characterise the real-life decision making are introduced through the concept of the intuitionistic fuzzy set (IFS), by

Atanassov [2]. The value and ambiguity index of IFN are applied to multi attribute decision-making inventory selection problems in which the ratings of alternatives on attributes are expressed with TIFNs. This model concludes that inventory selection can be treated as a multi-criteria problem and the method is quite effective, efficient, simple and modest. The model discussed herein can result in multiple preferences by taking new set of attributes to reach the final conclusion and has wide applicability in a number of sectors where the optimal selection of inventory policy is vital for its progress and development and can provide an edge over its competitors. Therefore, to meet the challenges of the dynamic business world which unfolds into newer horizon, the model proposed in this paper could give a newer dimension to the managers.

References

1. Arcelus, F.J., Srinivasan, G.: Discount strategies for one-time-only sales. IIE Trans. **27**(5), 625–633 (1995)
2. Atanassov, K.T.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**, 87–96 (1986)
3. Atanassov, K.T.: More on intuitionistic fuzzy sets. Fuzzy Sets Syst. **33**(1), 37–45 (1989)
4. Brooking, S.A.: Inventory system costs: source data for analysis. Eng. Costs Prod. Econ. **13** (1), 1–12 (1987)
5. Cetinkaya, S., Parlar, M.: Optimal myopic policy for a stochastic inventory problem with fixed and proportional backorder costs. Eur. J. Oper. Res. **110**(1), 20–41 (1998)
6. Chen, S.J., Hwang, C.L.: Fuzzy multiple attribute decision making methods, pp. 289–486. Springer, Berlin (1992)
7. Dohi, T., Okamura, H., Osaki, S.: Optimal control of preventive maintenance schedule and safety stocks in an unreliable manufacturing environment. Int. J. Prod. Econ. **74**(1), 147–155 (2001)
8. Gupta, A., Garg, R.K., Tewari, P.C.: Multi-criteria ranking of inventory ordering policies using fuzzy based-distance based approach for indian automotive industry. i-Manager's J. Manage. **8**(1), 411994 (2013)
9. Li, D.F.: Decision and game theory in management with intuitionistic fuzzy sets. In: Studies in Fuzziness and Soft Computing (2005)
10. Li, D.F.: A ratio ranking method of triangular intuitionistic fuzzy numbers and its application to MADM problems. Comput. Math Appl. **60**(6), 1557–1570 (2010)
11. Li, D.F., Nan, J.X., Zhang, M.J.: A ranking method of triangular intuitionistic fuzzy numbers and application to decision making. Int. J. Comput. Intell. Syst. **3**(5), 522–530 (2010)
12. Mitchell, H.B.: Ranking type-2 fuzzy numbers. IEEE Trans. Fuzzy Syst. **14**(2), 287–294 (2006)
13. Nayagam, G., Lakshmana, V., Venkateshwari, G., Sivaraman, G.: Ranking of intuitionistic fuzzy numbers. In IEEE International Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), pp. 1971–1974. IEEE (2008)
14. Petrovic, R., Petrovic, D.: Multicriteria ranking of inventory replenishment policies in the presence of uncertainty in customer demand. Int. J. Prod. Econ. **71**(1), 439–446 (2001)
15. Prasad, S.: Classification of inventory models and systems. Int. J. Prod. Econ. **34**(2), 209–222 (1994)
16. Wang, Q.J.: Survey on fuzzy multi-criteria decision-making approach. Control Decis. **23**, 601–606 (2008)

17. Wan, S.P.: Survey on intuitionistic fuzzy multi-attribute decision making approach. Control Decis. **25**(11), 1601–1606 (2010)
18. Xu, Z.: Intuitionistic fuzzy multiattribute decision making: an interactive method. IEEE Trans. Fuzzy Syst. **20**(3), 514–525 (2012)
19. Zadeh, A.L.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
20. Zhang, M., Nan, J.: A compromise ratio ranking method of triangular intuitionistic fuzzy numbers and its application to MADM problems. Iran. J. Fuzzy Syst. **10**(6), 21–37 (2013)
21. Zhou, B., Zhao, Y., Katehakis, M.N.: Effective control policies for stochastic inventory systems with a minimum order quantity and linear costs. Int. J. Prod. Econ. **106**(2), 523–531 (2007)
22. Zhang, X., Xu, Z.: A new method for ranking intuitionistic fuzzy values and its application in multi-attribute decision making. Fuzzy Optim. Decis. Mak. **11**(2), 135–146 (2012)

Part II
Artificial Neural Network

An NSGA II-Based Approach for Optimization of Reconfigurable Cellular Manufacturing System

Rajeev Kant, Vijay Pandey and L.N. Pattanaik

Abstract In this paper, the application of Non-dominated Sorting Genetic Algorithm (NSGA) II for optimization of reconfigurable cellular manufacturing system is presented. To sustain in a dynamic manufacturing scenario, a manufacturing system must be flexible enough to cope with demand fluctuation competing on both cost and time dimensions. In the present study, the drawbacks of conventional cellular manufacturing due to dedicated machines have been attempted to overcome using Reconfigurable Machine Tools (RMTs). The characteristics of RMTs are implemented to impart a certain degree of flexibility in a cellular manufacturing framework. An RMT encompasses a specific and customized range of operational requirements in terms of capacity and functionality. Multi-dimensional objectives and parameters are established to efficiently incorporate RMTs into machine cells and mathematical model is developed for the same. Then, an NSGA II-based approach is proposed to optimize the capacities and functionalities of RMTs pursuing the objective functions.

Keywords NSGA II · Cellular manufacturing · RMTs · Multi-objective optimization

1 Introduction

A classical GA yields good result with single objective function. But in Cellular Manufacturing System (CMS) design problem, where multiple criteria are taken into account, multiple objective functions need to be optimized simultaneously. All the objective functions are combined into one fitness function. However, in some cases of multi-objective problems, there does not exist a solution that is best with respect to all objective functions because of conflicting objectives. The objective functions cannot be conveniently incorporated into single fitness function because

R. Kant (✉) · V. Pandey · L.N. Pattanaik
Department of Production Engineering, Birla Institute of Technology,
Mesra, Ranchi 835215, India
e-mail: kant.rajeev@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_6

of obscure relationship between them. To provide a better solution for such problems, Srinivas and Deb [1] proposed first NSGA and further improved version of NSGA II by Deb et al. [2]. The application of NSGA II has been successfully demonstrated by various researchers to develop CMS model encompassing multi-directional objectives [3–5].

Cellular manufacturing (CM) has been considered the most promising strategy for batch production [6, 7]. In recent years, the justification for CM has been a subject of discrepancy due to lack of flexibility [8]. Many authors have advocated the inclusion of RMTs in CMS to achieve the highest degree of flexibility in CMS and to strengthen the applicability of CM in rather complex market situations [9–11]. An RMT is a modular machine designed for a specific and customized range of operational requirements, and can be economically converted to meet new requirements [12]. It comprises of basic modules and a library of auxiliary modules. These auxiliary modules can be added or removed according to the capacity and/or functionality required.

In this paper, the authors have presented an NSGA II-based novel methodology to incorporate RMTs with CMS. A CMS framework from the available literature is illustrated in the next section. In Sect. 3, various parameters and objective functions are defined to efficiently incorporate RMTs into machine cells. In later section, an NSGA II-based approach is developed to achieve an optimized solution for reconfigurable machine cell design.

2 Problem Formulation

The present work illustrates a discrete manufacturing model (Fig. 1) from Kant et al. [13] where a variety of products are manufactured on an assembly line. The part families are identified out of the sub-assemblies of these products and

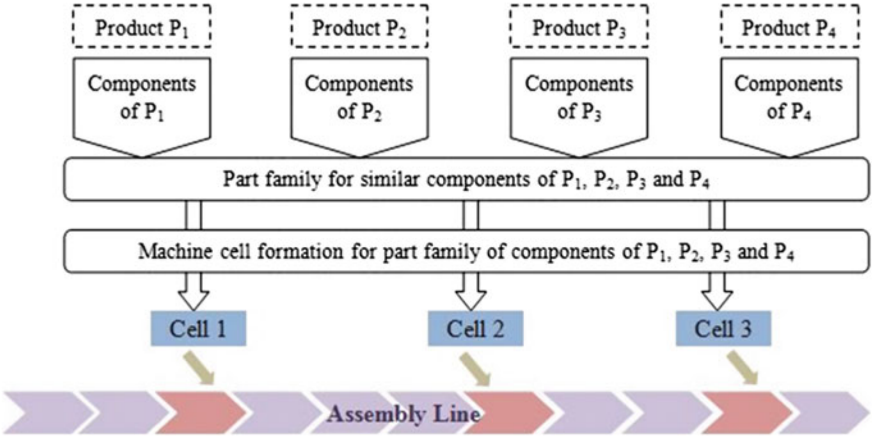


Fig. 1 Discrete manufacturing model comprising a variety of products

subsequently these part families are assigned to a cell. These cells produce the components corresponding to the products being manufactured and deliver the components to the assembly line at suitable locations.

The model presented here imitates a more realistic situation with a variety of products and varying levels of demand and the capacities and functionalities of RMTs can be varied accordingly.

3 Parameter Selection

The ultimate goal is to achieve the least reconfiguration effort and operation time, hence three key parameters were selected: (i) selection of RMTs, (ii) assignment of RMTs into machine cells, and (iii) the sequence in which products are to be manufactured. All the three parameters are mutually dependent and cannot be optimized independently. Three objective functions were established for optimization of each parameter:

(i) intercellular movements,
(ii) cell load imbalance, and
(iii) reconfiguration cost of the cellular system

The objective function (i) conflicts with (ii) as minimizing intercellular movement may impede the cell load balancing. Also objective (ii) and (iii) are conflicting since cell load balancing may incur high reconfiguration cost. These objective functions are abided by a number of suitable assumptions and constraints applicable for the cellular manufacturing model.

4 Implementation of NSGA II

The above problem consists of multi-dimensional objectives and sharing parameters, NSGA II is suggested to have better convergence than other multi-objective evolutionary algorithms for such problems [2]. Figure 2 shows the framework for proposed multi-objective genetic algorithm (NSGA II).

The simple GA procedure starts with a set of probable solutions, called population, and performs search for the best solution navigating through the natural evolution of new solutions (offsprings). It follows the Darwinian theory of the survival of the fittest making each new generation having better quality and fitness than the older one. The algorithm continues until a solution with best fitness value is obtained.

In NSGA II, the population is sorted according to an ascending level of non-domination. Each solution is compared with every other solution in the population and non-dominated sets are created until all solutions are classified into a non-domination level. Level 1 is the best level, 2 is the next best level, and so on. A solution $x(1)$ is said to dominate other solution $x(2)$, if

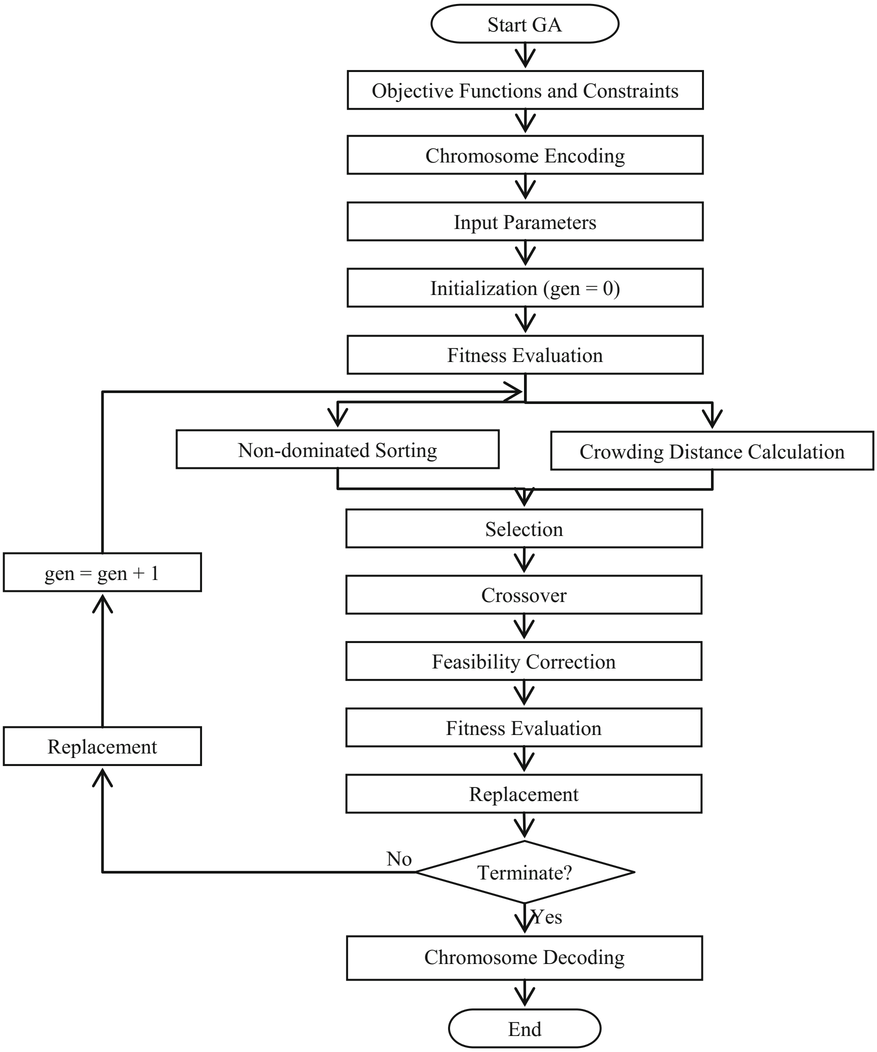


Fig. 2 Framework for the proposed NSGA II

(a) the fitness value of $x(1)$ is no worse than that of $x(2)$ for all objectives, and
(b) solution $x(1)$ possesses strictly better fitness than $x(2)$ for at least one objective.

In addition to the fitness value, another parameter ‘crowding distance’ is used to evaluate the solutions. The crowding distance is a measure of how close is an individual to its neighbours in that population. Large crowding distance will result in better diversity and larger search area.

4.1 Objective Functions

Assumptions. The following assumptions were made while formulating objective functions:

- Each component has a fixed process routing.
- Each RMT has its own basic and auxiliary modules.
- RMTs and their modules are sufficiently available for all operational requirements.
- The capacities of the RMTs are known and perform within a specific range of cycle times.
- The machining time includes the setup time.
- Reconfiguration time includes module change times.

Input Parameters.

P	Product type, indexed by p;
V_p	Actual demand of product type p;
C_q^p	Component q of product type p, indexed by q;
N_q^p	Number of component q required for product type p;
M_m	RMT type, indexed by m;
A	Auxiliary module, indexed by a;
O	Operation type, indexed by o;
T_o^a	Processing time for operation o on module a;
G_a	Maximum capacity of module a;
RT_a	Reconfiguration time for module a;
RC_a	Cost for reconfiguring module a;
C	Cell, indexed by c.

Decision Variables.

$$X_{q,p}^o = \begin{cases} 1 \text{ if operation } o \text{ is required for component } C_q^p \\ 0 \text{ otherwise} \end{cases}$$

$$Y_a^o = \begin{cases} 1 \text{ if operation } o \text{ is performed on auxiliary module } a \\ 0 \text{ otherwise} \end{cases}$$

$$Z_m^a = \begin{cases} 1 \text{ if auxiliary module } a \text{ belongs to machine } m \\ 0 \text{ otherwise} \end{cases}$$

$$U_c^m = \begin{cases} 1 \text{ if machine } m \text{ is assigned to cell } c \\ 0 \text{ otherwise} \end{cases}$$

$$V_{q,p}^c = \begin{cases} 1 \text{ if component } q \text{ of product type } p \text{ is assigned to cell } c \\ 0 \text{ otherwise} \end{cases}$$

$$W_c^a = \begin{cases} 1 \text{ if module } a \text{ is already configured in cell } c \\ 0 \text{ otherwise} \end{cases}$$

Total number of intercellular movements during the production cycle of all product types is

$$S = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{o=1}^{O} \sum_{a=1}^{A} \sum_{m=1}^{M} \sum_{c=1}^{C} X^{o}_{q,p} Y^{o}_{a} Z^{a}_{m} \left(1 - U^{m}_{c}\right). \tag{1}$$

Machining time for component q of product type p in cell c is given by

$$T^{c}_{p,q} = V_p N^{p}_{q} \sum_{0=1}^{O} \sum_{m=1}^{M} \frac{X^{o}_{q,p} Y^{o}_{a} Z^{a}_{m}}{U^{m}_{c}} T^{a}_{o}. \tag{2}$$

Average reconfiguration time of cell c in which component q of product type p is assigned for production is

$$R^{c}_{p,q} = \frac{1}{V_p} \left[\max \left\{ \left(V^{c}_{q,p} X^{o}_{q,p} Y^{o}_{a} \left(1 - W^{a}_{c}\right) RT_a \right) | \forall\, o, a \right\} \right]. \tag{3}$$

Hence, total time elapsed in processing of component q of product type p in cell c is obtained by

$$F^{c}_{p,q} = T^{c}_{p,q} + R^{c}_{p,q}. \tag{4}$$

Total cell load imbalance is calculated as the sum of squares of deviations from mean of flow time among cells for each product type

$$D = \sum_{p=1}^{P} \sum_{c=1}^{C} \left[\sum_{c=1}^{C} F^{c}_{p,q} - \frac{1}{C} \sum_{c=1}^{C} \sum_{q=1}^{Q} F^{c}_{p,q} \right]^{2}. \tag{5}$$

The reconfiguration cost of the cellular system for change over product p to product p’ is calculated as

$$RC^{p'}_{p} = \sum_{q'=1}^{Q'} \sum_{q=1}^{Q} \sum_{c=1}^{C} \sum_{a'=1}^{A'} \sum_{a=1}^{A} \sum_{o'=1}^{O'} \sum_{o=1}^{O} V^{c}_{q,p} V^{c}_{q',p'} X^{o}_{q,p} Y^{o}_{a} X^{o'}_{q',p'} Y^{o'}_{a'} \left(1 - W^{a'}_{c}\right) RC_{a'}. \tag{6}$$

Total reconfiguration cost of the cellular system throughout the production is given by

$$RC = \sum_{p=1}^{P} \sum_{p'=1}^{P'} RC^{p'}_{p}. \tag{7}$$

Equations (1), (5), and (7) represent the three objective functions intercellular movements, cell load imbalance, and the reconfiguration cost of the cellular system, respectively.

The following equations describe the various constraints applicable to our cellular model. Equation 8 ensures a machine to be assigned to only one cell at a time. Equation 9 guarantees that one component cannot be assigned to more than one cell. Equation 10 ensures that each cell has components from each product type. Equation 11 restricts the operation time of a module within its specified capacity.

$$\sum_{c=1}^{C} U_c^m = 1. \quad \forall M_m \tag{8}$$

$$\sum_{c=1}^{C} V_{q,p}^c \leq 1. \quad \forall p, q \tag{9}$$

$$\sum_{q=1}^{Q} V_{q,p}^c = 1. \quad \forall c, p \tag{10}$$

$$T_o^a \in G_a. \quad \forall o, a \tag{11}$$

4.2 Chromosome Encoding

The chromosome is composed of three building blocks representing product types, components of product types to be assigned to a cell, and machines, respectively. The length of the first block is equal to the number of product types under consideration. The second block is further divided into sub-strings that represent cells and each of them has components of each product types. Hence, the length of the second block is equal to the number of the cells multiplied by number of product types. The length of the third block is equal to the total number of machines employed. The chromosome structure is shown in Fig. 3.

The encoding scheme used here is permutation encoding. The value of each gene is a natural number and passes different information for each block. The number associated with genes in the first block states the production sequence of product types. In the second block, components are arranged in cell sub-strings by the order

$$\text{Chromosome} = [\underbrace{\{\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots, \mathrm{P}_\mathrm{P}\}}_{\text{Product type}} \underbrace{\{\mathrm{C}_1, \mathrm{C}_2, \mathrm{C}_3, \ldots, \mathrm{C}_\mathrm{C}\}}_{\text{Cell}} \underbrace{\{\mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_3, \ldots, \mathrm{M}_\mathrm{M}\}}_{\text{Machine}}]$$

$$\mathrm{C}_1 = \underbrace{(C_1^1, C_1^2, C_1^3, \ldots, C_q^p)}_{\text{Components from each product type}}$$

Fig. 3 Chromosome structure

of product types and the number correspondence to a gene in a cell sub-string indicates the component number belonging to that cell. In third block, the number associated with a gene shows that the machine is assigned to that cell number.

4.3 GA Operators

The functions of GA operators are to perform the evolution process through which the GA propagates searching for the best solution. The evolution process, in essence, involves three steps: selection of parents, reproduction of new off springs, and replacement of old chromosomes with the new ones. The following GA operators are used for our problem:

Selection. In NSGA II, tournament selection is followed for selection of parents based on crowding distance and non-domination rank. An individual with lower (better) rank is selected; however, if two individuals are placed on the same non-dominance level, the individual located farther from the crowd is preferred.

Reproduction. For the present problem, cell two-point uniform crossover operator is used in the proposed so that each block performs crossover in their own region thus maintaining the integrity of building blocks within the chromosome structure. Figure 4 illustrates the application of uniform crossover to an example chromosome. As can be seen in the figure, the crossover operator duplicates some genes and loses some necessary genes in first two blocks. In offspring 1, same sequence number is assigned to product type 2 and 3 and in offspring 2, product type 1 and 3 acquire the same sequence number. In addition, component number 3 and 2 of product type 1 gets duplicated in offspring 1 and 2, respectively. Offspring 1 and 2 violate the specified constraints and make them infeasible solution. Hence, a repair algorithm is designed to repair a chromosome putting it back into the search space.

In the present case, it can be easily noticed that mutation will have reverse impact on the chromosome as swapping will disturb the integrity of the chromosome.

Replacement. After the offspring are created, probabilistic crowding approach [14] is used for forming the new population where the parents (p) and offspring (o) compete in a probabilistic tournament established by the following rule

Fig. 4 Crossover illustration

↓ ↓ ↓

Parent 1 = (13**2** | 111 **2**32 324 443 | 12**4**131324)

Parent 2 = (21**3** | 121 **3**32 213 444 | 24**2**111334)

⇩

Offspring 1 = (13**3** | 111 **3**32 324 443 | 12**2**131324)

Offspring 2 = (21**2** | 121 **2**32 213 444 | 24**4**111334)

(Eq. 13), and the winner gets a place in the new population. In replacement policy, the individuals with better fitness will be retained in the new population, hence maintaining *elitism* throughout the GA procedure.

$$P_O = \frac{f(o)}{f(o) + f(p)}. \tag{13}$$

where P_O is the probability of winning o and f is the fitness function.

5 Conclusion

This study aims to demonstrate the application of NSGA II for multi-objective reconfigurable machine cell formation. In the present study, a combinatorial problem of assigning RMTs to machine cells is attempted. The capacities and functionalities of RMTs were addressed through three sharing parameters and optimized through three conflicting objective functions.

The proposed approach is applicable to multi-product discrete manufacturing problem, the kind of which does not exist in the literature. Although NSGA II has been proved more efficient than other GAs for such complex CMS design problem, it still needs to test on different sets of problems.

References

1. Srinivas, N., Dev, K.: multiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. **2**(3), 221–248 (1994)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA–II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
3. Neto, A.R.P., Filho, E.V.G.: A simulation based evolutionary multiobjective approach to manufacturing cell formation. Comput. Ind. Eng. **59**, 64–74 (2010)
4. Lian, J., Liu, -C.G., Li, W.J., Evans, S., Yin, Y.: Formation of independent manufacturing cells with the consideration of multiple identical machines. Int. J. Prod. Res. **52**(5), 1363–1400 (2014)
5. Aghajani, A., Didehbani, S.A., Zadahmad, M., Seyedrezaei, M.H., Mohsenian, O.: A multi-objective mathematical model for cellular manufacturing systems design with probabilistic demand and machine reliability analysis. Int. J. Adv. Manuf. Technol. **75**, 755–770 (2014)
6. Wemmerlov, U., Hyer, N.L.: Cellular manufacturing in the us industry: a survey of users. Int. J. Prod. Res. **27**(9), 1511–1530 (1989)
7. Wemmerlov, U., Johnson, D.J.: Cellular manufacturing at 46 user plants: implementation experiences and performance improvements. Int. J. Prod. Res. **35**(1), 29–49 (1997)
8. Johnson, D.J., Wemmerlov, U.: Why does cell implementation stop? factors influencing cell penetration in manufacturing plants. Prod. Oper. Manag. **13**(3), 272–289 (2004)
9. Pattanaik, L.N., Kumar, V.: Multiple levels of reconfiguration for robust cells formed using modular machines. Int. J. Ind. Syst. Eng. **5**(4), 424–441 (2010)

10. Renna, P., Ambrico, M.: Design and reconfiguration models for dynamic cellular manufacturing to handle market changes. Int. J. Comput. Integ. Manuf. **28**(2), 170–186 (2015)
11. Eguia, I., Racero, J., Guerrero, F., Lozano, S.: Cell formation and scheduling of part families for reconfigurable cellular manufacturing systems using tabu search. Simul. Trans. Soc. Modell. Simul. Int. **89**(9), 1056–1072 (2013)
12. Landers, R., Min, B.K., Koren, Y.: Reconfigurable manufacturing tools. Ann. CIRP **49**, 269–274 (2001)
13. Kant, R., Pattanaik, L.N., Pandey, V.: Framework for strategic implementation of cellular manufacturing in lean production environment. J. Manuf. Technol. Res. **6**(3/4), 177–191 (2015)
14. Mengshoel, O.J., Goldberg, D.E.: Probabilistic crowding: deterministic crowding with probabilistic replacement. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), pp. 409–416, Orlando FL (1999)

Advance Prediction of Adverse Digressions in Continuous-Time Systems Using ANN Kernels: A Generic Approach Instantiated in Steel Manufacturing

Arya K. Bhattacharya and K. Rajasekar

Abstract A domain-independent generic methodology is developed for online prediction of rapid adverse digressions in continuous-time systems, preceding the actual incipience of such digressions. The complete methodology itself consists of three stages. The first two stages are domain specific and involve statistical analysis and standard prediction tools like Artificial Neural Networks. The third stage that transforms the ANN outputs into a reliable measure of the instantaneous digression of the system is generic across domains and is the contribution of this work. The core novelty enabling this transformation is a paradigm shift in assessment of the ANN output, from that of “classification” to that of “continuity”. The development of this methodology is performed on a specific industrial process—continuous casting in steel manufacturing—and described in this paper. This can be applied with minor customization to continuous-time systems in domains such as the biological, industrial processes, vehicles, and economic and financial systems.

Keywords Continuous-time systems • Artificial neural networks • Classification paradigm • Continuity paradigm • Adverse digressions • Combinatorial relationships

1 Introduction

Continuous-time systems are natural not only to the process industry but to diverse domains such as the biological, geophysical, economic and many others. Almost any dynamical system is inherently continuous time though for purposes of analysis these are most often discretized. Here, we do not try to analyze a continuous system per se but seek to identify occurrences of a very specific feature of most of them—rapid

A.K. Bhattacharya (✉)
Mahindra Ecole Centrale, Hyderabad, India
e-mail: aryabhat9@gmail.com

K. Rajasekar
Tata Steel, Jamshedpur, India

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_7

digressions into regimes of adverse performance. This ‘adversity’ of performance is inherent to the system specifics—for an economy, for example, it represents a state of slowdown, for a moving vehicle, undesirable response to driver commands, and for an industrial process, a breakdown.

Importantly, we seek to identify these rapid digressions into adverse regimes *before they have initiated*, so that corrective action can be taken in time for prevention. The immediate implication is that the coming digression is sought to be captured at a point in time when the state-space representation of system dynamics is yet to sense this, i.e., feedback control laws cannot be applied.

Most often the state of a dynamical system is characterized by a large number of parameters. Prior to the initiation of a digression of a specific type, a number of these parameters would have acquired a certain combinatorial relationship, and in principle one can identify a one-to-one mapping between the acquisition of this relationship and the consequent adverse digression. Among the many (possibly hundreds of) parameters characterizing a dynamical system, it is a non-trivial task to identify those that combine to trigger a specific adverse digression. Equally or more challenging is to identify the characteristics of this combination itself. If these two steps can be successfully accomplished, then an impending adversity can be forecast with high reliability. In this work, we have developed a novel mechanism for predicting rapid digressions in dynamical systems based on the above lines. This is applied successfully for predicting breakdowns in a steel continuous casting process.

The steel manufacturing process is composed of a number of linked but significantly distinct sub-processes. One of the most important of these is the continuous casting (sub)-process [1, 2], which marks the transition from the liquid form of steel to its solid form. This process essentially converts molten steel coming in ladles from upstream production units into a continuous completely solidified rectangular section ‘strand’ that is cut into discrete slabs for rolling into sheets in downstream mills. Figure 1 provides an overview of the process. The primary cooling mold is of rectangular section and about a meter in length; sectional dimensions approximately 1.5 × 0.1 m—the shape finally acquired by the strand. The inner walls of the mold are made of copper plates to facilitate heat transfer from the moving steel strand to cooling water flowing within a network of pipes embedded in these plates. Upon

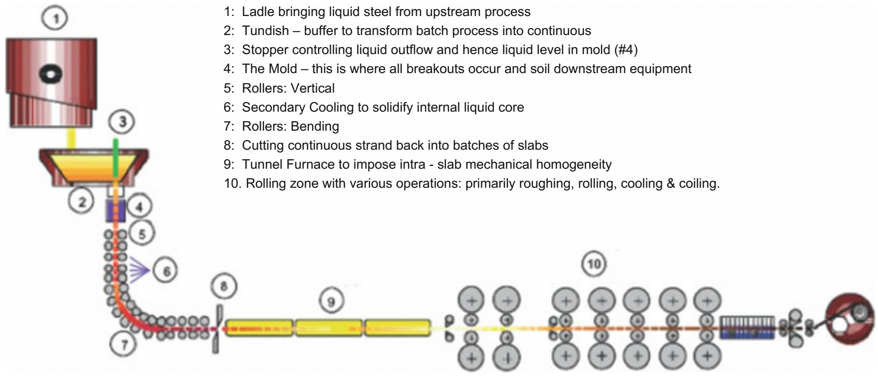

1: Ladle bringing liquid steel from upstream process
2: Tundish – buffer to transform batch process into continuous
3: Stopper controlling liquid outflow and hence liquid level in mold (#4)
4: The Mold – this is where all breakouts occur and soil downstream equipment
5: Rollers: Vertical
6: Secondary Cooling to solidify internal liquid core
7: Rollers: Bending
8: Cutting continuous strand back into batches of slabs
9: Tunnel Furnace to impose intra - slab mechanical homogeneity
10. Rolling zone with various operations: primarily roughing, rolling, cooling & coiling.
1
3
2
4
5
6
7
8
9
10

Fig. 1 Schematic of complete thin slab casting process

heat loss, a solidified shell forms around the perimeter of the downwards flowing steel, thickening gradually till the strand emerges from mold bottom with liquid still encapsulated inside. This is further cooled to complete solidification in the secondary cooling stage using spray cooling [2] (Fig. 2).

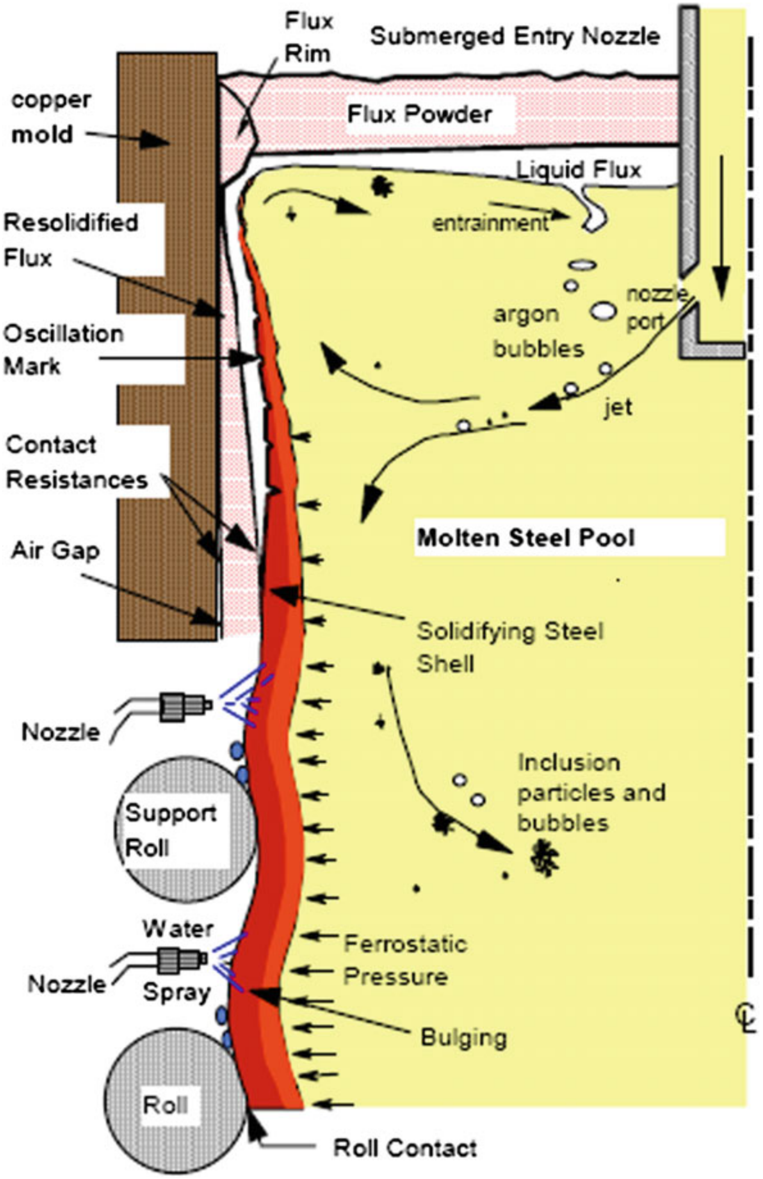


Fig. 2 Illustration of flow, cooling and solidification in mold [2]. Solid shell formation in mold, and emergence of strand from mold in secondary (spray) cooling phase

The continuous casting process is possibly the most critical among the steel manufacturing sub-processes from the viewpoints of productivity, stability and quality; hence, disruptions need to be avoided. The most typical of these disruptions is in the form of “breakouts”—where the formative steel shell locally sticks to the mold wall and tears (“stickers”), or cracks due to stresses (“cracks”), or simply ruptures at some location where it is weakly formed and hence unable to sustain internal liquid pressure (“thin shells”). In any of these cases, the internal liquid “breaks out” and then freezes on downstream machinery disrupting the casting process for a few hours. Thus, it is important to have mechanisms for detecting and preventing breakouts.

Among the different types of breakouts, “stickers” make up the majority and there exist mature Breakout Prevention Systems or BPS [3, 4] that use arrays of thermocouple sensors embedded in the mold copper plate, along with pattern recognition algorithms, to recognize the sticker after its occurrence and then command corrective actions to the caster control system to prevent the breakout. However, “cracks” make up a significant minority and provide no clear sensory signals, and hence cannot be prevented from resulting in breakout once the crack fault has crossed the stage of incipience. To prevent this class of breakouts, advance prediction mechanisms for detecting rapid digressions in the process, on the lines discussed in the earlier part of this section, need to be developed.

The rest of this paper explains the approach to advance prediction, starting with a classical approach using neural networks, and on to novel techniques that exploit the very nature of adverse digressions of continuous-time systems. Finally results are presented, and extensions to generic continuous systems are discussed.

2 The Basic Classification Approach

According to the above discussion, ‘crack’-induced breakouts are sought to be prevented by first recognizing the formation of the corresponding specific relational pattern within the hundreds of parameters that characterize continuous casting. It is the attainment of such a pattern that engenders the incipience of a crack. Furthermore, it is highly unlikely that all of the process parameters will participate in the formation of this pattern; most will be neutral while a specific few will combine to create this relationship. The crucial sub-steps are, then, to first identify these specific few variables, and next discover the combinatorial relationship between them that generates these faults.

The first task, then, is to reduce the hundreds of online variables to the specific few that matter. A two-step approach is adopted to achieve this. First, application of domain knowledge extracted from specialists in the specific field enables narrowing down the number of online variables relevant to crack formation from about three hundred to just around sixty. Second, performing a sequence of steps in statistical analysis using available data facilitates the reduction of the number of variables from about sixty to just six.

The next task is to suitably train an artificial neural network (ANN) with data gathered over a sufficiently large time window of operation of the continuous casting process in the caster of interest. This happens to be in a European plant of a major private sector steel manufacturer. Data are recorded consistently at 1-s intervals during operation of the caster. Each training data record of the ANN, i.e., row of the training data set, is composed of six input variables and one output variable, extracted from this recorded database. While the input data consist of the six variables identified by the above method, the output data are set at either ‘0’ or ‘1’ depending upon whether the record is taken from a “good” or “bad” casting period, respectively. A “bad” casting period is defined to lie between two and ten minutes preceding a breakout, while “good” data are taken from a period that precedes a breakout by at least three hours, and preferably from ‘sequences’ that do not terminate in breakouts. A casting ‘sequence’ normally lasts around ten to eleven hours.

The considered ANN is a multilayer feedforward network with two hidden layers, and is trained using two alternate techniques—by the steepest gradient backpropagation method, and by a Differential Evolution optimization approach. However, discussion on details of the ANN is tangential to the theme of this paper and hence is not described further here.

The ANN is thus trained to distinguish between normal casting conditions, and impending-breakout situations, based on the output value. It may be expected, somewhat simplistically, that when the ANN is fed with test data, it will generate a ‘0’ for a good casting period, and a ‘1’ for a breakout-impending case. However, an ANN cannot be perfectly accurate, and errors of three different origins will naturally creep into ANN computations. First, noise in the training data set. It has to be kept in mind that we are dealing with measured industrial data of a process that is defined by hundreds of variables. Second, numerical errors of the ANN process. If a perfect mathematical function were given to be learnt by the ANN, errors will never reduce to exact zero. Finally, noise of the test data. Because the test data are also of industrial origin, each data record will have its variations. Under these conditions the trained ANN is not expected to generate a ‘0’ or a ‘1’, instead, it may be realistically desired that for good cases, the output should be below 0.5, and for bad cases, it should lie above 0.5. Thus, the value 0.5 is the classification ‘watershed’, from the “classification perspective” of using the ANN to decide between the two ‘classes’ of normal and breakout-impending conditions.

Let us see how the ANN performs when it functions in the above manner, which we now call as the “classification perspective”. The ANN is composed of two hidden layers with 16 neurons each, along with an input layer with 6 input neurons and a single-neuron output layer. The data used for training have 434 data sets, out of which 143 are breakout cases, and 291 ‘good’ cases. At end of training, the accuracy of “classification” on above lines is analyzed. It is seen that 6.3 % of breakout cases and 2.1 % of good cases are classified wrongly. Figure 3 shows a plot of the difference between the desired value (0 or 1) and the corresponding ANN output. If a good case with desired value ‘0’ is misclassified as a breakout case with an ANN output value greater than 0.5, it implies that the above difference will be

Fig. 3 Divergence between target and ANN outputs; outside ±0.5 is misclassification; shows divergence on test data

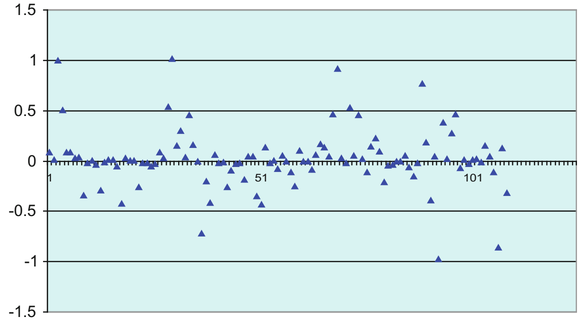
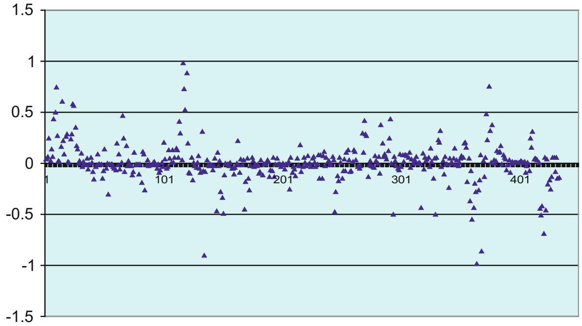


Fig. 4 Divergence between target and ANN outputs; outside ±0.5 is misclassification; shows divergence on training data



less than −0.5. Conversely, if the difference is greater than +0.5, it implies that an impending breakout (desired value ‘1’) is predicted as a good case with ANN output less than 0.5. All other cases (i.e., difference in range −0.5 to +0.5) are correctly classified. In addition, the average of all breakout cases comes to 0.863, and of all good cases is 0.049, which gives a separation of 0.814.

The trained ANN is then used on test data. Total of 109 data points are taken, of which 39 are impending-breakout cases and 70 are good cases. 18 % out of these 39 and 4.3 % out of 70 are classified wrongly. Figure 4 plots the difference between desired and actual outputs on lines analogous to Fig. 3. The average of all breakout cases comes to 0.774 and of all good cases at 0.087, giving a separation of 0.686. As expected, one can see a significant deterioration in classification accuracy as one moves from training data to test data.

There is a third element of testing of the ANN that is most crucial but cannot be performed a priori, namely on the real-time industrial data. These are the data on which the ANN is supposed to work when it is put in control mode, months or possibly years after it has been trained on data extracted from a certain operational window in time. Unlike any other process types, industrial processes tend to drift, sometimes severely, in timescales of weeks and months. The performance of the ANN on real industrial data—we may call these as ‘application data’—will degrade further compared to that on test data. This is shown illustratively in Fig. 5a–c by

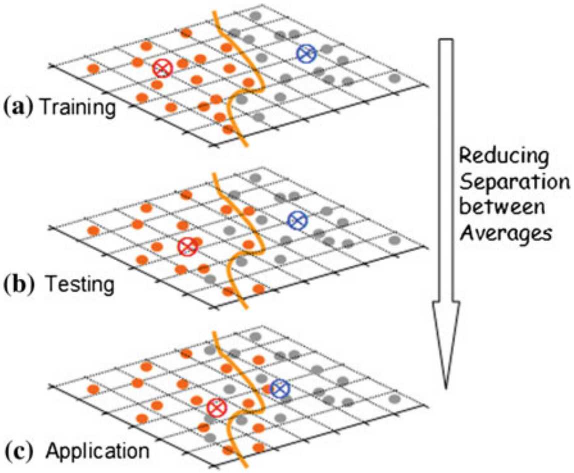


Fig. 5 A 2-D nonlinear classification problem; orange curve is the dividing surface. Notice how classification accuracy reduces as one moves from training to application data, and the separation between averages reduce

allusion to a simple classification problem in two-dimensional space. The red and gray balls (each ball representing an instance) are classified across the orange curve which is the dividing line in a nonlinear problem—Fig. 5a shows the classification result on training data, 5b on test data and Fig. 5c on application data. As one moves from Fig. 5a–c, the classification deteriorates in the sense more and more of the balls are misclassified, and the separation between the respective centroids (averages) on either side keeps reducing. Figure 5a and b alludes to above actual industrial observations.

Now let us evaluate the impact of these inaccuracies on the objective of preventing adverse digressions of a continuous process. The ANN operating cycle is 1 s, implying that every 1 s it acquires new set of field inputs and generates an output that is either “process-is-normal” or “impending breakout”. When a breakout alarm is raised, certain corrective control steps on the process have to be manually and immediately initiated. False alarms at a rate of 4 % (as seen in test data) imply that on an average, once in every 25 s, an alarm is raised and corrective action has to be taken, when nothing is wrong in the process! Let us assume for a moment that ANN prediction accuracy could somehow be improved to a (hypothetical) level of only 0.1 %—possibly the limit on accuracy that could be achieved on a real process. Even this would imply that a false alarm is raised once every 1000 s, i.e., ~3 times/hour or 70 times per day. If a crack breakout occurs on average once every 30 days, this would imply that to prevent one breakout, undesirable control action has to be taken 2000 (i.e., ~30 × 70) times! This will be unacceptable to any operations engineer.

The implication of the above is that using an ANN in the traditional “classification perspective”, the objective of achieving advance recognition, and consequent prevention, of adverse digressions in the continuous process cannot be achieved in practice. Hence, alternate mechanisms need to be developed.

3 Shifting to the Continuity Paradigm

Let us for a moment define a hypothetical variable called the normality, or rather the “abnormality” of casting. This is supposed to define at any instant how good, or how bad, the “state of casting” is. This variable is conceptual in nature because no such real physical parameter (unlike say temperature or velocity) exists in real and it cannot be measured by any kind of instrument. But it can be of value as a reference indicator of the instantaneous health of the process. Further, let this variable be ‘0’ when the state of casting is ideal, and ‘1’ just before occurrence of breakout, i.e., the most abnormal state. Since the process is continuous time, it is natural that this “abnormality” of casting will vary in a continuous manner (i.e., lower order derivatives exist) between 0 and 1. Importantly, it will stay closer to 0 most of the time, except on the verge of a breakout when it will *continuously and monotonously* shoot up, within a short span of time, from a lower value towards 1—reflecting the rapid adverse digression of the process.

Now let us look at the natural tendency of the output from the ANN that is trained at two extreme conditions, i.e., either at “process normal” and desired output at ‘0’, or at process on verge of breakout with desired output at ‘1’. The compulsion to train the ANN only at these binary levels flows from *the lack of accurate knowledge/awareness of the actual health of the process* at past operational periods; all that is known (in the absence of a hypothetical instrument as discussed above) is whether the process was stable or was about to breakdown. The output from the trained ANN though—running on ‘application data’ (as defined in Sect. 2)—is never a perfect ‘0’ or ‘1’ but mostly in-between, reflecting the fact that real-time input conditions are not matched perfectly with either those that generated a ‘0’, or a ‘1’, at the time of training, but lie between these two. *This means that the ANN (as long as it is validated software) has not only learnt the two extreme states of casting but has also aligned itself with the intermediate characteristics from which the bulk of actual training data is drawn, and places the current instance accordingly to lie between these extremes. The implication is that the ANN output varies with the continuity of the process, and behaves analogous to the variable “process abnormality” defined above*. Hence, it is also expected to demonstrate the characteristics of continuity and monotonicity that prevail on the verge of a breakout—the value shooting up sharply from a low normal to a high abnormal during rapid adverse digressions of the real process.

One may observe a significant shift in paradigm in the above assessment of the ANN output—from the “classification paradigm” that was discussed in Sect. 2 to the “continuity paradigm” which comes naturally to it and is yet in harmony with the natural continuity of the real process itself. *It is this shift in paradigm that transforms an impossible situation for practical application* (as explained in Sect. 2) *into one with easy potential for real-time tracking of emerging adverse digressions*.

To realize this potential, a few hurdles have to be crossed. One of these relates to output noise. As seen in Fig. 5, when application data are passed into the ANN, there are quite a few cases of “misclassification”, which may be called as large errors. To neutralize these errors in a real-time framework, the best option is to consider moving averages, i.e., average taken over the past few discrete values in time. This is illustrated in Fig. 6a, where the red and gray balls now reflect assessments at different instances, across two different time periods—one of good (say gray) and another of "bad" (red) casting, with some instances assessed and placed wrongly. While the red and gray balls can potentially be placed anywhere in the field, the moving averages—denoted by the respective crosses—will necessarily remain confined within the respective circles and well within their respective good and bad domains. Hence, the output from the ANN—which we may call "Abnormality Index" need not be used directly but only after taking moving average across a short time span. This may be denoted as "Moving Abnormality Index" or MAI.

The mechanism for real-time tracking of emerging adverse digressions on the basis of the "continuity paradigm" is illustrated in Fig. 6a and b. The figure has two parts aligned mutually. The lower part shows the trend of conceived parameter "actual process abnormality" across time. Let us say, hypothetically, that there is an instrument to measure this variable and it returns a precise value. The black curve shows this trend across time. In the initial period, casting is normal and the value is low. A breakout-like situation emerges towards the end of the considered time window. At that point, the abnormality shoots up and approaches 1, followed by a breakout unless that is detected and then prevented by corrective action. The blue (lighter) curve denotes the corresponding predicted MAI values. Because the ANN has a residual level of inaccuracy, there is always a difference with the precise (hypothetical) value. When the abnormality is approached, however, the MAI also shoots up—though remaining somewhat deviant from the actual value. *It is this sudden rise of the MAI that is interpreted as the impending breakout.* The upper part of Fig. 6, i.e., Fig. 6a, reflects two zones—good and bad—with respective instances and bounds of MAI shown. As the breakout period approaches and the

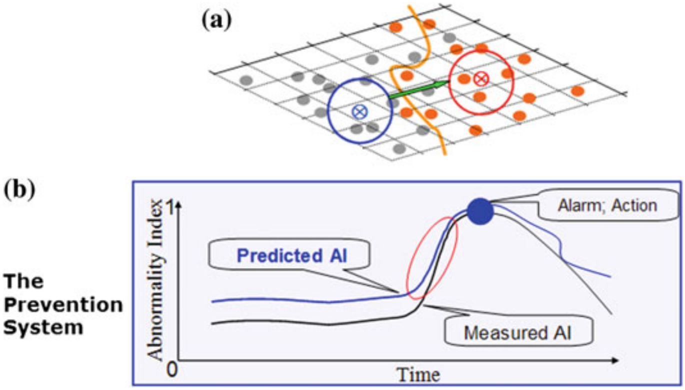


Fig. 6 Illustration of the dynamics of the continuity paradigm. There is always a difference between the ANN-predicted Abnormality Index (Al), and the "measured" or precise value. However, when process is normal, both are low—and under adverse digressions both shoot up. We look at the fast change in value—its jump from one regime into another, not the absolute numbers of the ANN—predicted Al

MAI changes rapidly, it effectively breaks through the good zone border into the bad. The alignment in time of the green arrow in the upper figure, with the pink oval in the lower, may be noticed—this is the "transition" period. Towards the end of the transition period, warning is issued and action taken.

The steps of the algorithm for detecting rapid adverse digressions are as follows:

Step 1 *The ANN runs on a processor that receives six field inputs in real time and generates an Abnormality Index (AI) in* 1-s *cycles*

Step 2 *The instantaneously predicted AI is used to generate the Moving Average computed over a period of past 'n' seconds called MAI*

Step 3 *At every instant there is calculated a Normal Index of Current Casting (NICC) which is an average value of AI of past* 1 h *of casting (explanation follows below)*[1]

Step 4 *The MAI is compared against the NICC and if it is some threshold ratio (or 'Rise Factor') more than NICC it is taken as indication of digression into bad casting—in which case the remedial action is flashed on HMI screens for operators to execute*

Step 5 *In either case—normal or digression detected—execution returns to Step 1.*

4 Results, Generalization and Further Work

The system CAFOS—Caster Advance Fault Forecasting System—has been put online at a steel manufacturing facility. Figures 7, 8, 9 and 10 illustrate results on four different casting sequences, two of which ended in breakouts and two normal. The MAI, NICC, and actual alarms—in red bold—are shown. Figures 7 and 8 relate to breakouts, while Figs. 9 and 10 represent good cases—shown here to demonstrate that false alarms are rare. On an average less than one false alarm is raised in every sequence, while approximately sixty sequences have one breakout. Recalling the discussion towards the end of Sect. 2, it follows that for preventing one breakout control action needs to be taken about fifty times—improvement by a factor of forty over the earlier estimated two thousand control actions needed on a near-zero error ANN (impractical in real). This has been made possible primarily by a shift in paradigm—from that of classification to that of continuity.

[1] *Explanation of NICC*:

The NICC is actually the baseline AI predicted by the ANN over periods of good casting. *In the* 168 h *of the week, it may be assumed that approximately* 160 h *of good casting is performed. In other words,* 95 % *of the time casting is "good"*. However, the value of AI can drift based on specific casting conditions. Hence, the average AI of last 1 h of casting can be considered as the reference baseline abnormality index of good casting—called Normal Index of Current Casting or NICC. In a different technical domain, the Normality Condition can be extracted based on analogous considerations, with appropriate variation of time window.

Fig. 7 A breakout case with sequence ended midway. Notice fluctuations in MAI; CAFOS captures breakout prior to occurrence

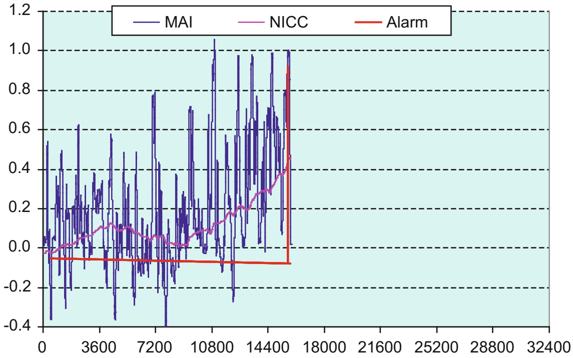


Fig. 8 Another breakout case, smooth casting terminating in sudden breakout; CAFOS predicts in advance

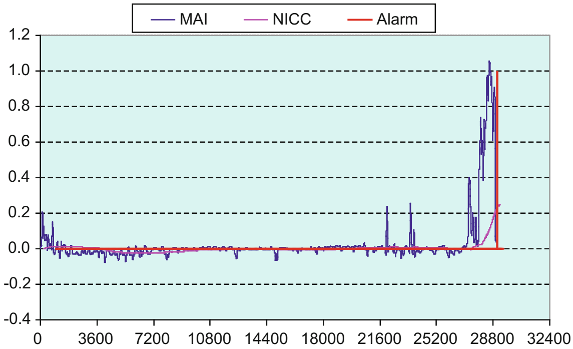
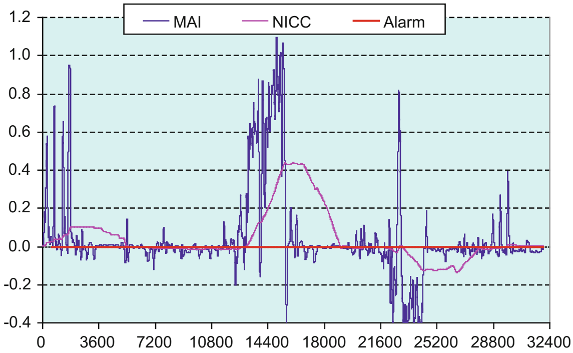


Fig. 9 A 'good' case, normal termination



Let us now return to what we had set out to achieve—a reliable mechanism for advance prediction of impending adverse digressions in continuous–time systems that can be applied generically to any system—natural or manmade—satisfying the "continuous-time" criterion. We had identified two non-trivial tasks as baseline requirements for such a mechanism—one, identifying the variables of the system that combine together to engender the digression, and two, deciphering the

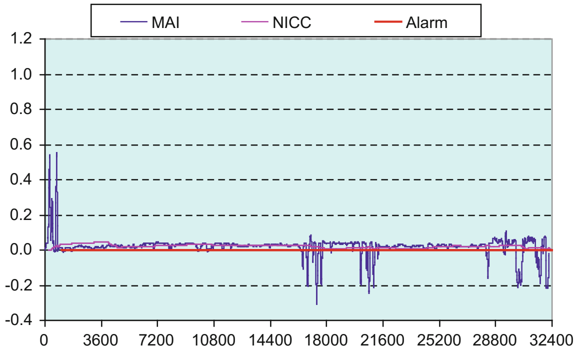


Fig. 10 Another 'normal' case

characteristics of this combinatorial relationship. The first task is domain specific and involves domain knowledge and basic statistics. The second can be attained using any ANN of good accuracy. The two tasks on accomplishment will still elude a real-time mechanism of high reliability to predict rapid adverse digressions. *It is the current development that completes the chain—which is based on a shift of paradigm from that of "classification" to that of "continuity". This is the core contribution of this work.*

The continuous casting process in steel manufacturing is an instance of successful application of this generic mechanism. All the constructs that have been created here, the "moving abnormality index", the "normality index", the "rise factor", etc., will hold equally well in any other process. *However, their timescales will be a function of the nature of the process.* In fact, industrial processes are more difficult candidates for application of this mechanism as by their very nature they tend to drift—complicating the attainment of the two aforementioned baseline tasks.

Considering the possibility of drifts in an industrial process, it is proposed to use adaptive systems for online modification of the ANN that is used for generating the abnormality index. In this application on the breakout forecasting problem, investigations on an 'Adaptive Critic' approach [5] are underway.

5 Conclusions

A generic methodology has been developed for online prediction of rapid adverse digressions in continuous-time systems, prior to the actual occurrence of these digressions when predefined corrective action can be taken to mitigate progression towards the undesirable state. The complete methodology itself consists of three stages. The first two stages that deal with recognition of participatory parameters and deciphering their combinatorial relationship—the attainment of which engenders the digression—are domain specific and involve statistical analysis and standard prediction tools like Artificial Neural Networks (ANN). The third stage that transforms the ANN output into a reliable measure of the instantaneous digression of the system from its normal is generic across domains and has been developed in this work. The conceptualization, development and validation of this methodology have been executed on a specific industrial process—the continuous casting process in steel manufacturing—and are described in this paper.

References

1. World Steel University website, continuous casting link. https://en.wikipedia.org/wiki/Continuous_casting Accessed 20 Oct 2015
2. Thomas, B.G.: Modelling of the continuous casting of steel—past, present and future. Metall. Mater. Trans. **33B**, 795–812 (2002)
3. Bhattacharya, A.K., Chithra, K., Jatla, S.S.V.S., Srinivas, P.S.: Fuzzy diagnostics system for breakout prevention in continuous casting of steel. In: Proceedings of 5th WCICA (2004). doi:10.1109/WCICA.2004.1343100
4. Bhattacharya, A.K., Srinivas, P.S., Chithra, K., Jatla, S.S.V.S., Das, J.: Recognition of fault signature patterns using fuzzy logic for prevention of breakdowns in steel continuous casting process. In: Lecture Notes in Computer Science, vol. 3776, pp. 318–324. Springer (2005). http://www.springerlink.com/content/p187n3722w2885v7/?p=f7974a892990423681ff52029ddd1dd6&pi=3
5. Yen, G.: Adaptive critic neural network control. In: Industrial Electronics Handbook—Intelligent Systems, 2nd edn, Ch. 17, pp. 302–331. CRC Press (2011)

Determination of Suitable ANN Architecture for Groundwater Fluoride Prediction

Kumari Neeta and Pathak Gopal

Abstract Using soft computing methods is considered as an innovative approach for constructing a computationally intelligent system, having humanlike expertise in a specific domain. They learn the system and adapt themselves with the varying pattern and at the end they take decisions for future trends. The applications of these techniques, for prediction of trends in environmental problems, are increasing day by day. In the present study, the ANN model was used for groundwater fluoride prediction in western parts of Jharkhand district, India. MATLAB software is used for artificial neural network and graphical user interface (GUI) toolboxes. The performance criteria used for the measurement of efficiency and accuracy were root mean square error (RMSE) and regression coefficient (R^2). The 6-year dataset for fluoride level in groundwater is used for prediction. The input parameters used were depth, groundwater level, electrical conductivity (EC), pH, chloride (Cl), bicarbonates (HCO_3), calcium (Ca), magnesium (Mg), and sodium (Na). The search of suitable architecture was done by the evaluation of predicted output with the actual fluoride level data. Simulation results reveal that three-layered feed forward backpropagation type of network with Levenberg–Marquardt (LM) training algorithm is the promising architecture for groundwater fluoride prediction.

Keywords Soft computing · Artificial neural network graphical user interface (GUI) toolboxes · Levenberg–Marquardt training algorithm · MATLAB · Performance criteria · Fluoride · Groundwater

K. Neeta (✉) · P. Gopal
Department of Civil & Environmental Engineering, Birla Institute of Technology,
Mesra, Ranchi 835215, Jharkhand, India
e-mail: neetak@bitmesra.ac.in; neeta.sinha2k7@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_8

1 Introduction

Water in the saturated zone is called groundwater. Fluoride is found in the groundwater in the form of dissolved ions. A small amount of fluoride is an important component for bone mineralization and dental enamel formation [1]. Fluoride bearing rocks such as cryolite, fluorite, fluorspar, hydroxyapatite and fluorapatite are the major sources of fluoride in groundwater [2]. The amount of fluoride present in groundwater depends on many factors such as presence of fluoride bearing minerals and their solubility in groundwater, flowing groundwater velocity, temperature, pH, and concentration of ions in water mainly Ca and HCO_3^- ions [3, 4]. As per the WHO (World Health Organization) standards, the maximum and permissible limits for drinking water fluoride are 1.5 and 1.0 mg/l, respectively [5]. The Indian standard specifications for drinking water IS 10500 prescribed the desirable limit for drinking water fluoride as 0.6–1.0 mg/l and the maximum limit is extended to 1.5 mg/l. If the fluoride content in the drinking water is below 0.6 mg/l, the water should be rejected. Artificial neural network is the most viable computational model for many hydrology problems as it can identify the complex non-linear relationship between input and output parameters. Regression analysis is one of the data processing methods but is not as efficient in understanding the relation between variables as ANN is [6]. It even extracts pattern from complicated and imprecise data which cannot be noticed by human brain or conventional computers or other computational techniques [7]. Literature reports that ANN is a very good technique in rainfall–runoff modeling, reservoir inflow forecasting, water quality prediction [8], groundwater simulation [9] and variation in groundwater quality [10, 11]. The most critical part of it is the determination of network structure which is done by trial and error method which is time consuming but it has an advantage that it can learn and apply the knowledge obtained from the previous data or training. Then validation of learning model can be done by validation data set to reach the desired error goal. Repeated training with different topology increases the accuracy of the model and can reach up to maximum. Thus, the proper determination of ANN model structure and training can result in optimal structure which can be efficiently used in forecasting or prediction in water resource management or planning.

2 Materials and Methodology

2.1 Study Area and Data

Jharkhand means ‘The Forest Area’, nearly 29 % of the total geographical area is under forest foliage. Jharkhand is the 28th state in India having 13th rank as per the population and 15th rank considering the geographical area having 30 % of the total population as tribal. The population of Jharkhand is 32,966,238 (2011 census) and

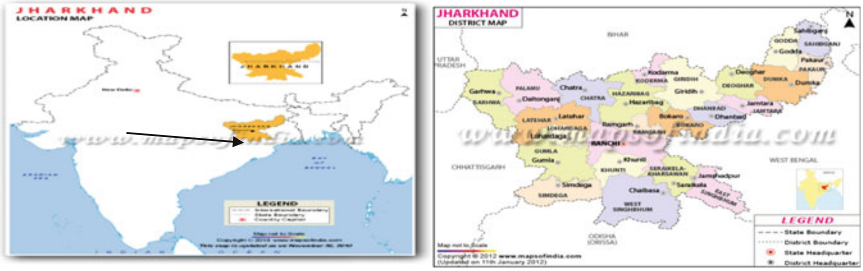
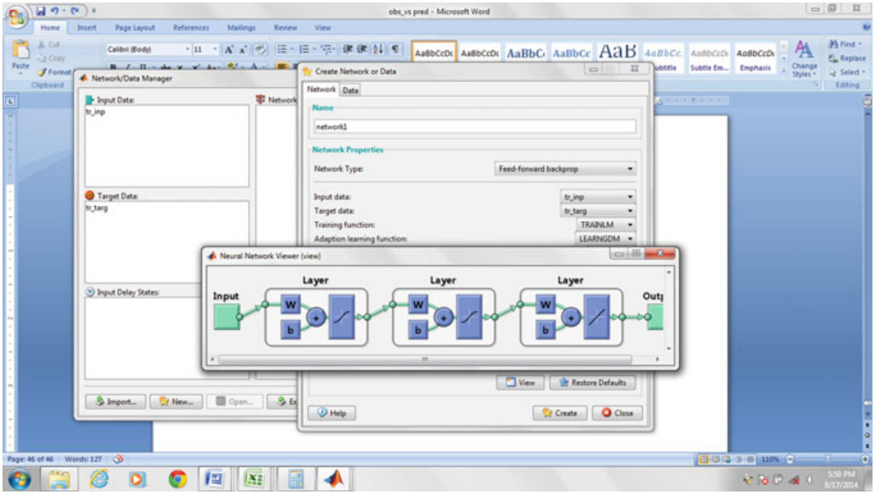


Fig. 1 Location map of study area [16]



Create Network or Data

Network | Data

Name: network1

Network Properties

Network Type: Feed-forward backprop

Input data: tr_inp

Target data: tr_targ

Training function: TRAINLM

Adaption learning function: LEARNGDM

Fig. 2 The screen caption (view) of proposed network (network/data manager)

covers an area of 79,714 km^2. Six years of both primary and secondary type of data are acquired from various government agencies for this study and meteorological data acquired from IMD Pune (Figs. 1 and 2).

2.2 *Artificial Neural Network Model*

MATLAB software ANN GUI toolboxes were used for this study. The data are acquired from various agencies pertaining to the water quality data of western parts of Jharkhand with special focus on fluoride. There is no certain fixed rule base for ANN model structure. It is determined on the basis of trial and error basis. The performance of the model generally depends on number of hidden layers, number of hidden neurons in one layer, learning rate and learning algorithm, momentum factor and transfer function. In ANN architecture, input layer has neurons equal to the number of input parameters, varying neurons in hidden layer and output layer has single neuron. The 165 dataset/patterns are divided as: 80 % of the dataset is used for training and 20 % is used for validation and testing. In this process, multilayered feedforward neural network is used with LM training algorithm. The best fit values are MSE = 0, R^2 = 1. Up to 75 % (0.75) of R^2 value is significant. Results are compared and the best model is selected and this process is reported in almost all literature [12]. Model is trained for five sets of input parameters and then testing and validation is done.

Data Normalization: Through min–max normalization the original data are linearly transformed. In this study, the data are normalized in the form of [0, 1]. The below formula is suggested by Jain and Bhandare in 2011 [13].

$$d' = d - \min(p) / \max(p) - \min(p);\ \text{Here}\ \min(\mathrm{p}) = 0, \max(\mathrm{p}) = 1, \text{and}\ \mathrm{d}'\ \text{is the new value of d.} \tag{1}$$

Denormalization of output data: The output is in the normalized form so it should be denormalised to get the original value. The formula used for denormalization is as follows:

$$\mathrm{C} = (d')(\max(p) - \min(p)) + \min(p);\ \text{Here}\ \mathrm{d}'\ \text{is the normalized output value of d.} \tag{2}$$

3 Model Architecture Determination

The model structure determined on the basis of trial and error method using ANN GUI toolboxes in MATLAB software. Network creation needed different strategies such as the effect of various neurons per layer, number of layers in the neural network, effect of transfer (activation) function, effect of training algorithm, effect of learning algorithm, effect of momentum factor and learning rate.

3.1 Performance Evaluation of the Model

The best ANN architecture was chosen with the comparison with all the chosen models and the selection is based on lowest RMSE and linear regression percentage, R.

4 Results and Discussion

Here, MSE is mean squared error and R is linear regression percentage, Epoch-MSE is the epoch at which best validation took place and the error is least. The maximum correlation between output and target and least MSE was found at the 10 neurons and after that 25, 30, 5 were the next best. On this basis, trial was done for various combinations of hidden neurons in the layers. The number of layers is to be decided which will give best optimal result.

With three layers, maximum R and least MSE are recorded in comparison to other two. So three layers are selected for the ANN architecture. As per literature, after three layers the performance of network decreases [7, 14] (Tables 1, 2, 3 and 4).

Table 1 The effect of number of neurons per layer

No. of neurons	Linear regression percentage; R (all)	Performance value (MSE)	Epoch-MSE
5	**0.90899**	**0.018374**	**12**
10	**0.94745**	**0.027241**	**12**
15	0.84978	0.066638	2
20	0.88444	0.020866	4
25	**0.92919**	**0.029762**	**3**
30	**0.91095**	**0.064125**	**8**
35	0.83911	0.039202	3
40	0.88273	0.020284	8
45	0.88165	0.058607	3
50	0.86467	0.067265	6
55	0.85595	0.040826	8
60	0.88316	0.046806	6

Table 2 Effect of layers in the neural network

Number of layers	R (all)	Performance value (MSE)	Epoch-MSE
2	0.93537	0.034164	12
3	**0.96023**	**0.0194**	**5**
4	0.92078	0.067466	15

Table 3 Effect of training algorithm—by changing training algorithm

Epoch	Algorithm	R all	MSE	MSE-Ep
200	**LM**	**0.95522**	**0.026321**	**6**
200	RP	0.90981	0.01219	95
200	GDM	0.39347	0.083728	200
200	BR	0.9187	0.40012	30
200	GD	0.42884	0.13068	200
200	GDX	0.86704	0.053364	127

Table 4 Effect of learning algorithm—by changing learning algorithm

Epoch	Training algorithm	Learning algorithm	R all	MSE	MSE-Ep
200	LM	LEARNGD	0.88867	0.048897	7
200	**LM**	**LEARNGDM**	**0.94289**	**0.023743**	**6**

Here, LM is the Levenberg–Marquardt algorithm; RP is the resilient backpropagation; GDM is the gradient descent with momentum; BR is the Bayesian regularization; GD is the gradient descent method; GDX is the variable learning rate backpropagation.

The best result is associated with Levenberg–Marquardt training algorithm. Most research papers have used Levenberg–Marquardt backpropagation with log-sigmoid or tan-sigmoid or pure linear transfer function [7, 12, 15].

Here, the best result is recorded with LEARNGDM learning algorithm.

The performance and regression plots of the best finding in case of neurons, number of layers, training algorithm and learning algorithm are given as below (Fig. 3):

Three layers (Fig. 4).

Training algorithm: Levenberg–Marquardt (Fig. 5).

Learning algorithm (Fig. 6).

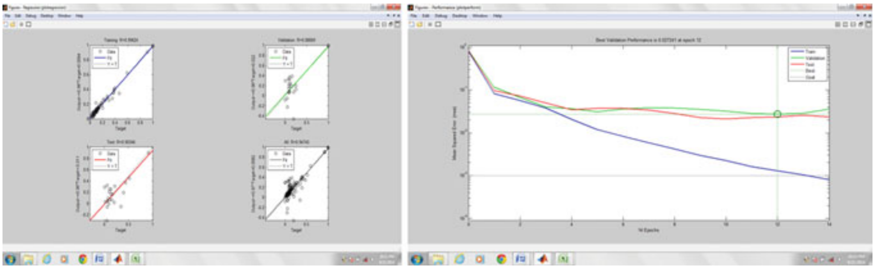


Fig. 3 Regression and performance plot for 10 neurons

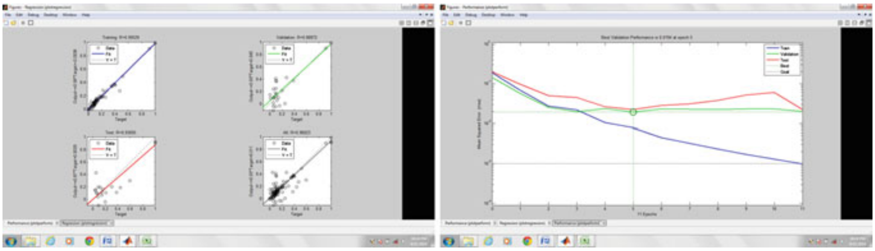


Fig. 4 Regression and performance plot for three layers

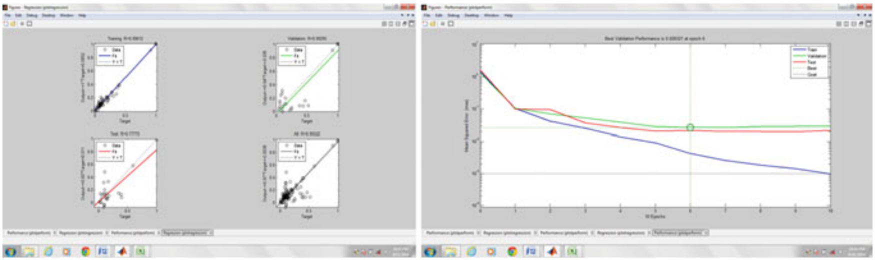


Fig. 5 Regression and performance plot for LM training algorithm

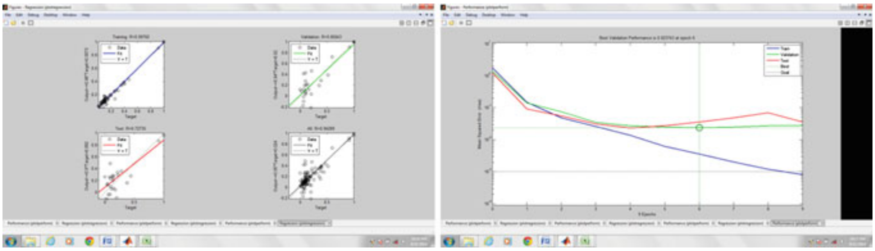


Fig. 6 Regression and performance plot for learning algorithm

5 Conclusion and Recommendation

Thus, the best ANN architecture for fluoride prediction in groundwater was found as below: feedforward backpropagation, Levenberg–Marquardt algorithm, LEARNGDM, three layered. Number of neurons in the layer is taken by the trial and error method; many combinations were run and the following three had shown some good results, so with different input combinations, the following three options were considered in finding the optimal model. They are 6-12-1, 9-15-1 and 12-20-1. Transfer function: Tansig/Logsig for first two hidden layers and Purelin for output layer, Epochs: 200, Training error goal: 0.001, Mu: 0.5, 0.7, 0.9, Mu (increment): 1.05, Mu (decrement): 0.2, 0.4, 0.7. The model performance was assessed by varying the number of input neurons.

References

1. Wood, J.M.: Biological cycle for toxic elements in the environment. Science **183**, 1049–1052 (1974)
2. Agarwal, V., Vaish, A.K., Vaish, P.: Ground water quality: focus on fluoride and fluorosis in Rajasthan. Curr. Sci. **73**(9), 743–746 (1997)
3. Chandra, S.J., Thergaonkar, V.P., Sharma, R.: Water quality and dental fluorosis. Ind. J. Publ. Health **25**, 47–51 (1981)

4. Largent, E.J.: The Health Aspects of Fluoride Compounds. Ohio State University Press, Columbus, O.H. (1961)
5. WHO.: International Standards for Drinking Water, 3rd edn. WHO, Geneva (1971)
6. Sirisha, P., Sravanti, K.N., Ramakrishna, V.: Application of artificial neural networks for water quality prediction. Int. J. Syst. Technol. **1**(2), 115–123 (2008)
7. Mayilvaganan, M.K., Naidu, K.B.: Application of artificial neural network for the prediction of groundwater level in hard rock region. CCSEIT, CCIS 204, pp. 673–682 (2011)
8. Hsu, K., Gupta, H.V., Sorooshian, S.: Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res. **31**(10), 2517–2530 (1995)
9. Jalalkamali, A., Jalalkamali, N.: Groundwater modeling using hybrid of artificial neural network with genetic algorithm. Afri. J. Agric. Res. **6**(26), 5775–5784 (2011)
10. Chang, F.-J., Kao, L.-S., Yi, M., Chen, W.L.: Artificial neural networks for estimating regional arsenic concentrations in a blackfoot disease area in Taiwan. J. Hydrol. **388**, 65–76 (2010)
11. Cho, K.H., Sthiannopkao, S., Pachepsky, Y.A., Kyoung-Woong, K., Joon Ha, K.: Prediction of contamination potential of groundwater arsenic in Combodia, Laos, and Thailand using artificial neural network. Water Res. **45**, 5535–5544 (2011)
12. Jha, M.K., Sahoo, S.: Efficacy of neural network and genetic algorithm techniques in simulating spatio-temporal fluctuations of groundwater. Hydrol. Proc. (2014). doi:10.1002/hyp.10166. http://www.Wileyonlinelibrary.com
13. Jain, Y.K., Bhandare, S.K.: Min-max normalisation based data perturbation method for privacy protection. Int. J. Comput. Commun. Technol. **2**(8), 45–50 (2011)
14. Dash, N.B., Panda, S.N., Remesan, R., Sahoo, N.: Hybrid neural modeling for groundwater level prediction. Neural Comput. Appl. **19**, 1251–1263 (2010)
15. Maedeh, P., Abbasi, M.N., Nabibidhendi, G.R., Abyaneh, Z.H.: Application of artificial neural network to predict total dissolved solids variations in groundwater of Tehran plain. Iran. Int. J. Environ. Sustain. **2**(1), 10–20 (2013). ISSN 1927-9566
16. Maps Of India. http://www.mapsofindia.com

Neural Network Based Breakout Predicting System for All Four Strands of Caster in a Continuous Casting Shop—A Case Study

Md Obaidullah Ansari, J. Ghose and R. Kumar

Abstract In continuous casting of steel, liquid steel is continuously solidified into a finished product. Bokaro Steel Limited have slab caster and during continuous casting, one of the major operational irregularities encountered is breakout. A breakout is a hazard whereby the thin shell of the strand breaks, allowing the still-molten metal inside the strand to leak out from the mould, thus necessitating an expensive machine shutdown. Hence the breakout has a profound influence on the caster availability; affecting the productivity, cost of production, damage of machinery, safety hazards, etc. Breakouts occur due to the presence of a primary crack before solidification of liquid steel. Controlling the primary cracks lead to reduce chances of breakouts. Breakout prediction system of Bokaro Steel Limited uses only thermocouples temperature as an input of logic based algorithm controlled by strand PLC to predict breakout and generate alarm. This system sometimes generates lots of false alarm and sometimes even fails to generate an alarm before breakout. In this paper, two back-propagation neural networks are developed to predict the presence of primary crack that might lead to breakout of liquid steel. One is neural network1 train only with one input parameter (temperatures of thermocouples attached to the mould) and the other is neural network2 train with three input parameters (temperatures of thermocouples, casting speed and mould level). Both the neural network1 and network2 have only one output. Output of neural networks is either 0 for no alarm (no primary crack) or 1 for alarm (presence of primary crack) in breakout conditions. Comparison of the network1 with network2 showed that network2 has less output error, better performance and identifies primary crack more accurately. Performance of neural networks are done using Levenberg–Marquardt training algorithm, and also log sigmoid transfer function is used for both the hidden and output layer. The neural networks model training, validation and testing is done by simulating in Mat-Lab/Simulink.

M.O. Ansari · J. Ghose (✉)
Department of Production Engineering, BIT, Mesra, Ranchi, India
e-mail: joyjeetghose@gmail.com

R. Kumar
Bokaro Steel Limited, SAIL, Bokaro, India

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_9

Keywords Continuous casting · Mould · Thermocouple temperature · Breakout prediction · Neural network · Mat-Lab/Simulink

1 Introduction

Steel Authority of India Limited (SAIL) is the leading steel making company in India. Bokaro Steel Limited (BSL) is one of the integrated steel plants of SAIL in India producing mainly flat products. In continuous casting, the molten steel from the steelmaking operation or ladle metallurgy step is cast directly into semi-finished shapes (slabs, blooms, and billets) [1]. Continuous casting represents a tremendous savings in time, labor, energy and capital. By casting the steel directly into semi-finished shapes, the following steps are eliminated: ingot teeming, stripping, & transfer to soaking pits and primary rolling. Continuous casting also increases yield and product quality. In continuous casting shop (CCS) one of the major operational irregularities encountered is breakout. A breakout is a hazard whereby the thin shell of the strand breaks, allowing the still-molten metal inside the strand to leak out from the mould, thus necessitating an expensive machine shutdown. Hence, the breakout has a profound influence on the caster availability; affecting the productivity, cost of production, damage of machinery, safety hazards, etc. [2]. Of all the breakout prediction systems, mainly two types of breakout prediction method are popular. One is thermocouple based breakout prediction system [3] and the other is mould friction based breakout prediction system [4]. The thermocouple based system is reliable and easy to use, therefore, BSL is using thermocouple measuring method. In thermocouple measuring method, mould temperature is measured using thermocouples which are attached on the straight mould plates made of copper. The dimensions of the mould are 900 mm length, 2000 mm width and thickness 200–225 mm as shown in Fig. 1. The thermocouples are attached at the wide sides and at the narrow sides of the mould as shown in Fig. 1. Breakout prediction system of Bokaro Steel Limited uses only thermocouples temperature as an input of logic based algorithm controlled by strand programmable logic controller to predict breakout and generate alarm. This system generates lots of false alarm and many a times even fails to generate an alarm before breakout. Thus, there is a tremendous need to have a BOPS system which can be very effective in predicting breakouts. Literature survey shows that artificial intelligence can be used to develop an effective BOPS system as shown in Table 1.

Literature survey shows that artificial intelligence (mainely neural network) can be used to develop an effective BOPS system. From the above works it is clear that the all papers used different types of tools, algorithms and techniques to train the neural network. Most of the work as shown in the above table used mould thermocouples temperature as an input of neural network.

Fig. 1 Copper mould used for continuous casting

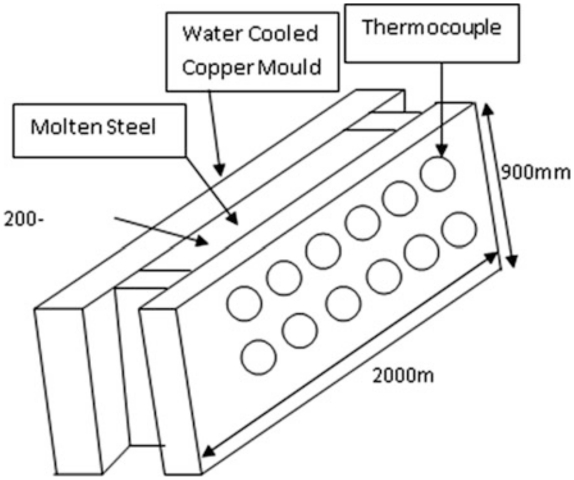


Table 1 Previous works on BOPS system

References	Neural network input	Tools, algorithms and methods
Mugwagwa et al. [5]	Thermocouple temperature	Feed-forward back propagation neural network
Cheng et al. [6]	Thermocouple temperature	BP neural network based on genetic algorithm
Qing et al. [7]	Thermocouple temperature	Neural network based on the least squares support vector machine (LSSVM)
Ben-guo et al. [3]	Thermocouple temperature	BP neural network, based on the Levenberg-Marquardt (LM) algorithm
Ren et al. [8]	Thermocouple temperature	Fuzzy neural network based on T-S (Takagi-Sugeno) model
Li et al. [9]	Thermocouple temperature, mould friction	BP neural network and dynamic fuzzy neural network
Zening et al. [10]	Thermocouple temperature	BP neural network combined with actual situation is redesigned (feedback)
J. Adamy [11]	Thermocouple temperature	Fuzzy meal automation to recognize the temperature patterns

2 Causes of Breakouts

Breakout can occur due to many causes. To develop an effective BOPS system, one has to know the causes of breakouts. Approximately last 6 years data of breakouts and their causes is collected from operational log books of BSL. Table 2 shows the number of breakout and their causes of all four strands of caster in BSL.

From the Table 2, it clearly shows that the major causes of breakouts are due to sticker formation, inappropriate casting speed, inappropriate mould level, and improper mould taper adjustment.

Table 2 Year wise data of causes and occurrence of breakout

Breakout in moulds of all four strands of caster	2008–09	2009–10	2010–11	2011–12	2012–13	2013–14 (till Dec)	Total
Sticker formation	11	5	5	7	16	1	45
Packing leakage	0	1	1	0	2	1	5
Dummy bar	0	0	0	0	0	0	0
Tundish slag flow	0	2	2	0	0	0	4
Casting speed problem	4	3	3	2	4	4	20
Low ‘Ca’ ring problem	0	1	1	1	0	0	3
Casting powder	4	1	1	1	0	0	7
Taper/mould problem	2	5	5	5	1	2	20
Foot roll problem	0	0	0	0	0	0	0
Spray cooling problem	0	0	0	2	0	0	2
Mould level/MOM	4	2	2	2	1	1	14
High nitrogen/‘B’	0	1	1	0	1	0	3
Total	25	21	21	20	25	11	123
Average breakout	**22.36 per year**						

3 Loss Due to Breakout

Breakouts in continuous casting may result in huge production losses. Table 2 clearly shows that there is an average 22.36 breakouts per year. Each breakout causes the caster to be shut down for about 4 h. One hour of shut down means loss of one heat (280 Tons of liquid steel). Table 3 shows that losses due breakout in last six years.

BOPS system is an important part of CCS. In BSL logic based BOPS system is used and this system fails to predict breakout. Annually 22.36 breakouts take place as shown in Table 2 and 89.44 heat losses per year as shown in Table 3 due to the

Table 3 Production losses due to breakouts

Particulars	Value
No. of breakout (April’08 to Dec’14) per year	22.36
After breakout an average 3–4 h required to restart the caster	
Average delay due to breakout = (4 × 22.36) = **89.44 h/years**	
Heat loss due to breakout per year	89.44
Average weight of one heat in ton	280
Annual production loss in term of liquid steel in ton	250432
Cost of one ton of steel	≠26000

failure of BOPS system. Most of the literature survey shows that artificial intelligence (mainly neural network) is used to predict breakout. However, all work used only thermocouple temperature as an input for their tools (neural network, fuzzy based BOPS system) as shown in Table 1. From Table 1 it is clear that not only thermocouple temperature but casting speed, mould level and taper/mould problem is also responsible for breakout of liquid steel. Therefore, a neural network has been developed with three input parameters, i.e. thermocouples temperature difference, casting speed and mould level to predict breakouts.

4 Network Design

A two layer-feed forward BP (Back Propagation) neural network is one of the most widely used model in breakout prediction system [12–16]. In this work the breakout prediction system has been constructed based on a BP neural network model which is optimized with Levenberg Marquardt (LM) training algorithm. Levenberg Marquardt (LM) training algorithm is considered to be the quickest training method. In this work two back-propagation neural networks are developed to predict the presence of primary crack that might lead to breakout of liquid steel. One is neural network1 train only with one input parameter (temperatures of thermocouples attached to the mould) and other is neural network2 train with three input parameters (temperatures of thermocouples, casting speed and mould level). The neural network1 and network2, both networks have only one output. Output of neural networks is either 0 for no alarm (no primary crack) or 1 for alarm (presence of primary crack) in breakout conditions as shown in Figs. 2 and 3, respectively. Both networks have only one output. Output of neural networks is either 0 for no alarm or 1 for alarm in breakout conditions.

4.1 Training Data for Networks

The data for training the developed networks were collected from log book of BSL continuous casting shop. The breakout data used was 10 × 100 matrix, representing 100 samples of 10 elements. The target data was 1 × 100 matrix (target is always 1 or alarm or presence of primary cracks on semi-solidified slab). The data

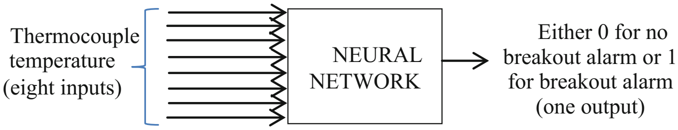

Thermocouple temperature (eight inputs)
NEURAL NETWORK
Either 0 for no breakout alarm or 1 for breakout alarm (one output)

Fig. 2 Neural network1 with only one input parameter

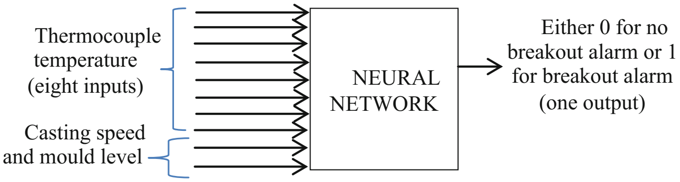

Thermocouple temperature (eight inputs)
Casting speed and mould level
NEURAL NETWORK
Either 0 for no breakout alarm or 1 for breakout alarm (one output)

Fig. 3 Neural network2 with three input parameters

of no breakout used was 10 × 100 matrix, representing 100 samples of 10 elements. The target data for no breakout was a 1 × 100 matrix (target is always 0 or no alarm or absence of primary cracks on semi-solidified slab).

In above input data there are 10 inputs (8 inputs for mould thermocouple temperature, 1 input for casting speed and 1 input for mould level).

4.2 Normalization by Scaling Between 0 and 1

The input signals were of varied magnitude. To eliminate the effect of the variations the input data was normalized as follows.

The normalized value of e¡ for variable E in the i^{th} row is calculated as:

$$\text{Normalized}\,(e_i) = \frac{e_i - E_{\min}}{E_{\max} - E_{\min}}$$

$E_{\min}$ = the minimum value for the variable E; $E_{\max}$ = the maximum value for the variable E; if $E_{\min}$ and $E_{\max}$ are equal, then $e_i = 0.5$.

4.3 Training Parameters

The Levenberg–Marquardt (trainlm) with the mean square error performance function was used as the training method. The network accepts ten inputs (eight mould thermocouple temperature, casting speed and mould level), and a total of ten neurons make up the input layer. The hidden layer is increased or reduced in the iterative training process. However, after iterating, 30 neurons were used for the hidden layer. The transfer function is the log-sigmoid because of its ability to accept large positive or negative inputs and squash them to between 0 and 1.

4.4 Training Neural Network1 and Network2

The neural network2 is trained with ten inputs (8 inputs for mould thermocouple temperature, 1 input for casting speed and 1 input for mould level). The neural network1 is trained with eight inputs (8 inputs for mould thermocouple temperature only). Both neural network 1 and neural network 2 were trained using normailized data. Neural network pattern recognition tool was used to develop the neural networks. The general predictive neural network is modelled with 10 inputs, 30 hidden neurons and one neuron in the output layer. The neural network1 and network2 were trained with 200 samples (normalized data). Random division of the 200 samples is shown in Table 4.

4.5 Result and Discussion

Performance of the neural network1 and neural network2 after the training is shown in Fig. 4a, b, respectively.

Best performance of the neural network1 and network2 is at epoch 25 and 29, respectively. From Fig. 4a, b it is clearly shown that neural network2 performance is good as compared to the network1. Now to compare the output error of the neural networks. Output error present in neural network2 is less as compared to the neural network1 as shown in Figs. 5 and 6.

Table 4 Random divide 200 sample for the networks

Training	Validation	Testing
80 % data	10 % data	10 % data
160 samples	20 samples	20 samples

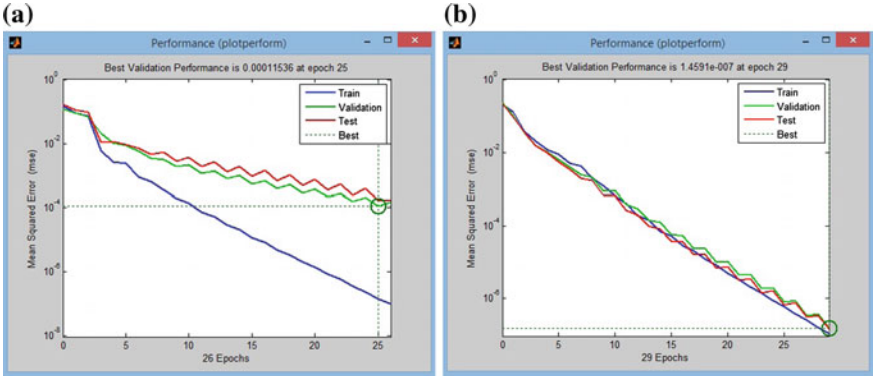


Fig. 4 **a** Best performance of the network1, **b** best performance of the network2

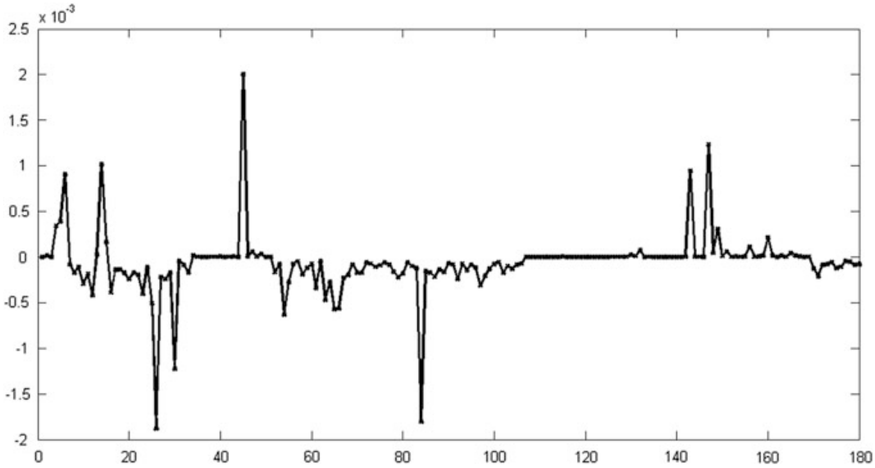


Fig. 5 Output error of network1

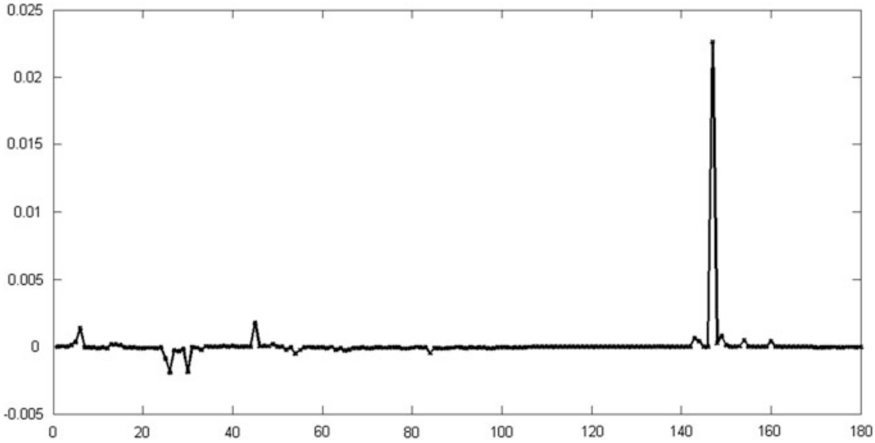


Fig. 6 Output error of network2

A simulink model of neural network 2 is developed to predict the breakout is shown in Fig. 7. This simulink model can be fed with different inputs and should be able to predict any break out (i.e. presence or absence of primary crack on semi-solidified slab during casting). The inputs for this model are the set of mould thermocouple temperatures, casting speed and mould level.

The simulink model can be tested with various inputs. Two set of inputs were applied to test the simulink model. First set of input is {0.13287, 0.7826, 0.46341, 0.70182, 0.1796, 0.8571, 0.9204, 0.7755, 0.8144, and 0.8841} and corresponding output is 1 as shown in Fig. 8. When there is a breakout on the slab, the output neural network 2 is 1. Second set of input is {0.23776, 0.8587, 0.73171, 0.92727,

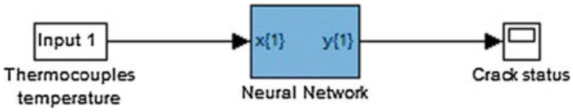


Fig. 7 Simulink model of BOPS system

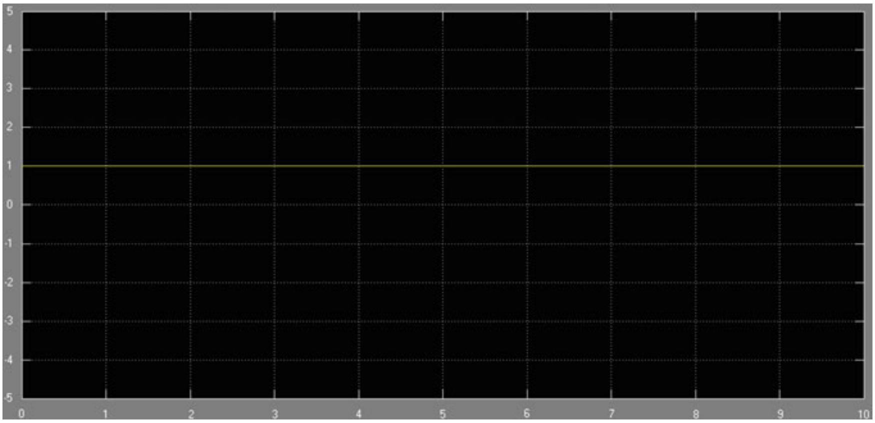


Fig. 8 In case of a crack (1)

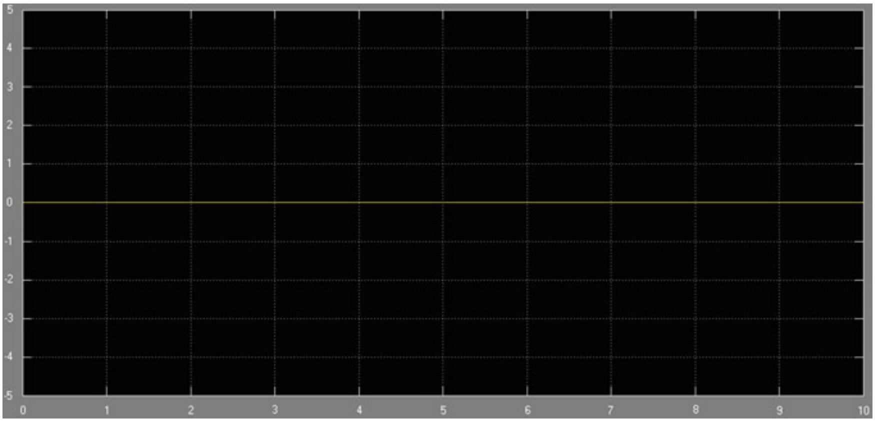


Fig. 9 In case of a no crack (0)

0.1737, 0.9643, 0.6106, 0.7347, 0.9588 and 0.9034} and corresponding output is 0 as shown in Fig. 9. Absence of any breakout on the slab output is 0.

5 Conclusion

In BSL, logic based BOPS system is used to predict breakout during casting, this system generates false alarm and many a times even fails to generate an alarm before breakout. Artificial intelligent (mainly neural network) can be used effectively to predict breakout. Previous works considered only mould thermocouples temperature as an input of neural network. But this paper considered three parameters, i.e. temperatures of mould, casting speed, and mould level as input for the neural network. It has also been shown that such neural network with three input parameter yield better results in comparison with neural network based with only mould thermocouples temperature as the input. Hence, to accurately predict breakouts, neural network based prediction system with temperatures of mould, casting speed, and mould level as input parameters can be used.

References

1. Mazumdar, S., Ray, S.: Solidification control in continuous casting of steel. Sadhana **26**(1), 179–198 (2001)
2. Raja, B.V.R.: Breakout in continuous casting of steel. Steel world monthly journal report, pp. 24–30 (2009)
3. Qing, T., Xue, J.S.: Research for breakout prediction system based on support vector regression. In: IEEE Symposium on Robotics and Applications (ISRA) (2012)
4. Xu, Q., ChaofuZhu, L.Z.: Study of the forecasting of molten steel breakouts based on the frictional force between mould and slab shell. In: International Conference on Mechanic Automation and Control Engineering, pp. 2593–2596 (2010)
5. Lameck, M., Lungile, N., Samson, M.: Neural network breakout prediction model for continuous casting. Int. J. Eng. Adv. Technol. (IJEAT) **2**(2), 2249–8958 (2014)
6. Ji, C., Cai, Z., Tao, N., Yang, J., Zhu, M.: Molten steel breakout prediction based on genetic algorithm and BP neural network in continuous casting process. In: Proceedings of the 31st Chinese Control Conference, Hefei, China (2012)
7. Zhang, B., Li, Q., Wang, G., Gao, Y.: Breakout prediction based on BP neural network of LM algorithm in continuous casting process. In: International Conference on Measuring Technology and Mechatronics Automation. IEEE (2010)
8. Tingzhi, R., Xueliang, S., Dongmei, L., Xin, J., Weiqiang, S.: Research on breakout prediction system based on multilevel neural network. In: International Conference on Electrical and Control Engineering. IEEE (2010)
9. Wentao, L., Yang, L., Yuan, Z.: Study of mould breakout prediction technique in continuous casting production. In: 3rd International Conference on Biomedical Engineering and Informatics. IEEE (2010)
10. Xu, Z., Ma, L., Zhang, Y.: Study on neural network breakout prediction system based on temperature unit input. In: International conference on measuring technology and mechatronics automation (2010)
11. Adamy, J.: Breakout prediction for continuous casting by fuzzy meal automation. In: Proceeding of the 3rd European congress of Intelligent Technique and Soft Computing (EUFIT), Aachen, Germany (1995)
12. Li, Y., Wang, Z., Ao, Z.: Optimization for breakout prediction system of BP neural network. Control Decis. **25**(3), 453–456 (2010)

13. Xu, Z., Ma, L., Zhang, Y.: Study on neural network breakout prediction system based on temperature unit input. In: International Conference on Measuring Technology and Mechatronics Automation, pp. 612–615 (2010)
14. Li, W., Li, Y., Zhang, Y.: Study of mould breakout prediction technique in continuous casting production. In: 3rd International Conference on Biomedical Engineering and Informatics, pp. 2966–2970 (2010)
15. Bellomo, P., Palchetti, M., Santa, M.E.: Neural network utilization for breakout monitoring. In: Steelmaking Conference Proceedings, pp. 345–348 (1995)
16. Hu, Z., Bi, X., Chen, C.: Study on the application of breakout pattern recognition by BP network. J. Wuhan Univ. Sci. Technol. (Natural Science Edition) **23**(2), 121–124 (2000)

Software Reliability Prediction Based on Radial Basis Function Neural Network

Pravas Ranjan Bal and Durga Prasad Mohapatra

Abstract Reliability of software is the key factor of software quality estimation during the testing period of software. This paper proposes a nonparametric method using radial basis function neural network for predicting software reliability. The Bayesian Regularization method is applied in the proposed model to improve the generalization and to avoid the over fitting problem. The proposed scheme has been tested on five benchmark datasets. The results of the system are compared with other states of the traditional method and it is observed that the proposed architecture outperforms its competent systems. The results of the proposed architecture have been adequately tested with a single feed-forward neural network model and a linear parametric software reliability growth model. The experimental result shows that the radial basis function of neural network yields better performance than the traditional software reliability growth model (SRGM).

Keywords Radial basis function · Software reliability · Feed forward neural network · Software reliability growth model

1 Introduction

The reliability of a software, as defined by ANSI, indicates that the probability of software failure free operation for a specified period in a specified environment [1, 2]. The prediction of software reliability is the task in which we can predict the future failure data and its performance using some past failure software reliability datasets of many software industries. In the last three decades, most of the software reliability growth models have been developed to predict the software reliability

P.R. Bal (✉) · D.P. Mohapatra
Department of Computer Science & Engineering, National Institute of Technology,
Rourkela 769008, India
e-mail: pravasranjan90@gmail.com

D.P. Mohapatra
e-mail: durga@nitrkl.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_10

and its cost and time factors. They are broadly categorized into two types: such as non-parametric models and parametric models. Some parametric models are popularly used in some practical engineering of software reliability and these models are based on the non-homogeneous Poisson process (NHPP) [3, 4]. Finally, it has been proven that all models cannot predict accurately for all cases [5]. Most of the non-parametric models such as different types of neural network (such as feed-forward and recurrent neural network), probabilistic neural network (PNN), support vector machine (SVM) and self-organizing map (SOM) can predict the reliability of the software metrics like failure rate, next time to failure, detection of cumulative failures, etc. These models are specially based on statistical models. Finally, we conclude that the non-parametric architectural models have better prediction capability than other types of the parametric models [6–9].

In this work, we propose a non-parametric model of software reliability prediction using radial basis function neural network which can accurately predict software reliability than other feed forward neural network and another Software Reliability Growth Model (SRGM) [10, 11].

The rest of this paper has been organized in the following manner. Section 2 briefly describes some related work for prediction of software reliability of different datasets. Section 3 briefly describes the proposed model for software reliability prediction. Section 4 presents the experimental results and finally, Sect. 5 concludes the paper.

2 Literature Survey

In this section, we briefly discuss the related works using different types of neural networks for prediction of software reliability.

Karunanithi et al. [8, 9] first introduced different types of neural networks such as feed-forward and recurrent for prediction of software reliability of some standard datasets. They had applied some feed forward neural network (FFNN) and recurrent neural network (RNN) along with elman neural network (ENN) and jordan neural network (JNN) are used to predict the cumulative number of software detected faults using execution time of software as input. They observed that the neural network could construct many models by varying its performance and complexity for different types of datasets.

Sitte [12] suggested two types of prediction methods for software reliability such as neural networks and its recalibration of some parametric architectural models. These two methods were compared using some standard datasets and measurement of the software reliability. He observed that the neural networks gave better prediction than other models.

Ho et al. [13] have used a modified Elman Recurrent Neural Network (ERNN) model to predict the number of software failures of standard datasets for prediction of software reliability. They observed that the effectiveness of different weights in proposed architecture. They also presented a comparative study of the proposed model with other existing models namely jordan recurrent neural network (JRNN), feed forward neural network (FFNN) model and some other parametric models.

Zhang et al. [14] studied that the effectiveness of back-propagation neural network (BPNN) model for prediction of software reliability. They have predicted the next failure time by considering multiple recent failure time as the input. Finally, they concluded that the effectiveness of the neural network model and its performance depends on the nature of these standard datasets.

Tian and Noore [15, 16] used an evolutionary neural network model for prediction of software reliability using multiple input single output architecture. They have also used a genetic algorithm to optimize the input nodes and the hidden nodes of the network and observed that whenever new failure data comes, the neural architecture model is configured dynamically.

Su et al. [7] proposed a Dynamically Weighted Combinational Model (DWCM) for prediction of software reliability using neural network. They have used different types of activation functions in the hidden layer of the proposed model and noticed that the proposed architecture provides a better prediction than other traditional NHPP models.

Literature survey reveals that most of the network models used feed forward and recurrent neural network to predict the software reliability of some standard datasets. However, use of radial basis function is limited. So, this work focuses only on radial basis function with a goal to obtain better prediction.

3 Proposed Work

In this section, we have presented the block diagram of the proposed model for prediction of software reliability data based on the radial basis function neural network. The overall block diagram of this model is shown in Fig. 1.

Dataset. We have used five types of datasets Data1, Data2, Data3, Data4 and Data5 [1] for our model. All datasets are collected from real-time command and control application, flight dynamic application, flight dynamic application, flight dynamic application and single user workstation, respectively.

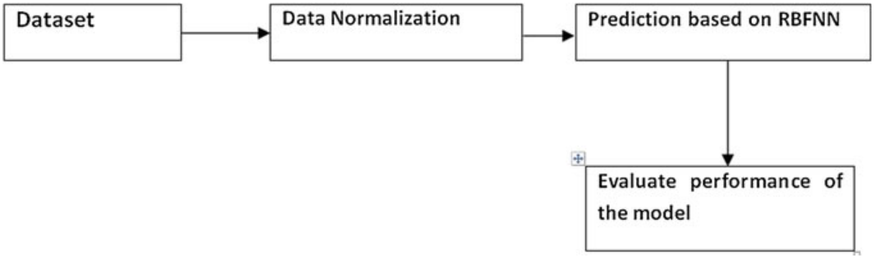


Fig. 1 Block diagram of RBFNN model

Data Normalization. We have normalized all datasets in the range between [min, max] (e.g. [0, 1]) using min-max formula is given by the equation as follows.

$$\text{Normalized } (t_i) = (t_i - \min_A)\,((\max - \min)\,/(\max_A - \min_A)) + \min \qquad (1)$$

Prediction Based on RBFNN. RBFNN is used to predict the reliability of software data based on the radial basis function as activation function in the hidden layer of neural network.

Evaluate Performance of the Model. We have used two types of errors like MSEs and REs for performance comparison of the proposed model with other models.

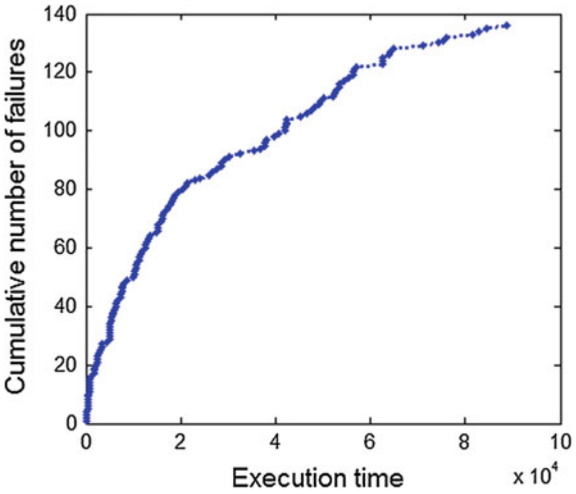
3.1 Reliability of Software Data

The reliability of software data are organized in pair $\{t_i, N_i\}$, where t_i is the execution time used for input of the proposed model and N_i is the cumulative number of failures used as the output of the proposed model. The software reliability data of Data1 [1] are depicted in Fig. 2.

3.2 RBFNN Model

The prediction system based on the radial basis function neural network (RBFNN) is shown in Fig. 3. RBFNN model is three layer architecture such as an input layer, a hidden layer consists of radial basis function as transfer function and an output

Fig. 2 Example of software reliability data



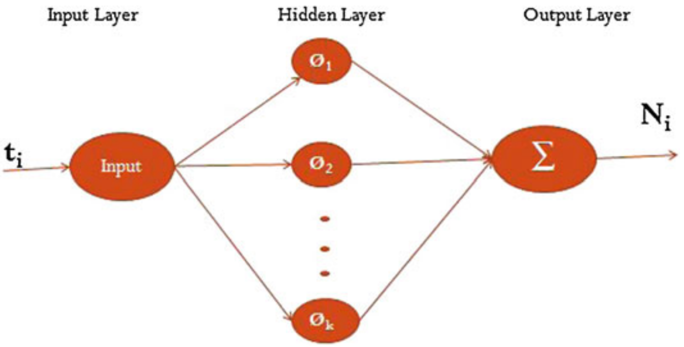


Fig. 3 The architecture of RBFNN model

layer. This architecture also follows single-input and single-output system. We have used the activation function as radial basis function in the hidden layer. The hidden layer consists of **k** neurons. The execution time of software $\mathbf{t_i}$ is supplied as input to the model and the number of software failures $\mathbf{N_i}$ for prediction is produced the output of our proposed model.

3.3 Derivation of Radial Basis Function

We have chosen the activation function based on radial basis function as follows. The input of the radial basis function can be designed as the real numbers $t \in R^n$ and the output of radial basis function network can be conceived as $f(x) \in R^n \rightarrow R$ is given as follows

$$f(x) = \sum_{i=1}^{k} w_i \, \varnothing(x - c_i) \tag{2}$$

where, k is the number of neurons, x is the number of inputs, w_i is the weights of neuron i and c_i is the centroid for neuron i.

The function $\phi(x - c_i)$ is calculated by the Euclidian distance as given by

$$\phi\,(x - c_i) = \sqrt{\sum\nolimits_{\mathrm{i}=1}^{\mathrm{k}} (x - c_i)^2} \tag{3}$$

We have used the Bayesian Regularization method to improve generalization and avoid the over fitting problem [17] of the network. The error is defined as follows

$$\text{Error} = \alpha \ \text{MSE} + (1 - \alpha) \ \text{MSW} \tag{4}$$

where, α is the performance ratio, MSW is the mean square weights and MSE is the mean square error.

3.4 Performance Measures

We need some meaningful performance measures to compare the software reliability prediction. Here, the proposed model is trained with some part of data and the rest of the failure data is applied for testing purpose. In this approach, we have used two types of errors in performance measurement. The relative error (RE) and Mean Square Error (MSE) are defined as follows

$$\mathrm{RE} = \frac{(\hat{y}_i - y_i)}{y_i} \times 100 \tag{5}$$

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{6}$$

where, n is the total number of data samples, $\hat{y}_i$ is the predicted value and y_i is the actual value.

4 Experimental Results and Comparisons

The proposed model is simulated in MATLAB 2014a. We have used five types of datasets Data1, Data2, Data3, Data4 and Data5 [1] for software reliability prediction in this experiments. The dataset Data1 is collected from real-time command and control application with 21,700 numbers of assembly instructions and 136 numbers of failures. The dataset Data2 is collected from flight dynamic application with 10,000 assembly instructions and 118 numbers of failures. The dataset Data3 is collected from flight dynamic application with 22,500 assembly instructions and 180 numbers of failures. The dataset Data4 is collected from flight dynamic application with 38,500 assembly instructions and 213 numbers of failures. The dataset Data5 is collected from a single user workstation with 397 failures. These datasets contain two attributes such as execution time and the cumulative number of failures and we normalized these execution times and the cumulative number of failures data of each dataset in the range [0, 1].

In our experiments, this model is trained with 50 % of the data for dataset Data1, 66 % of the data for dataset Data2, 50 % of the data for dataset Data3, 55 % of the data for dataset Data4 and 65 % of the data for dataset Data5. The remaining data for each dataset is used for testing. We have used different training ratios of different datasets for better prediction results.

We compare our proposed model with feed forward neural network (FFNN) and another software reliability growth models (SRGM) [18, 19] called Duane model: The Prediction Model is given by $\mu(t) = a\ t^b\ a > 0,\ b > 0$, where a is the size parameter of the curve and b is the shape of the growth curve.

4.1 Performance Measures

We have chosen 14 neurons for RBFNN model, whereas for FFNN model 10 neurons in the hidden layer. The prediction results of RBFNN model and relative errors of different model on five datasets Data1, Data2, Data3, Data4 and Data5 are depicted in Figs. 4, 5, 6, 7 and 8, respectively. The mean square error of different models on datasets Data1, Data2, Data3, Data4 and Data5 are shown in Table 1. We observed that RBFNN model shows lower prediction errors than FFNN model and another parametric model. For all datasets parametric model such as Duane model shows worst performance than others. From the above results and discussion, it is observed that the RBFNN is superior to the other models.

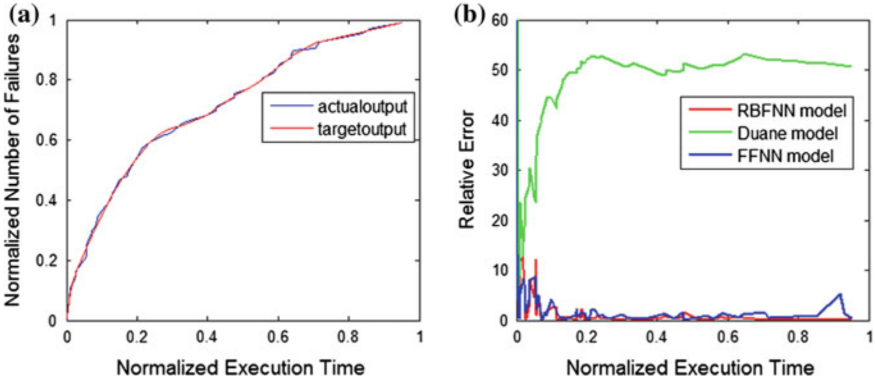


Fig. 4 Prediction results of RBFNN model and relative error of different models of Data1 in (**a**) and (**b**)

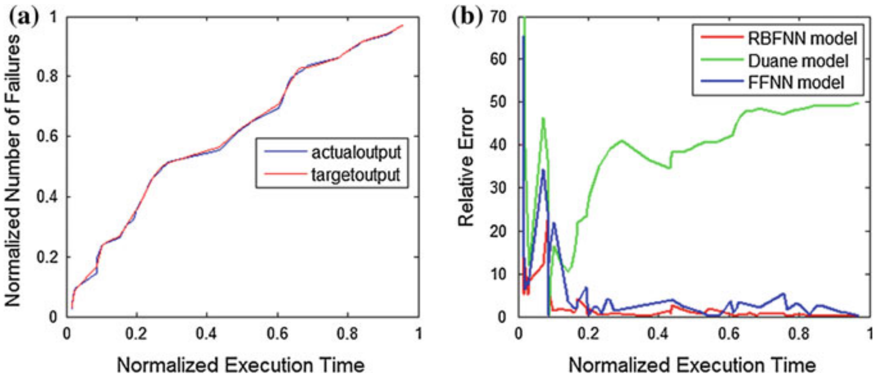


Fig. 5 Prediction results of RBFNN model and relative error of different models of Data2 in (**a**) and (**b**)

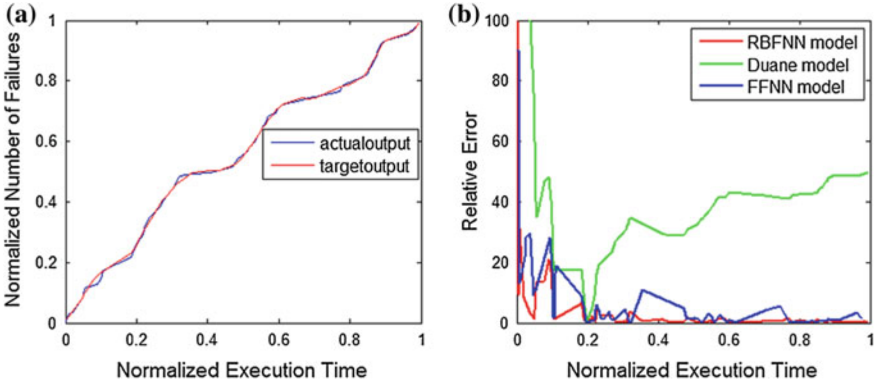


Fig. 6 Prediction results of RBFNN model and relative error of different models of Data3 in (**a**) and (**b**)

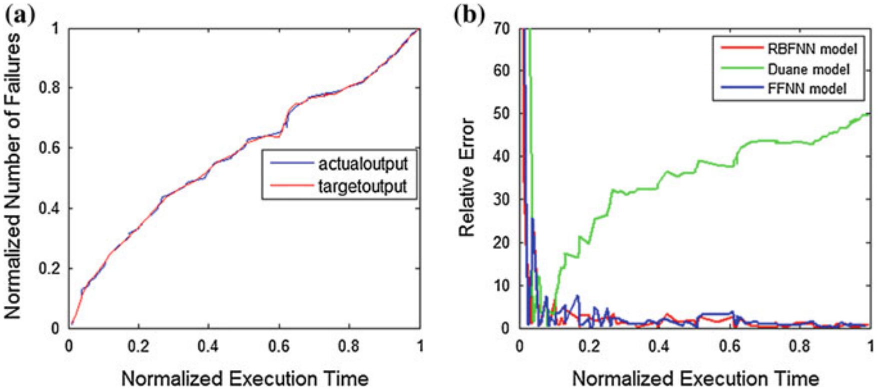


Fig. 7 Prediction results of RBFNN model and relative error of different models of Data4 in (**a**) and (**b**)

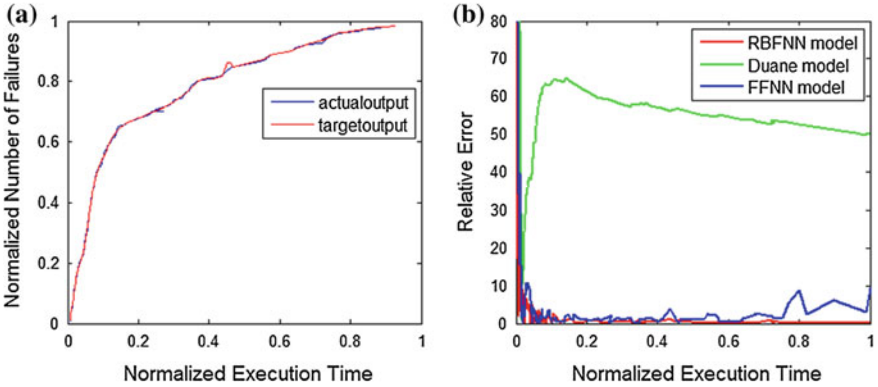


Fig. 8 Prediction results of RBFNN model and relative error of different models of Data5 in (**a**) and (**b**)

Table 1 Mean square error for different models

Dataset	Mean square error		
	RBFNN	FFNN	Duane model
Data1	1.1257	1.1851	48.2112
Data2	1.8549	4.1169	31.5692
Data3	1.8549	4.9761	20.1511
Data4	1.5170	1.6551	31.5112
Data5	1.2974	2.9521	51.6780

5 Conclusion

In this paper, we have applied the non-parametric approaches such as neural network methods based on the radial basis function for software reliability prediction. The proposed radial basis function neural network (RBFNN) shows better prediction than other neural network architecture and another traditional parametric model. The experimental results show that the proposed model proves the lower prediction error than other artificial neural networks and linear software reliability growth model (SRGM).

References

1. Lyu, M.R.: Handbook of Software Reliability Engineering. McGraw-Hill, New York (1996)
2. Musa, J.D.: Software reliability engineering. More Reliable Software, Faster Development and Testing. McGraw-Hill, New York (2004)
3. Malaiya, Y.K., Li, M.N., Bieman, J.M., Karcich, R.: Software reliability growth with test coverage. IEEE Trans. Reliab. **51**, 420–426 (2002)
4. Pham, H., Nordmann, L., Zhang, X.M.: A general imperfect software-debugging model with S-shaped fault detection rate. IEEE Trans. Reliab. **48**(2), 169–175 (1999)
5. Li, S.M., Yin, Q., Guo, P., Lyu, M.R.: A hierarchical mixture model for software reliability prediction. Appl. Math. Comput. **185**, 1120–1130 (2007)
6. Zheng, J.: Predicting software reliability with neural network ensembles. Expert Syst. Appl. **36**(2), 2116–2122 (2009)
7. Su, Y.S., Huang, C.Y.: Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. J. Syst. Softw. **80**, 606–615 (2007)
8. Karunanithi, N., Whitley, D., Malaiya, Y.K.: Using neural networks in reliability prediction. IEEE Softw. **9**, 53–59 (1992)
9. Karunanithi, N., Whitley, D., Malaiya, Y.K.: Prediction of software reliability using connectionist models. IEEE Trans. Softw. Eng. **18**, 563–574 (1992)
10. Orr, Mark, J.L.: Introduction to radial basis function networks. Technical Report, center for cognitive science, University of Edinburgh (1996)
11. Leonard, J.A., Kramer, M.A., Ungar, L.H.: Using radial basis functions to approximate a function and its error bounds. IEEE Trans. Neural Netw./Publ. IEEE Neural Netw. Council **3** (4), 624–627 (1991)
12. Sitte, R.: Comparison of software-reliability-growth predictions: neural networks vs. parametric recalibration. IEEE Trans. Reliab. **48**(3), 285–291 (1999)

13. Ho, S.L., Xie, M., Goh, T.N.: A study of the connectionist models for software reliability prediction. Comput. Math Appl. **46**, 1037–1045 (2003)
14. Cai, K.Y., Cai, L., Wang, W.D., Yu, Z.Y., Zhang, D.: On the neural network approach in software reliability modeling. J. Syst. Softw. **58**, 47–62 (2001)
15. Tian, L., Noore, A.: On-line prediction of software reliability using an evolutionary connectionist model. J. Syst. Softw. **77**, 173–180 (2005)
16. Tian, L., Noore, A.: Evolutionary neural network modeling for software cumulative failure time prediction. Reliab. Eng. Syst. Safety **87**, 45–51 (2005)
17. Foresee, F.D., Hagan, M.T.: Gauss–Newton approximation to Bayesian learning. In: Proceedings of the 1997 IEEE International Conference on Neural Networks, Houston, TX, pp. 1930–1935 (1997)
18. Smith, Stephen A., Shmuel, S.O.: Reliability growth of repairable systems. Naval Res. Logist. Q. **27**(4), 539–547 (1980)
19. Shanmugam, L., Florence, L.: A comparison of parameter best estimation method for software reliability models. Int. J. Softw. Eng. Appl. **3**(5), 91–102 (2012)

Part III
Genetic Algorithm and Bio-inspired Computing

Design of Two-Loop PID Controller for Inverted Cart-Pendulum System Using Modified Genetic Algorithm

D. Sain, S.K. Swain and S.K. Mishra

Abstract This study focuses on the design of controllers for inverted cart-pendulum system. This cart-pendulum system, a nonlinear one, has been linearized around the equilibrium point to obtain linearized model transfer function. In this paper, the design of two-loop proportional integral derivative (PID) controller that gives more flexibility to control the inverted cart-pendulum system has been considered and the controller parameters have been optimized through modified genetic algorithm (GA). Furthermore, other methods such as LQR and LQG have been applied to verify closed loop response of the system for unit step signal. A performance comparison between the above mentioned techniques is investigated and the analysis shows superiority of two-loop PID controller in terms of both overshoot and settling time as compared to other mentioned methods. A comprehensive analysis of robustness to model uncertainties will be incorporated in the future work.

Keywords Inverted cart-pendulum · Genetic algorithm · Two-loop PID · LQR and LQG

1 Introduction

Inverted cart-pendulum is an example of inherently nonlinear, unstable, multivariable and non-minimum phase system. It has always been a challenging task for the engineers to develop a proper control strategy to control this system. Many researchers have proposed different control techniques which include use of classical methods to modern methods, such as artificial neural network (ANN), fuzzy logic, genetic algorithm (GA), and swarm intelligence. [1–4]. In last few decades,

The first author would like to thank AICTE, Govt. of India for the financial support.

D. Sain (✉) · S.K. Swain · S.K. Mishra
Department of EEE, BIT Mesra, Ranchi, India
e-mail: hiidebdoot@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_11

the use of these modern methods has gained immense popularity in different field of research because of the ability to provide satisfactory and accurate result [5, 6].

The aim of this paper is to design a two-loop PID controller for cart-pendulum system so that it moves the cart to a desired position and pendulum stabilizes in an upright equilibrium position [7, 8]. In this paper, modified genetic algorithm has been applied to optimize the controller parameters.

This paper is organized in eight sections. Section 1 gives the introduction of the paper. In Sect. 2 brief overview, specifications, free body diagram, mathematical model and state space representation of cart-pendulum system is provided. Sections 3 and 4 deals with dominant pole calculation and the structure of plant with two-loop PID controller. In Sect. 5, optimization of objective function through modified GA has been provided. Section 6 is about the simulation diagram and response of the system with two-loop PID, LQR and LQG control. Performance comparison between different methods and conclusion are provided in Sects. 7 and 8 respectively.

2 Cart-Pendulum System

The schematic diagram of cart-pendulum system considered in this paper is provided in Fig. 1.

The system consists of an inverted pendulum mounted on a motorized cart. In this paper, a two dimensional problem is considered where the pendulum is allowed to move in the vertical plane with the presence of the control force 'F'. This force works as the input to this system and it moves the cart horizontally. The outputs of the systems are the horizontal position of the cart 'x' and the angular position of the pendulum 'θ'.

The parameters of the cart pendulum system [9] are taken as follows:

M—mass of the cart (2.4 kg), m—mass of the pendulum (0.23 kg), b—coefficient of cart friction (0.055 Ns/m), L—length of the pendulum (0.4 m), J—moment of inertia of pendulum (0.099 kg m^2), g—gravitational constant (9.81 m/s^2),

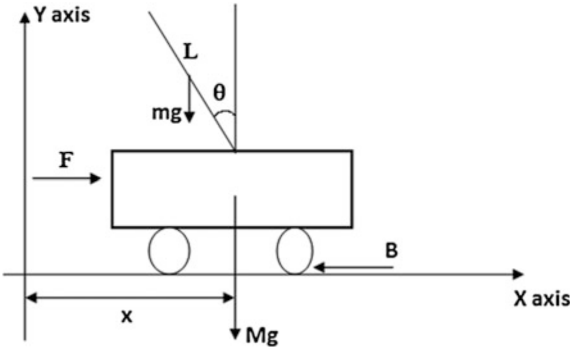


Fig. 1 Cart-pendulum system

F—applied force to cart ±24 N, x—position of the cart from reference (±0.3 m), θ—angle of pendulum with respect to vertical (≤ 0.1 rad).

By applying the laws of dynamics on cart-pendulum system, the equations of motions [9] are obtained as:

$$\dot{\theta} = mL/\sigma\{[F - b\dot{x}]\cos\theta - mL(\dot{\theta})^2\cos\theta\ \sin\theta + (m+M)g\,sin\theta\} \tag{1}$$

$$\ddot{x} = 1/\sigma\{\left(J + mL^2\right)\left[F - b\dot{x} - mL\dot{\theta}^2\sin\theta\right] + mL^2g\ \sin\theta\ \cos\theta\} \tag{2}$$

where

$$\sigma = mL^2\left(M + mcos^2\theta\right) + J(M+m) \tag{3}$$

Linearizing Eqs. (1) and (2) for a small angle θ from the vertical equilibrium point and by neglecting friction coefficient b which is very small, the state space model is obtained as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0.2381 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 6.8073 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.3894 \\ 0 \\ 0.2638 \end{bmatrix} F, \mathrm{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{4}$$

where $x, \dot{x}, \theta, \theta$ are the states and y is the output vector. A DC motor is used to convert control voltage U to force F and represented by a single block with gain taken as unity, i.e., $U = F$. For $U = F$, from this state space representation, the transfer function can be derived as [9]:

$$\frac{X(s)}{F(s)} = \frac{X(s)}{U(s)} = P1 = \frac{0.3894}{s^2}, \frac{\theta(s)}{F(s)} = \frac{\theta(s)}{U(s)} = P2 = \frac{0.2638}{s^2 - 6.807} \tag{5}$$

3 Dominant Pole Calculation

The design specifications for this study have been taken as

$$\text{Maximum overshoot}\left(M_p\right) \leq 8\%\ \text{and Settling time}\ t_s \leq 3\ \text{s}$$

According to these specifications, dominant poles have been found to be at $s_{1,2} = -1.33335 \pm 1.65845i$.

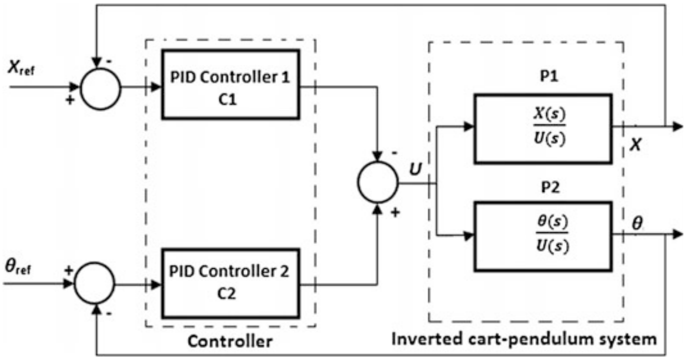


Fig. 2 Cart-Pendulum system with two-loop PID controller

4 Cart-Pendulum System with Two-Loop PID Controller

The cart-pendulum system with two-loop PID Controller is shown in the following Fig. 2

The characteristics equation for the system given in Fig. 2 becomes

$$1 - P_1C_1 + P_2C_2 = 0 \tag{6}$$

And the PID controllers are represented by

$$C_1 = k_{p1} + \frac{k_{i1}}{s} + k_{d1}s\,,\; C_2 = k_{p2} + \frac{k_{i2}}{s} + k_{d2}s \tag{7}$$

After substituting $s = -1.33335 + j1.65845$ in Eq. (6) and separating the real (R) and imaginary (I) part, R and I will be

$$R = 1 + 0.0184k_{p1} - 0.036k_{i1} + 0.1145k_{d1} - 0.0256k_{p2} + 0.0127k_{i2} + 0.01k_{d2} \tag{8}$$

$$I = -0.0839k_{p1} + 0.0179k_{i1} + 0.1424k_{d1} + 0.0145k_{p2} + 0.005k_{i2} - 0.0617k_{d2} \tag{9}$$

where k_{p1}, k_{i1} and k_{d1} denote the coefficient for the proportional, integral and derivative terms, respectively, for Controller 1 (C1) and k_{p2}, k_{i2} and k_{d2} denote the same for Controller 2 (C2).

The objective function 'f' considered for obtaining the value of k_{p1}, k_{i1}, k_{d1}, k_{p2}, k_{i2} and k_{d2} has the following format:

$$f = |R| + |I| + |\theta|, \text{where}\, \theta = \tan^{-1}\left(\frac{|I|}{|R|}\right) \tag{10}$$

5 Objective Function Optimization Using Modified GA

In this paper, we have proposed modified genetic algorithm (GA) where best solution in each iteration does not take part in the process of crossover and mutation, rather its cloned one is used for that purpose. In the next stage, best solution along with off springs created is taken into consideration. It could be useful for optimizing those multimodal functions where there are number of maxima and minima having close value, due to which there is a chance of the solution to be trapped in local optima while solving directly through normal GA. The specifications taken for writing MATLAB code are as follows:

Population size, bit size, crossover probability, mutation probability and number of iteration are taken as 100, 10, 0.8, 0.125 and 25, respectively.

The objective function considered over here has six unknowns, i.e., k_{p1}, k_{i1}, k_{d1}, k_{p2}, k_{i2} and k_{d2}. The range of these parameters taken for writing the code is: $3 \leq k_{p1} \leq 5, 0 \leq k_{i1} \leq 0.002, 2 \leq k_{d1} \leq 4, 220 \leq k_{p2} \leq 222, 120 \leq k_{i2} \leq 122$ and $59 \leq k_{d2} \leq 61$, respectively, and it has been decided after a number of trial runs.

After optimizing the objective function through modified GA one obtains the values of $k_{p1}, k_{i1}, k_{d1}, k_{p2}, k_{i2}$and k_{d2} as 4.5213, 0.0012, 2.5712, 220.3164, 120.9867 and 60.7956, respectively.

It should be noted that the objective function value for these set of parametric values is not exactly zero as expected because of approximate linearization of nonlinear system around the equilibrium point.

6 Simulation Diagram and Result

6.1 With Two-Loop PID Controller

The simulation diagram for the cart-pendulum system with two-loop PID is shown in Fig. 3.

Unit step signal has been taken as reference signal for this study and the closed loop simulation is carried out for 100 s. The output response in terms of position and angle is shown in Fig. 4.

6.2 With LQR Control

For LQR design,

$$u = r - kx, \dot{x} = Ax + Bu, y = Cx + Du \tag{11}$$

The gain matrix ‘*k*’ has been identified by optimizing the cost function

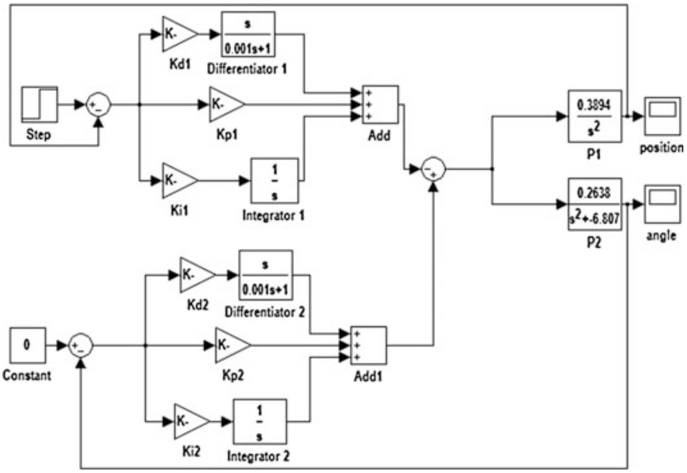
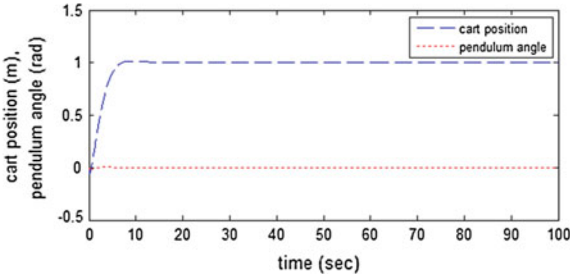


Fig. 3 Cart-pendulum system with two-loop PID simulation diagram

Fig. 4 Position of cart and angle of pendulum with two-loop PID control



$$J(u) = \int_{0}^{\infty} (x^T Q x + u^T R u + 2x^T N u)\mathrm{d}t \tag{12}$$

where Q is the semi positive definite, R is the positive definite and N is the state reference gain matrix and all the notations are conventional.

The state feedback gain matrix 'k' required for LQR control is given by

$$k = [\,-1 \quad -3.0620 \quad 71.7899 \quad 27.6065\,]$$

The simulation diagram for the cart-pendulum system with LQR control is shown in Fig. 5.

The output response of LQR control in terms of position and angle is shown in Fig. 6.

Fig. 5 Cart-pendulum system with LQR control simulation diagram

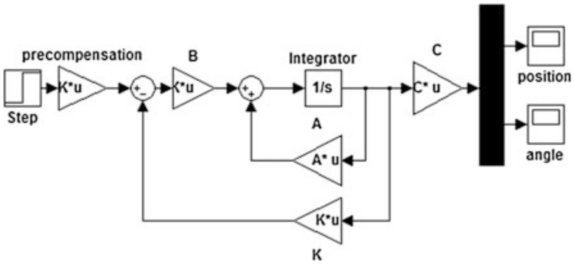
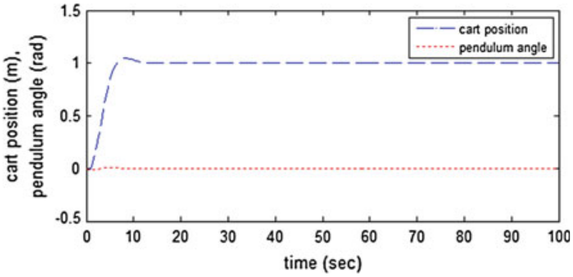


Fig. 6 Position of cart and angle of pendulum with LQR control



6.3 With LQG Control

For LQG design

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}),\ y = Cx + Du \tag{13}$$

The cost function that to be minimized in LQG control is given by

$$J = E\left\{ \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} [x^T, u^T] Q_{xn} \begin{bmatrix} x \\ u \end{bmatrix} dt \right\} \tag{14}$$

The simulation diagram for the cart-pendulum system with LQG control is shown in Fig. 7.

The matrix L required for LQG control is mathematically represented as

$$L = \begin{bmatrix} 42.8514 & 1.0395 \\ 458.3565 & 22.6295 \\ 0.9384 & 43.1486 \\ 20.1527 & 471.4483 \end{bmatrix}$$

The output response of LQG control in terms of position and angle is shown in Fig. 8.

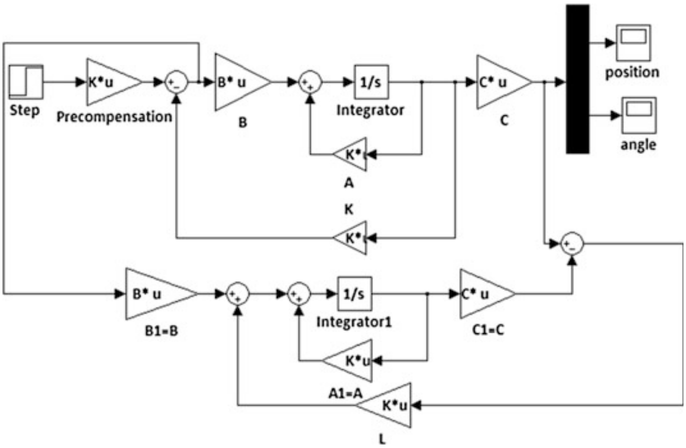
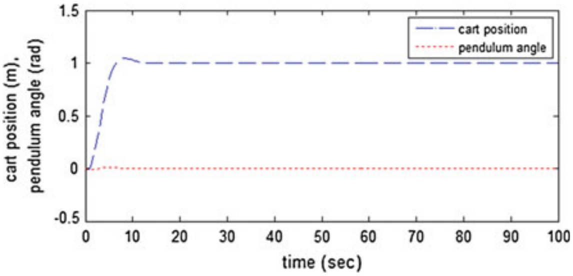


Fig. 7 Cart-pendulum system with LQG control simulation diagram

Fig. 8 Position of cart and angle of pendulum with LQG control



In this study, pre-compensation technique has been used for both LQR and LQG and it has been shown as a gain block after the reference step input. The value of the gain is taken as −1 and has been confirmed from MATLAB code written for the LQR and LQG control of cart-pendulum system.

7 Performance Comparisons Between Two-Loop, LQR and LQG Control

The performance of different control techniques is provided in the Table 1 which clearly indicates the superiority of two-loop PID controller as compared to LQR and LQG control.

Table 1 Performance comparison between different control techniques

Control technique	Overshoot (M_p) (%)	Settling time (t_s) (s)
Two-loop PID	1.34	6.35
LQR	4.25	10.4
LQG	4.26	10.5

8 Conclusion

In this study, two-loop PID, LQR and LQG controllers have been designed for the cart-pendulum system and the performance of these controllers has been compared. It has been found that two-loop PID controller along with modified GA provides better response than LQR and LQG controller in terms of both overshoot and settling time. For improving time domain response and robustness, this particular approach of using two-loop PID along with proposed modified GA can be extended to the other class of MIMO plants. A comprehensive analysis of robustness to model uncertainties for inverted cart-pendulum system will be incorporated in the future work.

References

1. Chaturvedi, D.K., Qamar, T., Malik, O.P.: Quantum inspired GA based neural control of inverted pendulum. Int. J. Comput. Appl. **122**(23), 46–52 (2015)
2. Malik, S., Mishra, S.K.: Optimal design of a fuzzy PID controller for inverted pendulum system. In: International Conference on Communication, Computing and Power Technologies, 22–23, April. Chennai, India. Springer (2015)
3. Moghaddas, M., Dastranj, M. R., Changizi. N., Khoori, N.: Design of optimal PID controller for inverted pendulum using genetic algorithm. Int. J. Innov. Manag. Technol. **3**(4), 440–442 (2012)
4. Jain, N., Gupta, R., Parmar, G.: Intelligent controlling of an inverted pendulum using PSO-PID controller. Int. J. Eng. Res. Technol. **2**(12), 3712–3716 (2013)
5. Mishra, S.K., Panda, G., Majhi, R.: A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. Swarm Evol. Comput. Elsevier **16**, 38–51 (2014)
6. Gaurav, K., Sahoo, A., Mishra, S.K.: Nonlinear system identification using functional link multilayer perceptron artificial neural network. Int. J. Appl. Eng. Res. **10**(44), 31542–31546 (2015). ISSN 0973-4562
7. Ogata, K.: Modern Control Engineering, 4th edn. Pearson Education (Singapore) Pvt. Ltd., New Delhi, Chapter 12 (2005)
8. Mandal, A.K.: Introduction to Control Engineering. New Age International Pub., New Delhi, Chapter 13, (2000)
9. Ghosh, A., Krishnan, T.R., Subudhi, B.: Robust proportional-integral-derivative compensation of an inverted cart-pendulum system: an experimental study. IET Control Theory Appl. **6**(8), 1145–1152 (2012)

An Improved Heuristic K-Means Clustering Method Using Genetic Algorithm Based Initialization

D. Mustafi, G. Sahoo and A. Mustafi

Abstract In this paper, we propose methods to remove the drawbacks that commonly afflict the k-means clustering algorithm. We use nature based heuristics to improve the clustering performance offered by the k-means algorithm and also ensure the creation of the requisite number of clusters. The use of GA is found to be adequate in this case to provide a good initialization to the algorithm, and this is followed by a differential evolution based heuristic to ensure that the requisite number of clusters is created without minimal increase in the running time of the algorithm.

Keywords K-means · GA · Precomputation · DE heuristic

1 Introduction

Over the last few decades many algorithms have been developed inspired by natural processes and phenomena to solve optimization problems. These algorithms have been shown to be extremely successful in solving a wide variety of problems [1]. They have met with extensive success in different fields of real world applications related to computer science, agriculture, industry, medicines, engineering, etc. A survey of such algorithms demonstrates the influence of a wide variety of real life processes. Simulated annealing (SA) [2] inspired by the method of annealing steel is one of the most popular optimization algorithms. Particle swarm optimization uses the characteristic of insect swarms, i.e., velocity and current position to solve the optimization problems [3]. The ant colony optimization [4] has been inspired by

D. Mustafi (✉) · G. Sahoo · A. Mustafi
Department of CSE, Birla Institute of Technology, Mesra 835215, India
e-mail: debjani.mustafi@bitmesra.ac.in

G. Sahoo
e-mail: gsahoo@bitmesra.ac.in

A. Mustafi
e-mail: abhijit@bitmesra.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_12

the characteristic nature of real ants which use their intuition to find the shortest path between their nest and a source of food and also guides other ants to this path.

Another interesting optimization algorithm is developed by the inspiration of the flashing behavior of the fire fly in nature and is popularly used as firefly algorithm (FA) [5]. One of the recently introduced algorithms, the artificial bee colony (ABC) algorithm [6] simulates the intelligent foraging behavior of the honey bee and has been found to generate good results in many problems. The bat algorithm (BA) [7] was developed by the capability of bats to prey and recognize different types of insects in complete darkness using ultrasonic waves. The gravitational search algorithm (GSA) [8] was constructed based on the Newtonian notion of gravity and mass interaction. Another heuristic algorithm, the black hole (BH) algorithm [1] is formulated inspired by the phenomenon of a black hole in space.

Clustering is a typical data mining technique which consists of grouping similar data objects together. Objects within same cluster are highly similar, whereas different clusters have highly dissimilar data [9]. Normally to measure the similarity between two data objects a distance measurement metric is used. Some of the common metrics used to measure the similarity of objects are the Euclidean distance, Manhattan method, cosine similarity (for text data), the Jacquard coefficient, etc. [10]. Using the Euclidean method the distance between any two objects can be calculated as

$$d(o_i, o_j) = \sqrt{(o_i^2 - o_j^2)} \tag{1}$$

where o_i, o_j are the *i*th and *j*th data objects.

Any crisp clustering algorithm is considered to be valid if and only if it satisfies the following conditions:

1. A data object must be assigned to only a single cluster.
2. A cluster must not be empty.
3. All data objects must be assigned a cluster.

While various clustering methods have been documented in the literature, the two primary techniques of clustering are the hierarchical and partitioning methods. Hierarchical clustering technique generates a nested sequence of partitions. It may start with a single set of data and group them into a larger set in the next higher level and vice versa. In the partition algorithm, the larger data set is partitioned into desired number of clusters at every level. Among the partitioning clustering methods, the k-means [11] is the most widely used algorithm for its simplicity and efficiency. However, the algorithm suffers from some restrictions. The number of clusters has to be pre-decided and algorithm may get stuck into the local optima. The basic k-means algorithm is presented as [12]:

```
Select K points as initial centroids
Repeat
 From K clusters by assigning each point to its closest centroid.
 Re-compute the centroid of each cluster
Until centroid do not change.
```

2 Genetic Algorithm

The nature inspired evolutionary method genetic algorithm (GA) is based on the Darwin's theory of evolution. GA is a computational abstraction of biological evolution which can solve several optimization problems. GA is influenced by the basic principles of life which uses simple genetic operations like crossover, selections and mutation to a sequence of alleles [13] which represent a chromosome. GA use a population of size *N*, where each individual in the population [14] is represented by a binary string of length *l*. The collection of the chromosomes is referred to as a population. Each chromosome [15] is associated with a fitness value. Chromosomes having the best fitness values are selected as the parents for the population in the next generation. Genetic operators like crossover [13] is performed to generate better offsprings for the next population which may improve the overall performance of the algorithm. To generate better result, mutation operation can be applied to the offsprings to escape from local optimas. The basic GA algorithm is given below [13]:

```
Initialize Population
 -Create initial population
 -Evaluate individuals in initial population
Create new population
 -Select fit individuals for reproduction
 -generate offspring with genetic operator crossover
 -mutate offspring
 -evaluate offspring
```

3 Proposed Methodology

The k-means remains an extremely popular choice to cluster data. The clustering speed offered by it and the rate of convergence offered by the algorithm makes it an extremely attractive choice to cluster objects. However, the basic k-means algorithm in its original incarnation suffers from some very grave issues [12]. The algorithm requires an initial estimate of the actual number of clusters and is prone to create empty clusters or clusters centered around outliers. This is usually due to the fact that the algorithm struggles to break away from local optimal centroids which may seem correct but are actually misleading. It has been well documented in the literature [16] that the choice of the initial centers has a strong influence on the performance of the algorithm and often decides the quality of clusters that are created. Choosing points too close to each other can lead to the creation of empty clusters as distinct clusters may collapse into each other while choosing distant points may be affected by the existence of outliers.

Various methods have been proposed in the literature [12] to handle these issues. One common approach suggested is to choose “K” random points in the data space which are as distant as possible from the center of the data space. Another approach suggests to continuously choose centers which are maximally distant from all other points which have been chosen as centroids so far. While these methods definitely improve the cluster quality of the algorithm they are extremely computationally intensive and require the computation of individual distances between many points to decide on a good choice of initial centers. It is observed that this problem provides an extremely effective case for the application of heuristic algorithms as the result sought need not to be exact but should be sufficiently close.

In this paper, we have used two different heuristics to improve the clustering performance offered by the k-means algorithm.

1. A GA based fitness function [14] is proposed to choose initial centers which cover the entire solution space, and ensure that clusters do not miss certain isolated data points.
2. A differential evolution [16] based heuristic has been proposed to ensure that no empty clusters are created, and every run of the k-means algorithm always produces the requisite number of clusters.

For the GA based initialization of centers the following fitness function has been used,

$$\text{Maximize}\, y = \sum_{i=1}^{K} \text{dist}(c_k, m) + \sum_{i=1}^{K} \sum_{\substack{j=1 \\ i \neq j}}^{K} \text{dist}(c_i, c_j), \tag{2}$$

Where “m” is the center point of the entire data space and “c_i” is the center of the “i^{th}”, cluster, and “dist” is the chosen distance metric. As can be observed, Eq. (2) tries to choose initial points which are far from the center of the data space and also far from each other. An iterative method of finding such centroids would be extremely computationally intensive, and this paper suggests the use of floating point GA to find the set of “K” such initial centroids. The chromosome used in this case is an array of size *K*no_of_features*, where “*no_of_features*” is the dimensionality of the data space. It is interesting to note that it is not required to run the GA for all iterations as we are not looking for any exact results. Any set of points that approximately meets the condition stated in Eq. (2) can be used to start the actual k-means iterations.

It was observed that while Eq. (2) guarantees a good set of initial centers in the majority of cases there were still cases where the initial centers chosen were outliers. This resulted in the creation of empty clusters as these outliers were neglected by the k-means algorithm in subsequent iterations. To overcome this shortcoming, the following heuristics have been proposed to ensure the creation of the requisite number of clusters while still maintaining a good choice of centers.

If required number of clusters == 2 but only one cluster has been created

- Choose any random point as the second cluster point and continue.

If required number of clusters == 3 but only two clusters have been created

- Choose the third center as the point obtained by the vector addition of the two centers obtained, i.e., $\mathbf{C_1 + C_2}$.

If required number of clusters > 3 and only three clusters have been created

- Choose a new cluster center as $\mathbf{C_1 + F^*(C2 - C3)}$, where C1, C2 and C3 are the three clusters with the smallest cluster densities in decreasing order, and F is a mutation factor represented by a floating point constant.

4 Justification for the Heuristic

The heuristic proposed in the previous section is based on the observation that if the requisite number of clusters have not been formed then it is best to split the existing cluster that has the largest dispersion. Based on this observation, we choose a point that is in proximity of the center of the most dispersed cluster and directed towards the next most dispersed cluster. This can be seen from the fact that we have used the DE based heuristic [16] that has **C1** as the basis point and the difference vector $\mathbf{C2 - C3}$ as the direction of exploration. If nothing else this guarantees the generation of a new centroid that is in the neighbourhood of the centroid of a large (or dispersed) cluster.

5 Results and Discussions

We demonstrate the results using the classic Iris dataset that contains 150 samples made up of three different species of the Iris flower, i.e., Setosa, Virginica and Versicolor. Each sample is made up of four numerical attributes and the distance metric used for the experiments is the Euclidean metric. All simulations have been performed using MATLAB®.

Figures 1, 2, 3 and 4 demonstrate the results of performing the basic k-means algorithm over the Iris dataset for a test set of size 100. The data was normalized for

Fig. 1 The original Iris dataset

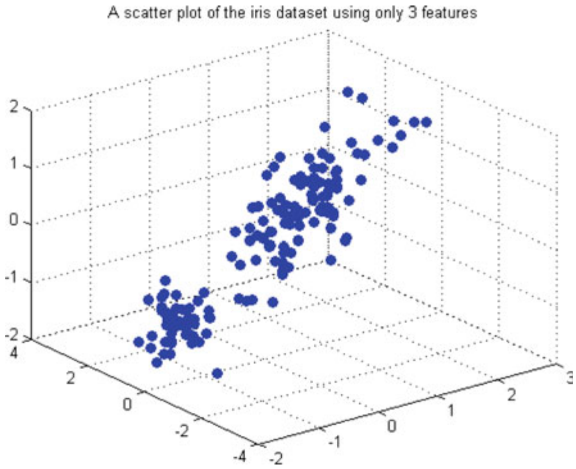


Fig. 2 A count of number of clusters created over 100 runs of normal k-means

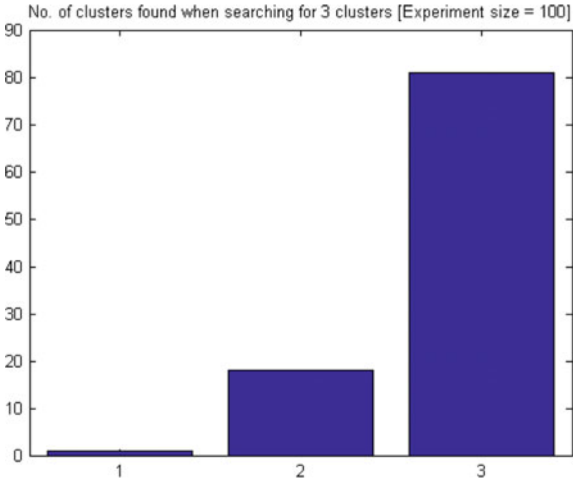
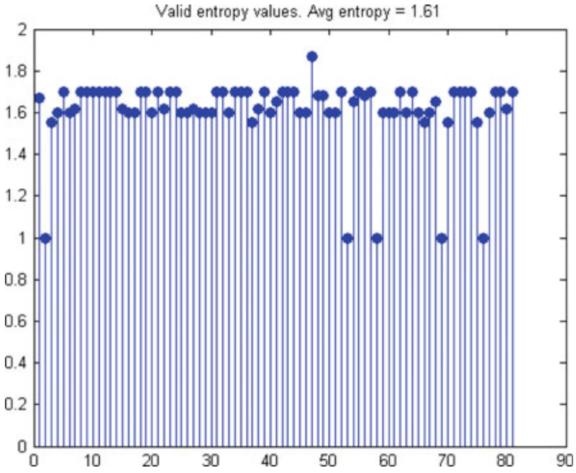


Fig. 3 A plot of the entropy for 100 runs of the original experiment



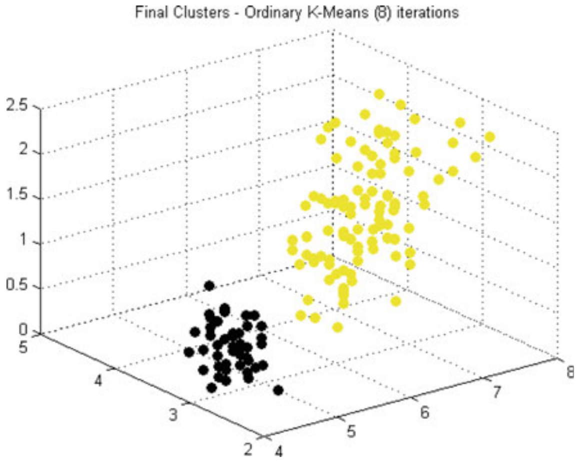


Fig. 4 A result of performing k-means which has only detected two clusters

the purpose of eliminating any anomalous feature values. As can be observed in the figure that an attempt to find three different clusters in the data has often resulted in the finding of only one or two clusters as seen in Fig. 2. Figure 3 shows the entropy of the clusters obtained when the requisite number of clusters was actually formed during the course of a trial. As can be seen that three clusters were only formed about 80 times out of a test set size of 100. Figure 4 shows a typical case where the k-means algorithm could only deduce two clusters at termination.

The results of performing the k-means algorithm using the proposed method have been captured in Figs. 5, 6 and 7. Table 1 provides the parameters used to perform the initial GA phase. It is interesting to observe that the GA preprocessing has only been carried out for a very small number of iterations (i.e., 10) and the fitness function used is as described in Eq. 2. This is sufficient as it is not required

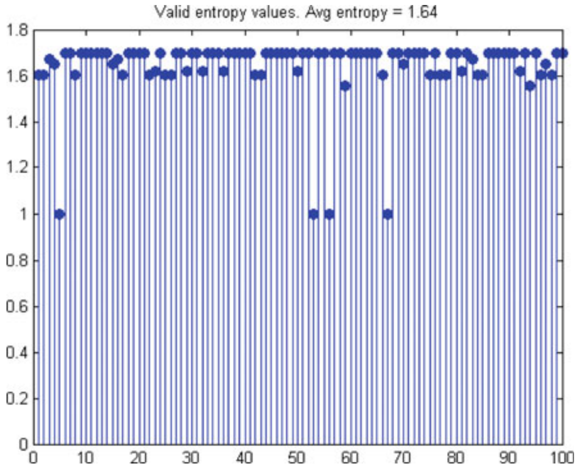


Fig. 5 A plot of the entropy for 100 runs of the improved k-means experiment

Fig. 6 A count of number of clusters created over 100 runs of normal k-means

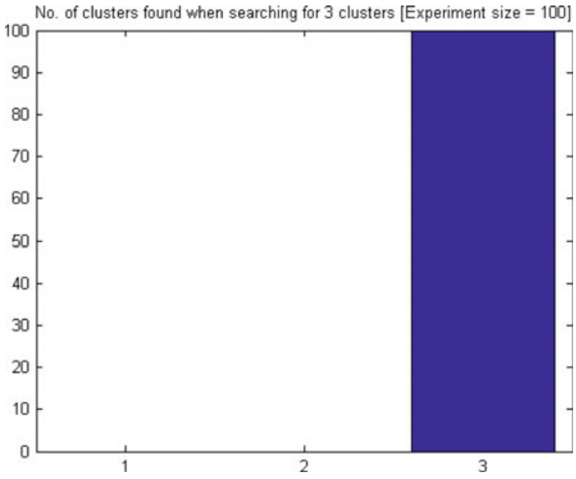


Fig. 7 A demonstration of the final clusters formed using the proposed method

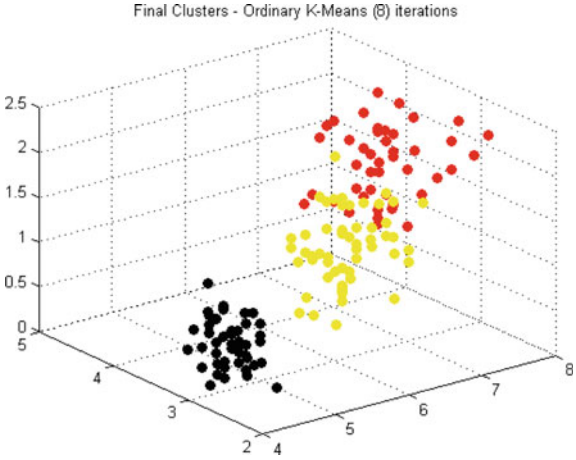


Table 1 Parameter values for GA

Parameter	Value
Population size	50
No. of generations	10
Stall generation limit	10
Crossover	Single point
Crossover fraction	0.8
Mutation	Uniform
Mutation fraction	0.1
Elite count	2

to find any exact set of seed points to launch the k-means algorithm. As can be observed, the proposed algorithm has produced the requisite number of clusters (i.e., 3 in this case), in every single test run. It can also be seen the performance of the algorithm with regards to the veracity of the formed clusters is exactly the same as the basic version of the k-means algorithm, with a reported average entropy value of approximately 1.64 bits.

6 Conclusion

The paper presents a heuristic variant of the k-means algorithm which is assisted by the use of GA in the choice of its initial centers. The proposed algorithm achieves almost the same clustering speed as the original method and guarantees the formation of requisite number of clusters, in very few iterations. Moreover, because the initial centers are not chosen randomly but by evaluating a fitness function that gives good initial points, the clusters formed are very compact and in many cases found to outdo the original version of the k-means algorithm. We are currently in the process of further exploring ways to improve the proposed method even more.

References

1. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. Inf. Sci. **222**, 175–184 (2013)
2. Pham, D., Karaboga, D.: Intelligence Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. Springer Science and Business Media (2012)
3. Song, W., Qiao, Y., Park, S.C., Qian, X.: A hybrid evolutionary computation approach with its application for optimizing text document clustering. Expert Syst. Appl. **42**(5), 2517–2524 (2015)
4. İnkaya, T., Kayalıgil, S., Özdemirel, N.E.: Ant colony optimization based clustering methodology. Appl. Soft Comput. **28**, 301–311 (2015)
5. Bramer, M., Ellis, R., Petridis, M., Yang, X.S.: Firefly algorithm, Levy flights and global optimization. Research and development in Intelligent Systems, vol. XXVI, pp. 209–218. Springer, London (2010)
6. Oztruk, C., Hancer, E., Karaboga, D.: Improved clustering criterion for image clustering with artificial bee colony algorithm. Pattern Anal. Appl. (2014)
7. Gonzalez, J.,Petla, D., Cruz, C., Terrazas, G., Krasnogor, N., Yang, X.-S.: A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, pp. 65–74. Springer (2011)
8. Hatamlou, A.: A Robust data clustering approach using gravitational search algorithm and a heuristic search approach. Global J. Technol. 74–83 (2014)
9. Cui, X., Gao, J., Potok, T.E.: A flocking based algorithm for document clustering analysis. J. Syst. Arch. **52**, 505–515 (2006) (Elsevier)
10. Kalogeratos, A., Likas, A.: Document clustering using synthetic cluster prototypes. Data Knowl. Eng. **70**(3), 284–306 (2011)

11. Singh, V.K., Tiwari, N., Garg, S.: Document clustering using k-means, heuristic k-means and fuzzy c-means. In: 2011 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 297–301. IEEE (2011)
12. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education (2006)
13. Rothlauf, F.: Representations for Genetic and Evolutionary Algorithm. Springer, Berlin (2006)
14. Premalatha, K., Natarajan, A.M.: Genetic algorithm for document clustering with simultaneous and ranked mutation. Modern Appl. Sci. **3**(2) (2009)
15. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Coello, C.: A survey of multiobjective evolutionary algorithms for data mining: part I. IEEE Trans. Evol. Comput. **18**(1), 4–19 (2014)
16. Das, S., Abraham, A., Konar, A.: Automatic hard clustering using improved differential evolution algorithm, Stud. Comput. Intell. 137–174 (2009)

Regression Test Case Prioritization Technique Using Genetic Algorithm

Dharmveer Kumar Yadav and Sandip Dutta

Abstract Regression testing is a maintenance action in which it ensures validity of changed software. Regression testing takes much time to execute the entire test suite and this activity is very costly. In this paper we present a technique which is based on Genetic algorithms (GA) for test case prioritization. Genetic algorithm is a generative algorithm based on natural evolution which generate solutions to optimization problem. In this paper, a new Genetic algorithm is used for regression testing that will prioritize test cases using statement coverage technique. The Algorithm finds fitness function using statement coverage. The results shows the efficiency of algorithms with the help of Average Percentage of Statement Coverage (APSC) metric. This prioritization technique shows optimum results to prioritize the test case. Genetic Algorithm is used to produce the population and it finds the optimal sequence order of test case in regression testing.

Keywords Regression testing · Genetic algorithm · Test case prioritization

1 Introduction

In software development life cycle there is importantly four phases such as phase for requirement, phase for design, phase for coding and testing phase before ready for use [1]. In the coding phase software developers saves the test cases which is used to test the code and that can be reused later whenever software is modified. Software tester team creates a new test case when testing phase begins to validate the software to meet customer requirement. When final product is delivered to user

D.K. Yadav (✉)
Department of Computer Science & Engineering, NIT Jamshedpur, Jamshedpur, India
e-mail: dharmveer.cse@nitjsr.ac.in

S. Dutta
Department of Computer Science & Engineering, Birla Institute of Technology, Mesra
Ranchi, Ranchi, India
e-mail: sandipdutta@bitmesra.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_13

we assumed there is end of software life cycle. But maintenance phase begins after the delivery of the software at the customer's site. Sometimes user need modification, requirement change, addition of new functionality during the use of software product. Then maintenance becomes more difficult due to new released versions and new changes made to previous version. Due to modifications into the existing software or addition of new functionality new faults may be introduced in software which cause it to work improperly. Maintenance phase for software is one of the important activity which includes enhancements of the existing software, corrections of errors, is performed. Regression testing process ensures that no any new errors have been introduced while modified code is retested into previously tested code [2]. When any modification is made to software it is not efficient to rerun all the test cases from test suite. Thus regression testing process is very time consuming activity so software tester need to save the amount of effort and cost. During software testing process, number of test cases is created. In regression testing instead of running the entire test case from test suite we run some of the test cases [3]. So with the selection of subset of test cases from existing test suite regression test suite can be reduced. Regression testing select subset of test cases which maximize the code coverage. In regression testing prioritization of test case schedule test according to some criteria, such as higher coverage, fault.

In the next section paper is organized in five section. Section 2 describes related work in the field of test case prioritization. Section 3 describe about Genetic Algorithm (GA) and proposed method for prioritization of test cases using GA. Section 4 presents results and discussion of our proposed methodology. Finally, conclusion of this research work is described in Sect. 5.

2 Related Work

Many researchers have proposed regression test case prioritization problem and addressed various techniques to solve the problem. Number of methods have been proposed by researchers for test case prioritization like Greedy algorithms prioritization [4], Optimal algorithms [5], non-evolutionary algorithms [6], Logarithmic least square [7], Weight least square method [7], evolutionary algorithms [8], and fault detection technique [9]. In 2010, Huang et al. presented a prioritization technique using genetic algorithm and historic information [10]. Authors evaluated the proposed method's effectiveness. Their technique does not consider similarity between test cases. In 2011, Sabharwal et al. investigated prioritization of test case technique using scenarios from activity diagram and basic information flow of matric with genetic algorithm [11]. In 2011, Andrews et al. Presented technique using genetic algorithm to get the best suitable test cases [12]. In 2009, Chen et al. proposed a new approach called edge partitions dominator graph and Genetic algorithm (EPDG-GA) which use the for the branch coverage [13]. In 2007, Li et al. proposed new search algorithm for test case prioritization [14]. In 2005, Hyunsook et al. presented prioritization techniques based on mutation faults [15]. In 1999, Rothermel et al. have proposed new approach for prioritize test cases using fault detection method in regression testing [16].

3 Proposed Algorithm Using Genetic Algorithm

Genetic algorithm (GA) is called population-based technique for searching. GA is an optimization method which is applied to number of problem to get the best solution. GA can be applied to solve NP-hard problem. In 1975, GA was first introduced by John Holland [17] but further it was studied by author Goldberg [18], De Jong [19] and many others authors. GA use fitness function to get best solution survive and it varies until we get good results.

The population of chromosome in GA is represented by variety of codes like binary code, real number, permutation etc. It uses operator called Genetic operator like selection, crossover, and mutation. Genetic operator is performed on chromosome so that fittest chromosome can be found. Fitness of chromosome calculated using best suitable objective function. Genetic algorithm is different from a random searching method. GA combines features of iterative search technique with random search technique which produce best result for a given problem. One of the best feature of GA is to discover the search space by seeing the entire population of the chromosome.

GA uses the following steps:

1. Generate Chromosome (Population)
2. Calculate the fitness function of generated population
3. Apply selection operator
4. Apply crossover and mutation
5. Evaluate and reproduce the chromosome

3.1 Generating Population (Chromosome)

At the start population is selected randomly basis and it is encoded. Each chromosome represents the possible solution of a given problem. For example 10 test cases are generated T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}, the sequence of test cases is $T_1 \rightarrow T_2 \rightarrow T_5 \rightarrow T_7 \rightarrow T_4 \rightarrow T_{10} \rightarrow T_3 \rightarrow T_6 \rightarrow T_9 \rightarrow T_8$.

3.2 Estimate Fitness of Generated Population

An objective function is used to calculate the fitness of chromosomes. An objective function defines how much it is good or bad about chromosome. Objective function gives a real number from the given input. The real number helps to compare two or more generated population.

In these paper objectives function is taken to find the fitness of test cases. Objectives are statement coverage which is discussed below:

Statement coverage: Statement coverage can be defined as number of statement covered in a given program by a n number of test cases such as $\{T_1, T_2, T_3, T_4, \ldots, T_n\}$. Suppose there is m no. of statement present in program P such as $\{S_1, S_2, S_3, S_4, \ldots, S_m\}$. Suppose statement coverage ST is a function which return the average numbers of statements S covered from $T_{i.}$ test cases.

$$ST_i = \sum S(T_i)/m$$

$\sum S(T_i)$	Total number of statements executed by T_i test cases.
m	Total no. of statements present in the program.
n	Number of n test cases $\{T_1, T_2, T_3, T_4, \ldots, T_n\}$.

3.3 Apply Selection for Individual

Fitness value of the chromosome determines the selection of individual.

3.4 Apply Crossover and Mutation

Parents are chosen and combined on the basis on randomly. Crossover technique used to generate a chromosome randomly. Crossover can be classified into two types:

(i) Single point crossover
(ii) Multi point crossover

For example—suppose we have two sequences of test cases

P_1 (Parent): $T_6 \to T_{10} \to T_1 \to T_4 \to T_5 \to T_7 \to T_8 \to T_{10} \to T_3 \to T_2$
P_2 (Parent): $T_5 \to T_4 \to T_7 \to T_2 \to T_1 \to T_8 \to T_6 \to T_{10} \to T_3 \to T_9$

Offspring C1 and C2 using single point crossover is given below:

C_1: $T_6 \to T_{10} \to T_1 \to T_2 \to T_8 \to T_6 \to T_3 \to T_9$
C_2: $T_5 \to T_4 \to T_7 \to T_4 \to T_5 \to T_8 \to T_{10} \to T_3 \to T_2$

C_1 offspring consists first part of P1 and then second part of P_2. C_2 offspring consists of first part of P_2 and second part of P_1. Randomly two genes are selected to perform mutation with chromosome and swapped with each other.

$T_6 \to T_9 \to T_1 \to T_2 \to T_3 \to T_8 \to T_6 \to T_4 \to T_{10}$
T5 $\to T_3 \to T_7 \to T_8 \to T_5 \to T_4 \to T_{10} \to T_4 \to T_2$

3.5 Termination

The termination criteria can be used when we reach to predefined fitness value.

To apply GA, there is need of two important steps

1. Encoding: Solution of the solution space can be represented using the encoding method.
2. Fitness function: Which measure the fitness of solution, fitness function is required.

3.6 Analysis of Algorithm

To measure the performance of algorithm Average Percentage of Statement Coverage (APSC) metric is used. APSC measures prioritized test suite which covers the maximum number of modified statements. Table 2 shows various orders and corresponding APSC is computed. In this paper, we have used APSC approach, this is given as:

APSC can be calculated as:

$$\mathrm{APSC} = 1 - \frac{TC1 + TC2 + TC3 + TC4 + \cdots + TCn}{n^{*}m} + \frac{1}{2n}$$

where {T_1, T_2, T_3, T_4, …, T_n} is n number of test cases, m represents number of statement covered and TC_i represents First test case covers the statement as shown in (Table 1).

Suppose a software tester initially created test suite in which order of test case is T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10} contains in test suite. Program will be tested through scheduling the test case using this test suite.

Table 1 Shows each test case covered no. of statement

Test case	No. of statement covered										No of statement executed
	1	2	3	4	5	6	7	8	9	10	
T_1		×							×		2
T_2	×		×			×				×	4
T_3										×	1
T_4				×				×			2
T_5	×				×	×		×			4
T_6	×	×			×	×					4
T_7				×			×				2
T_8			×	×							2
T_9			×	×							2
T_{10}					×				×		2

$$\begin{aligned}\mathrm{APSC} &= 1 - \frac{2+1+2+4+5+5+7+4+1+2}{100} + \frac{1}{20}\\ &= 0.72\\ &= 72\,\%\end{aligned}$$

If an order of test cases are arranged in sequence of T_2, T_7, T_1, T_4, T_5, T_3, T_8, T_6, T_{10}, T_9 then APSC can be calculated as for prioritized test cases:

$$\begin{aligned}\mathrm{APSC} &= 1 - \frac{1+3+1+4+5+1+2+4+3+1}{100} + \frac{1}{20}\\ &= 0.80\\ &= 80\,\%\end{aligned}$$

In this paper we proposed four technique to order the test cases unordered, random, and optimal and GA order technique to run the test. In unordered technique, schedule the test cases in the order suppose in test suite TS1. Test cases scheduled randomly in test suite TS2 and optimal test case order is present in test suite TS3, and test case are ordered using GA technique and it is in test suite TS4. We have used APSC metric to measure the effectiveness of all these prioritization techniques.

4 Results and Discussion

The proposed algorithm is implemented on MATLAB. First, statement coverage of each test case is computed. Then algorithm finds the fitness functions which are statement coverage in the program. Fitness function is used to find the best alternative solutions from population (random generation order of test cases). Proposed algorithm is compared with other prioritization techniques such as random prioritization, reverse prioritization. The proposed algorithm using GA order shows that order of test cases cover the maximum number of statement coverage compared to other prioritization method as shown in Table 2. If test cases are scheduled in order T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10} then efficiency of the prioritized test suite is measured through APSC metric which gives 70 % value. We have prioritized suite in T_2, T_7, T_1, T_4, T_5, T_3, T_8, T_6, T_{10}, T_9 order which gives higher value than non-prioritized technique. The Average Percentage Statement Coverage (APSC) measures the average of statement covered by the test cases. APSC values ranges from 0 to 100 and higher value of APSC shows better result. We prioritized test case those covers more no of modified line of code, and a test case which has highest fitness value will be higher priority in prioritization. When genetic algorithm are used a large number of time then we will get a nearly a optimized result.

Table 2 APSC value of prioritized test suite

Technique	% of APSC (%)
No order	72
Random	70
Reverse	75
Optimal	76
GA order	80

5 Conclusion

In this paper to prioritize the test cases Genetic Algorithm we have proposed. Genetic Algorithm finds the optimal sequence order of test case in regression testing. We have generated the test case for object oriented programming to test the program. We have used statement coverage which consists combination of method and parameter in the program. The Algorithm finds fitness function based on statement coverage. The results are shown with the help of APSC (Average Percentage Statement Coverage) values. The APSC has been used evaluate the effectiveness of the proposed algorithm. The result shows that the proposed algorithm which we schedule the test cases in an order that covers the maximum statement coverage and maximize value of APSC (Average Percentage Statement Coverage). The proposed algorithm is more efficient as compared to the other prioritization algorithm. In this paper we have used four technique to order the test cases unordered, random, and optimal and GA order technique to test the program. We have used APSC metric to measure the effectiveness of all these prioritization techniques which is shown in Table 2.

References

1. Agarwal, K.K., Singh, Y.: Software Engineering, 3rd edn. New Age International Publishers (2008)
2. Malishevsky, A.G., Ruthruff, J.R., Rothermel, G., Elbaum, S.: Cost-cognizant test case prioritization. Technical Report. Department of Computer Science and Engineering, University of Nebraska-Lincoln (2006)
3. Rothermel, G., Harrold, M.J.: A safe, efficient regression test selection technique. ACM Trans. Softw. Eng. Methodol. (TOSEM) **6**(2), 173–210 (1997)
4. Pratihar, D.K., Deb, K., Ghosh, A.: A genetic-fuzzy approach for mobile robot navigation among moving obstacles. Int. J. Approx. Reason. **20**(2), 145–172 (1999)
5. Duggal, G., Suri, B.: Understanding regression testing techniques. In Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology (2008)
6. Byson, N.: A Goal Programming Method for Generating Priorities Vectors, pp. 641–648. Journal of Operational Research Society, Palgrave Macmillan Ltd., Houndmills, Basingstoke, Hampshire, RG21 6XS, England (1995)
7. Chu, A.T.W., Kalaba, R.E., Spingarn, K.: A comparison of two methods for determining the weights of belonging to fuzzy sets. J. Optim. Theory Appl. **27**(4), 531–538 (1979)

8. Al-salami, N.M.: Evolutionary algorithm definition. Am. J. Eng. Appl. Sci. **2**(4), 789–795 (2009)
9. Askarunisa, M.A., Shanmugapriya, M.L., Ramaraj, D.N.: Cost and coverage metrics for measuring the effectiveness of test case prioritization techniques. INFOCOMP J. Comput. Sci. **9**(1), 43–52 (2010)
10. Huang, Y. C., Huang, C. Y., Chang, J. R., Chen, T. Y.: Design and analysis of cost-cognizant test case prioritization using genetic algorithm with test history. In: 2010 IEEE 34th Annual Computer Software and Applications Conference (COMPSAC), pp. 413–418 (2010)
11. Sabharwal, S., Sibal, R., Sharma, C.: A genetic algorithm based approach for prioritization of test case scenarios in static testing. In: 2011 2nd International Conference Computer and Communication Technology (ICCCT), pp. 304–309 (2011)
12. Andrews, J.H., Menzies, T., Li, F.C.: Genetic algorithms for randomized unit testing. IEEE Trans. Softw. Eng. **37**(1), 80–94 (2011)
13. Chen, C., Xu, X., Chen, Y., Li, X., Guo, D.: A new method of test data generation for branch coverage in software testing based on EPDG and Genetic Algorithm. In: 3rd International Conference on IEEE Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009, pp. 307–310 (2009)
14. Li, Z., Harman, M., Hierons, R.M.: Search algorithms for regression test case prioritization. IEEE Trans. Softw. Eng. **33**(4), 225–237 (2007)
15. Do, H., Rothermel, G.: A controlled experiment assessing test case prioritization techniques via mutation faults. In: Proceedings of the 21st IEEE International Conference on IEEE Software Maintenance, ICSM'05, pp. 411–420 (2005)
16. Kaur, A., Goyal, S.: A genetic algorithm for regression test case prioritization using code coverage. Int. J. Comput. Sci. Eng. **3**(5), 1839–1847 (2011)
17. John H. Holland.: Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, (1975)
18. Goldberg, D.E.: Genetic Algorithm in Search, Optimization and Machine Learning. Addison Wesley Publishing Company, Reading, MA, vol. 1, p. 98 (1989)
19. De Jong, K.A.: Analysis of the behavior of a class of genetic adaptive systems (1975)

A Comparison Between Genetic Algorithm and Cuckoo Search Algorithm to Minimize the Makespan for Grid Job Scheduling

Tarun Kumar Ghosh, Sanjoy Das, Subhabrata Barman and Rajmohan Goswami

Abstract Major subjects like heterogeneity of resources, dynamic and autonomous character of Grid resources are most important challenges for Grid job scheduling. Additionally, there are issues of various strategies being maintained by the resource providers and followed by resource users for execution of their jobs. Thus optimal job scheduling is an NP-complete problem which can easily be solved by using heuristic approaches. This paper compares two heuristic methods: Genetic Algorithm (GA) and Cuckoo Search Algorithm (CSA) for job scheduling problem in order to efficiently allocating jobs to resources in a Grid system so that the makespan is minimized. Our empirical results have proved that the CSA performs better than the GA.

Keywords Computational grid · Job scheduling · Makespan · GA and CSA

T.K. Ghosh (✉) · S. Barman
Department of Computer Science & Engineering,
Haldia Institute of Technology, Haldia, West Bengal, India
e-mail: tarun.kr.ghosh@gmail.com

S. Barman
e-mail: subhabrata_barman@yahoo.co.in

S. Das
Department of Engineering & Technological Studies,
Kalyani University, Kalyani, West Bengal, India
e-mail: dassanjoy0810@hotmail.com

R. Goswami
Department of Computer Applications, Pailan College
of Management and Technology, Kolkata, West Bengal, India
e-mail: rajmohan.goswami@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_14

1 Introduction

A computational Grid is considered as a hardware and software infrastructure, which provides dependable, consistent, pervasive, and economical access to computational resources existing on the network [1]. Discovery and distribution of resources are done by the Grid Resource Manager (GRM) which is a major component of the Grid scheduler. GRM then performs the scheduling, submission and supervision of jobs in the system. A computational Grid comprises of a large number of computing resources that are scattered globally under control of several organizations each having its own cost, access strategy and other parameters. Therefore, the job scheduling in such complex systems is a challenging problem. More exclusively, the scheduling problem is considered as multi-objective in general. The most important objective is the minimization of makespan of the system. Makespan is defined as the time when Grid executes the latest job.

In this paper, we have used Genetic Algorithm (GA) and Cuckoo Search Algorithm (CSA) separately to minimize the key performance parameter, viz. makespan, of a computational Grid system. Genetic Algorithm (GA) is one of the widely used evolutionary heuristic algorithms for constrained optimization problems; but the main demerit of the algorithm is that it can easily be trapped in local minima. Cuckoo Search Algorithm (CSA) is relatively new evolutionary heuristic algorithm for constrained optimization problems that can be used to perform searches efficiently.

The paper is arranged as follows. In Sect. 2, the framework of Grid job scheduling problem has been defined. Sections 3 and 4 describe GA and CSA methods for Grid scheduling respectively. Section 5 exhibits and compares results obtained in this study. Finally, Sect. 6 concludes the paper.

2 Problem Definition

The paper uses a simulation model of Ali et al. [2] to produce realistic simulations of job scheduling in Grid systems. To present the problem using such simulation model, the computing capacity of each resource, an estimation of the previous load of each resource and an estimation of the computational load of each job are to be provided. Such simulation model is known as *Expected Time to Compute* (ETC) model that is nothing but an $m \times n$ matrix in which m denotes the number of jobs and n denotes the number of resources. Probable run time of a given job on each resource is specified by each row of the ETC matrix. Likewise, probable run time of a given resource for each job is specified by each column of the ETC matrix. The probable run time of job i on the resource j is denoted by ETC[i, j]. For our simulation studies, a variety of possible heterogeneous environments is considered and accordingly characteristics of the ETC matrices are varied.

Here, our objective is to minimize the makespan of the job scheduling problem in Grid system. The makespan is formally defined as:

$$\text{makespan} = \min_{S_j \in Sch} \left\{ \max_{i \in Jobs} F_i \right\} \tag{1}$$

where *Sch* denotes the collection of all possible schedules in the system, F_i indicates the time when job i finishes, and *Jobs* represents the collection of all jobs to be scheduled.

3 Genetic Algorithm (GA)

Developed in 1975 by John Holland, Genetic Algorithm (GA) is a stochastic search optimization technique that mimics the evolutionary processes in biological systems. The main idea behind the GA is to combine exploitation of best solutions from previous searches with the exploration of new areas of the solution space. The GA for scheduling problem can be formulated as: First, an initial population of chromosomes is generated either randomly or using other heuristic algorithm. A chromosome in the population specifies a possible solution. As far as our scheduling problem is concerned, a solution is a mapping sequence between jobs and resources. Next, each chromosome is evaluated for a fitness value associated with it. The fitness value measures degree of goodness of individual chromosome compared to others in the population. A new iteration is created next by using the genetic operators, namely selection, crossover and mutation; in other words, the population evolves [3]. Lastly, the fitness value of each chromosome from this newly evolved population is assessed again. This way one iteration of the GA is completed. When a predefined number of iterations is met or all chromosomes have converged to the same mapping, the algorithm stops. The major steps of GA can be summarized as pseudo code in Algorithm 1:

Algorithm 1: Genetic Algorithm (GA)

```
An initial population $P(t = 0)$ of $n$ individuals (chromosomes) is generated.
Fitness of each individual of the population is evaluated. Evaluate ($P(t)$).
while (terminating condition or maximum number of iterations is not reached) do
    A subset of $x$ pairs from $P(t)$ is selected. Let $P_1(t)$ = Select($P(t)$).
    Each of the $x$ chosen pairs is crossed with probability $p_c$. Let $P_2(t)$ = Cross ($P_1(t)$) be the set of off-springs.
    Each offspring in $P_2(t)$ is mutated with probability $p_m$. Let $P_3(t)$ = Mutate($P_2(t)$).
    Fitness of each offspring is evaluated. Evaluate ($P_3(t)$).
    A new iteration from individuals in $P(t)$ and $P_3(t)$ is created. Let $P(t + 1)$ = Replace ($P(t); P_3(t)$); $t = t + 1$.
end while
Post process results and visualization.
```

4 Cuckoo Search Algorithm (CSA)

Yang and Deb in 2009 have proposed the Cuckoo Search Algorithm (CSA) based on the Lévy flight behavior and brood parasitic behavior [4]. The CSA has been shown to produce very good results in constrained optimizations. The cuckoo birds follow an aggressive reproduction process. Female birds take control and lay their fertilized eggs in other birds' nests. If the host bird realizes that the egg is not its own, it either throws the alien egg away or rejects the nest and constructs a new one at other place.

Every single egg in a nest indicates a probable solution and each cuckoo egg indicates a new solution. Here, purpose is to find out new and possibly superior solutions (cuckoos) to substitute old solutions in the nests. For simplicity, it is assumed that each nest has one egg. In [4, 5], it is proved that the CSA can be stretched further to more complex situations when each nest contains multiple eggs representing a set of solutions. The CSA can be summarized by following three important rules:

- A single egg is laid by each cuckoo at a time and is disposed in a haphazardly selected nest.
- Most promising nests carrying excellent quality of eggs (solutions) will switch to the next iterations.
- There are a fixed number of available host nests and a host-bird can recognize a foreign egg with probability $p_a \in [0, 1]$. In this situation, the host-bird can either eliminate the foreign egg from the nest or dump the nest to erect an entirely new nest in a new place [4].

In practice, the third hypothesis can be estimated by a probability p_a of the n nests being substituted by new ones, having new arbitrary solutions. In the case of an optimization problem, the superiority or fitness of a solution is directly related with the objective function.

In CSA, the walking steps of a cuckoo are determined by the Lévy flights. Following Lévy flight is computed to generate new solutions $x_i\ (t + 1)$ for the ith cuckoo:

$$x_i(t+1) = x_i(t) + \alpha \oplus L\acute{e}vy(\lambda) \tag{2}$$

where $\alpha > 0$ symbolizes the step length that must be correlated to the degree of the problem concerned. The product symbol $\oplus$ specifies entry-wise multiplications [5]. This paper has considered a Lévy flight where step-lengths are scattered as per the probability distribution given as:

$$L\acute{e}vy\, u = t^{-\lambda}, \quad 1 < \lambda \leq 3 \tag{3}$$

which has an unlimited variance. The successive steps of a cuckoo effectively constitute a random walk procedure obeying a power-law step-length distribution with a heavy tail. On the basis of previously cited rules, the main ideas of the CSA are summed up as pseudo code in Algorithm 2.

Algorithm 2: Cuckoo Search Algorithm (CSA)

```
Fitness function $f(x), x = (x_1, ....., x_d)^T$ is determined.
An initial population of $n$ host nests $x_i$ $(i = 1, ...., n)$ is generated.
while (terminating condition or maximum number of iterations is not met) do
    A cuckoo (say, $i$) is obtained arbitrarily and a new solution by Lévy flights is
    generated.
    Quality / fitness of the new solution is evaluated. Let it be $F_i$.
    A nest among $n$ (say, $j$) is chosen arbitrarily.
    if ($F_i > F_j$)
        Nest $j$ is replaced by the new solution.
    end if
    A fraction ($p_a$) of worse nests is abandoned and new ones are constructed at
    new sites using Lévy flights.
    Most promising solutions (or nests having higher quality solutions) are
    recorded.
    All available solutions are graded and the current best solution is determined.
end while
```

5 Experiments and Results

For solving the job scheduling problem in computational Grid systems, we have used GA and CSA separately. In both algorithms, the objective is to minimize the makespan. The algorithms were developed using MATLAB Release 2013A and run on a PC with 2.6 GHz processor and 4 GB of RAM. In order to achieve the maximum possible performances of GA and CSA, fine alteration has been done and best values for their parameters are chosen which have been provided in Table 1. In this work, the benchmark proposed by Ali et al. [2] for simulating the heterogeneous computing environment is adopted.

Like the simulation model in [2], our model is also based on ETC matrix for 512 jobs and 16 resources. Based on the three parameters: job heterogeneity, resource heterogeneity and consistency; the instances of the benchmark are categorized into 12 different types of ETC matrices. Job heterogeneity in ETC matrix represents by the quantity of inconsistency among the run times of jobs by a given resource. Resource heterogeneity specifies possible disparity among the run times for a given job by all the resources. Also an ETC matrix is said to be consistent when any job J_j is executed by resource R_i in lesser time compared to resource R_k. That means,

Table 1 Parameter settings for the algorithms

Algorithm	Parameter name	Parameter value
GA	Number of individuals	25
	Crossover probability (p_c)	0.8
	Mutation probability (p_m)	0.07
	Scale for mutations	0.1
CSA	Number of nests	20
	Mutation probability value (p_a)	0.25
	Step size (α)	1.5

resource R_i executes all jobs faster than resource R_k. On the other hand, an ETC matrix is said to be inconsistent when resource R_i may take lesser times than resource R_k for some jobs and longer for others. Partially-consistent matrices are inconsistent matrices that contain a consistent sub-matrix of a predefined size [2]. In our work, we have used uniform distribution to generate the matrices. 512 jobs and 16 resources are considered for each instance which is labeled as *x-yy-zz* according to:

- *x* represents the class of inconsistency; *c* implies consistency, *i* implies inconsistency, and *p* implies partially-consistency.
- *yy* reflects the job heterogeneity; *hi* implies high and *lo* implies low.
- *zz* reflects the resource heterogeneity; *hi* implies high and *lo* implies low.

The obtained makespans using GA and CSA are compared in Table 2. The results are actually obtained after taking average of five simulations. In Table 2, the first column represents the instance name, the second and third columns represent the mean makespan (in second) obtained by GA and CSA respectively. The comparison of statistical results using two algorithms in terms of the mean makespan for the 12 instances is depicted in Fig. 1. As it is evident from Fig. 1 and Table 2, CSA can minimize the makespan better than GA in general.

Table 2 Comparison of statistical results for mean makespan (in second) obtained by GA and CSA

Instance	GA	CSA
c-hi-hi	21507697	19578688
c-hi-lo	235654	226309
c-lo-hi	696122	554048
c-lo-lo	8102	7786
i-hi-hi	21041956	20120873
i-hi-lo	244908	237037
i-lo-hi	692452	648302
i-lo-lo	8278	7594
p-hi-hi	21250983	20064476
p-hi-lo	242350	227859
p-lo-hi	712203	692873
p-lo-lo	8233	8044

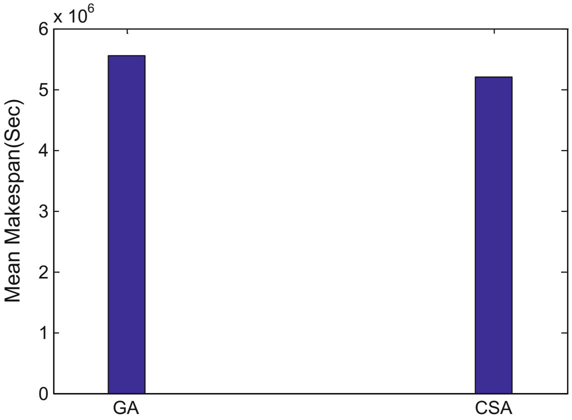


Fig. 1 Mean makespan comparison between GA and CSA

6 Conclusions

Job scheduling in computational Grid is considered as an NP-complete problem. Thus, employing heuristic techniques is an apt idea in order to deal with the diverse issues associated with the job scheduling problem in Grid. This paper investigates two heuristic job scheduling algorithms in Grid environments as optimization problems. The GA and CSA are implemented in the paper for scheduling jobs in computational Grids so as to minimize the makespan. After performing an exhaustive simulation tests on varied settings as cited earlier, the performances of two algorithms are recorded separately. Experimental results show that CSA outshines GA in all respects.

References

1. Foster, I., Kesselman, C.: The grid 2: blueprint for a new computing infrastructure. Morgan Kaufmann (2003)
2. Ali, S., Siegel, H.J., Maheswaran, M., Hensgen, D., Ali, S.: Representing task and machine heterogeneities for heterogeneous computing systems. Tamkang J. Sci. Eng. **3**(3), 195–207 (2000)
3. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. Wiley, New York, NY, USA (2004)
4. Yang, X.S., Deb, S.: Cuckoo search via levy flights. In: Proceedings of World Congress on Nature and Biologically Inspired Computing, pp. 210–225 (2009)
5. Yang, X.S., Deb, S.: engineering optimization by cuckoo search. Int. J. Math. Model. Numer. Optim. **1**(4), 330–343 (2010)

A Framework for Budget Allocation and Optimization Using Particle Swarm Optimization

Keshav Sinha, Annu Priya and Moumita Khowas

Abstract One of the most frequently encountered problems in the real world financial sector is Budget Allocation. Budget Allocation keeps in line the planning of the actual operations by managing concern problem before they arise. A given budget information supports the development operations of managing, planning and controlling aspects of any given setup. There are several conventional technique for budget allocation like OCBA, EA, MOCBA, etc. For large scale budgeting problem the performance of conventional technique degrades. We like to propose an evolutionary computing based infrastructure for Budget Allocation and Optimization and make a comparative performance analysis on Particle Swarm Optimization (PSO) based Optimal Computing Budget Allocation (OCBA) and PSO based Equal Allocation (EA).

Keywords Particle swarm optimization • Budget allocation • OCBA • MOCBA

1 Introduction

A modern heuristic based approach to solve nonlinear and dis-continuous predicaments is particle swarm optimization. PSO is generally used to optimize the problem by dubbing the particles that present in the search space using mathematical formula over the velocity and position to find local best position. In the

K. Sinha (✉) · A. Priya
Department of Computer Science and Engineering,
Birla Institute of Technology, Mesra, India
e-mail: keshav.sinha@yandex.com

A. Priya
e-mail: annu.priya12@yahoo.com

M. Khowas
Department of Economics, Sidho-Kanho-Birsha University, Purulia, India
e-mail: moumita_khowas@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_15

searching space the best known position can be found by the other particles. This move the swarm towards the best solution. Kennedy and Eberhart (1995) first proposed PSO based on the behavioral analogy of swarm bird and school of fish. Mimicking the biological analogy each particle in the swarm will move through a search space according to its velocity value based on the location information of both the best solution that is found by any of the particles that this particle can communicate with. To resolve issues related to rapid convergence to local optimal and avoid the global optimal with slow convergence rate, inertia weight was incorporated in the velocity update equation by Shi and Eberhart (1998). Another important factor known as the "constriction factor" into the velocity update equation was introduced by Clerc and Kennedy [1] to build a general purpose PSO model, based on similar considerations of Shi and Eberhart [2]. Beside PSO a traditional approach to resolve optimization problems [3–5] (Sahana et al. 2015) is Genetic Algorithm (GA). The GA follows the inspiration of the fundamental laws of genetic and evolution, and mimics the reproduction behavior seen in biological populations.

This paper will be primarily concerned with discussing and comparing the computational efficiency and effectiveness of PSO based implementation for OCBA and EA optimization problem. The PSO fits with the OCBA via computing budget allocation by maximization of the convergence rate along with the probability of incorrect selection.

The motivation for budget allocation is one of the most challenging tasks since it requires an optimization from the coarse level to fine level. Besides budget allocation is the key which decides the working flow of any sector whether it's for profit organization or a NGO for community service. The key task is to distribute an amount of '*X*' optimally for the fascial year such that the all the goals or at least near about the results to the objectives thought have been achieved considering user constraints.

This paper is organized in five major sections. Section 1 gives introduction followed the literature survey in Sect. 2. Section 3 deals with the research methodology consist of purposed algorithm and flow chart. In Sect. 4, the comparative study has been performed on two evolutionary computing based implementations. Finally, Sect. 5 draws the conclusion.

2 Literature Survey

There are various techniques used for budget allocation problem. Among those, the state-of-the-art approach to intelligently allocate and computing budget for efficient simulation optimization is OCBA. The OCBA approach can intelligently determine the most efficient simulation replication numbers or simulation lengths for all simulated alternatives. Ranking and selection methodologies make use of statistical methods specially developed to select the best system design or a subset that contains the best system design from a set of '*n*' competing alternatives. Ranking and selection methodology mostly focus on a single measure of system performance. However, most real-life systems and designs often have multiple objectives in nature. For example, in product design and development, the optimization of cost and the quality of products are two conflicting objectives. The concept of Pareto optimality into the ranking and selection scheme have been introduced to look for all of the non-dominated designs rather than a single "best" one. Pareto optimality, is a state of allocation of resources in which it is impossible to make any one individual better off without making at least one individual worse off. The major challenges for budget allocation problem lies on (i) How to allocate replications to certain designs and (ii) Multi-objective ranking and selection (MORS) problem. A subsequent number of researches have been performed by different researches on this topic are as cited below in Table 1:

3 Research Methodology

After exhaustive studying of different research papers the methodology for solving budget allocation problem can be suggested and proposed as shown in Fig. 1.

The step by step procedure for proposed budget allocation strategy is discuss as follows:

1. In infrastructure model, a database need to be created regarding number of assets for future use.
2. The infrastructure performance can be evaluated by the condition index which may be represented in terms of quality and quantity. The information can be used for budget allocation and optimization.
3. The objective of budget allocation is to identify the asset for infrastructure. Our motive is to enhance the performance and select minimum acceptable target.
4. An evolutionary computing technique need to be chosen according to the problem specification and environmental setup. The first step of any Evolutionary computing technique (GA, PSO etc.) is to set the initialization parameters. Initialization parameter for Genetic Algorithm are as follow (i) Number of Generations, (ii) Number of population, (iii) Mutation Rate, (iv) Mutation percentage on population, (v) Crossover percentage on population. The initialization parameter for PSO are consist of (i) Number of particles, (ii) the size of the key, (iii) maximum number of Iterations, (iv) Self-confidence factor, (v) Swarm confidence factor, and (vi) inertia weight. The most important step is to find fitness assignment for the selected evolutionary computing technique.

In this paper PSO is considered for the evolutionary computing technique. The working principle of PSO is as shown below with some basic modifications:

(a) The first phase of PSO is initialization, where the particles are present in consistent order across the searching space. The velocity for each particles are present in randomized form. According to the Fitness value for each particles,

Table 1 Chronological literature review

Research	Technique used	Methodology	Advantages	Disadvantages
Bechhofer el al. [6], Swisher el al. [7], Kim and Nelson [8]	Indifference zone (IZ)	Selection is guaranteed and it select the best design among other design	Increase in accuracy	Replication is present there
Rinott [9]	Two stage—IZ	Selecting the best design on single performance and in second stage replication is allocated based on variance	Improve the probability of correct selection	Replication allocation
Fu et al. [10]	Independence assumption in OCBA	Correlated sampling	Performance measure is independent	Decrease in efficiency in the case of multiple performance measure
Chen et al. [11]	Applicability of OCBA	Allocation rule in finding optimal subset instead of single best design	Increase in efficiency	Less performance in multi objective
Lee et al. [12]	Multi objective OCBA	Minimize the resources idle time. And selection of alternatives with high waiting time	Performance increased	Scheduling is biggest challenge
Davis [13]	GA	Binary encoding scheme for fitness function	It provides flexibility	Layout problem, optimization problem
Hussien Al-Battaineh and AbouRizk [14]	GA_OCBA	Chromosome encoding	Multiple problem solving	No assurance of finding the global best
Rubinstein and Shapiro [15]	Stochastic approximation	Estimated by noisy observation	Maximize the exception	Unreliable stopping criteria
Kennedy and Eberhart [16]	PSO	Decision making using neighbors expressences	Robustness and easy implementation	Replication
Storn and Price [17]	Differential evolution	Random searching in the solution space	Random search	Unstable convergence
Zhang et al. [18]	PSObw_OCBA PSOs_OCBA	Maximize the convergence rate and probability of correct selection	Improve the computation efficiency	Replication

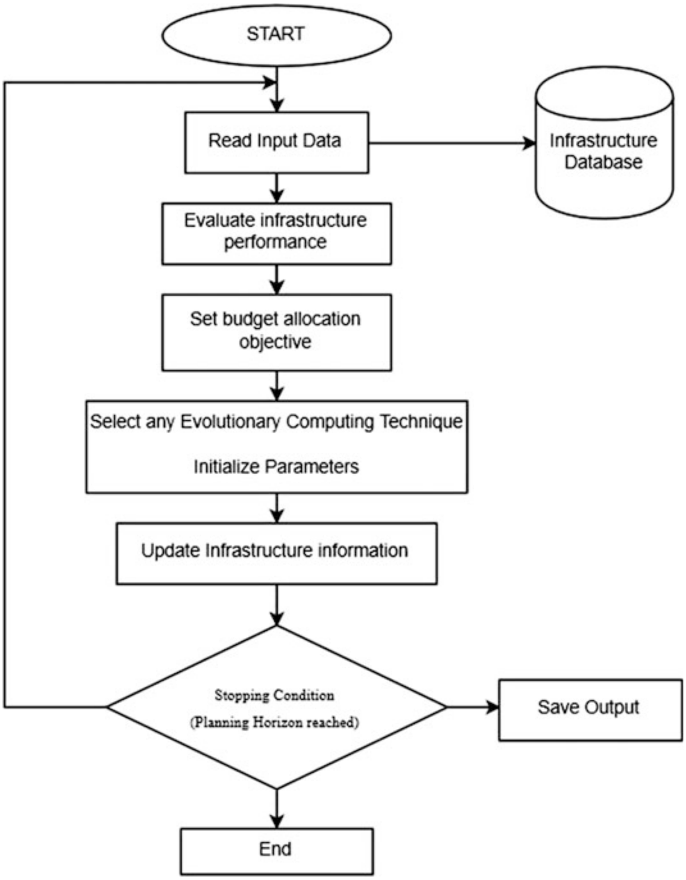

START
Read Input Data
Infrastructure Database
Evaluate infrastructure performance
Set budget allocation objective
Select any Evolutionary Computing Technique
Initialize Parameters
Update Infrastructure information
Stopping Condition
(Planning Horizon reached)
Save Output
End

Fig. 1 Solution strategy for budget allocation problem

we get the initial values are $P_i = (P_{i1}, P_{i2},\ldots, P_{i\mathrm{D}})$ and $P_g = (P_{g1}, P_{g2},\ldots, P_{gD})$. Set $t = 1$;

(b) The second phase is updating the position of swarm.

For each particle '*i*'. Start the loop and update the velocity 'V_i' and position 'X_i' using the Eq. (1) and (2).

$$V_{id} = X\left(v_{id} + c_1\varepsilon_1\left(p_{id} - x_{id}\right) + c_2\varepsilon_2\left(p_{gd} - x_{id}\right)\right) \tag{1}$$

$$X_{id} = x_{id} + v_{id} \tag{2}$$

End of the loop;

Calculate the fitness value for new particle's and then update the value for P_i and P_g.

(c) In final phase we check for stopping criteria if is satisfied then stop the loop, otherwise the loop is set to be $t = t+1$ for updating the loop.

Whereas:

P_i	Distance to personal best
P_g	Distance to global best
V	Update velocity
X	Constrictive factor
C_1	Local best convergence
C_2	Global best convergence
ε_1 and ε_2	Random number

5. The stopping criteria is set up according to the optimization objective in terms of time or convergences specification. The stopping condition of PSO is the maximum number of iterations or minimum error requirement present or acceptable convergences. The stopping condition for GA is (i) if the generation's value reaches the intended value of Generations, (ii) amount of time is equal to intended Time limit, (iii) fitness function is less than or equal to intended Fitness limit, (iv) Stall generations is less than intended Function tolerance and (v) objective function interval equal to Stall time limit. Finally the information is to be updated and optimum strategy is developed for the infrastructure.

4 Implementation and Comparative Study

The OCBA is one of the most efficient approach for finding the simulation replication or length from all the simulated alternatives. For attaining the desired result OCBA approach for quality decision simulation using fixed computing budget. In Equal Allocation (EA) all the design are simulated in equal environment so that we get better performance.

We have considered the sphere function for simulation purposes. For selecting the Sphere Function in dimension D and finding d local minima except the global one.

$$\text{Sphere Function:} f(x) = \sum_{i=1}^{d} Xi^2$$

The minimal value for sphere function is zero. In two dimension space the optimum value is (0, 0). We set the range of [−50, 50] and variance 10^2 is added to sphere function for simulation under stochastic environment. For PSO implementation we set 20 particles for each iteration. The goal is to optimize range and find the optimal solution for each function

$$X = \frac{\max\_iter + 1 - i}{\max\_iter + 1} \cdot \frac{2}{\left|2 - c1 - c2 - \sqrt{(c1 + c2)2 - 4(c1 + c2)}\right|}$$

Where $c1$ and $c2$ is set to be 2.05, as proposed by Bratton and Kennedy [19]. Max_iter is equal to maximum number of iteration. For budget allocation, we set the $\Delta = 100$ and $n_0 = 10$ for all experiment. For each iteration the total budget allocation at PSO is 3000. The computing budget allocation models for two versions of PSO using OCBA allocation rules PSOs_OCBA (PSO Standard OCBA) and PSObw_OCBA (PSO Best half and Worst Half OCBA) are analyzed for finding the Mean performance of the Global Best. In standard PSO, the personal best of one particle is the location of this particle's own previous best performance and the global best is the best solution that any particle has found. In PSObw, a modified version of PSOs, the global best is the best particle in the swarm. For each particle in S_{worst}, we need to find the particle in S_{best} nearest to it as its personal best while the personal best for particles in S_{best} are themselves.

Further, another two version of Equal Allocation (EA) ruled based PSO, namely PSOs_EA (PSO standard EA) and PSObw_EA (PSO best half and worst half EA) are also analyzed for finding the mean performance of global best.

The mean Performance and global best for PSO implementation of PSObw_OCBA, PSOs_OCBA, PSObw_EA, and PSOs_EA are as shown in Figs. 2 and 3 respectively and the corresponding result dataset is cited in Table 2.

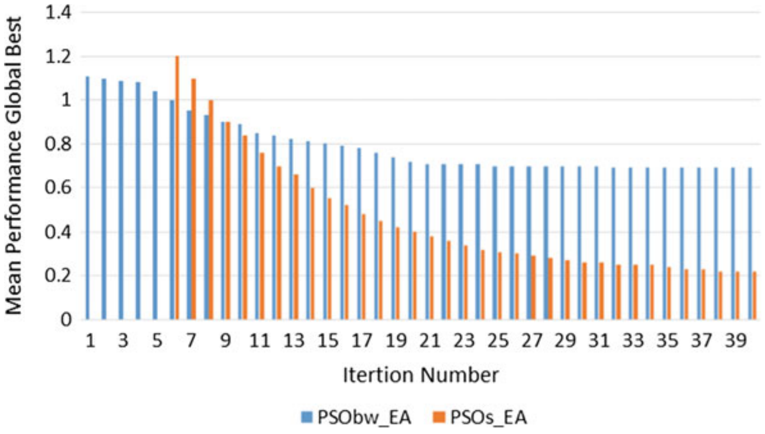


Fig. 2 PSObw_EA and PSOs_EA using sphere function

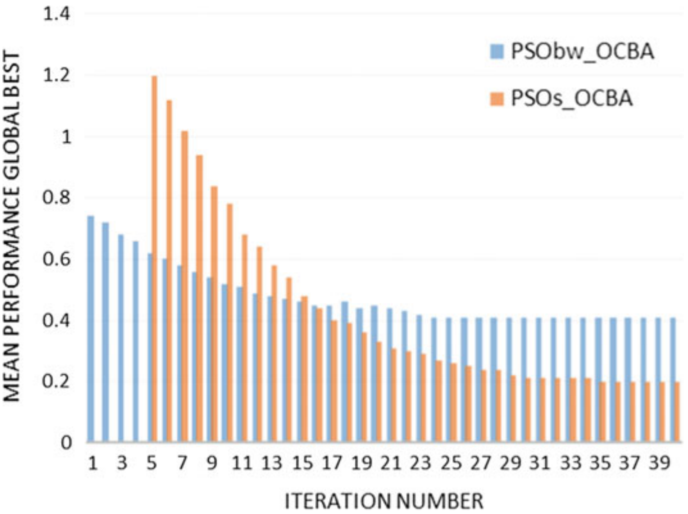


Fig. 3 PSObw_OCBA and PSOs_EA using sphere function

Table 2 The result of PSObw_OCBA, PSObw_EA, PSOs_OCBA and PSOs_EA for finding the mean performance of the global best

Iteration no	PSObw_OCBA	PSObw_EA	PSOs_OCBA	PSOs_EA
1	0.74	1.11		
2	0.72	1.10		
3	0.68	1.09		
4	0.66	1.08		
5	0.62	1.04	1.20	
6	0.60	1.00	1.12	1.2
7	0.58	0.95	1.02	1.1
8	0.56	0.93	0.94	1.0
9	0.54	0.9	0.84	0.9
10	0.52	0.89	0.78	0.84
11	0.51	0.85	0.68	0.76
12	0.49	0.84	0.64	0.70
13	0.48	0.82	0.58	0.66
14	0.47	0.81	0.54	0.60
15	0.46	0.80	0.48	0.55
16	0.45	0.79	0.44	0.52
17	0.45	0.78	0.40	0.48
18	0.46	0.76	0.39	0.45
19	0.45	0.74	0.36	0.42
20	0.45	0.72	0.33	0.40

(continued)

Table 2 (continued)

Iteration no	PSObw_OCBA	PSObw_EA	PSOs_OCBA	PSOs_EA
21	0.44	0.71	0.31	0.38
22	0.44	0.71	0.30	0.36
23	0.43	0.71	0.29	0.34
24	0.42	0.71	0.27	0.32
25	0.41	0.70	0.26	0.31
26	0.41	0.70	0.25	0.30
27	0.41	0.70	0.24	0.29
28	0.41	0.70	0.24	0.28
29	0.41	0.70	0.22	0.27
30	0.41	0.70	0.21	0.26
31	0.41	0.70	0.21	0.26
32	0.41	0.69	0.21	0.25
33	0.41	0.69	0.21	0.25
34	0.41	0.69	0.21	0.25
35	0.41	0.69	0.20	0.24
36	0.41	0.69	0.20	0.23

5 Conclusion

In this paper a novel try has been performed for introducing PSO for budget allocation problem. It is found that PSObw_OCBA and PSObw_EA has a better performance than PSOs_OCBA and PSOs_EA. OCBA exhibits better execution results than EA, which is a traditional approach. Implementation using GA instead of PSO can also be possible for budget allocation problem as per the proposed framework and the simulation results can be analyzed to find out the scope of GA for the said problem.

References

1. Clerc, M., Kennedy, J.: The particle swarm: explosion, stability and convergence in a multi-dimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
2. Shi, Y., Eberhart R.C.: Empirical study of particle swarm optimization. IEEE (2009)
3. Sahana, S.K., Jain, A.: An improved modular hybrid ant colony approach for solving traveling salesman problem. Int J Comput. (JoC) **1**(2), 123–127 (2011)
4. Sahana, S.K., Jain, A., Mahanti, P.K.: Ant colony optimization for train scheduling: an analysis, I.J. Intell. Syst. Appl. **6**(2), 29–36 (2014)
5. Srivastava, S., Sahana, S.K., Pant, D., Mahanti, P.K: Hybrid microscopic discrete evolutionary model for traffic signal optimization. J. Next Gener. Inf. Technol. (JNIT) **6**(2), 1–6 (2015)

6. Bechhofer, R.E., Santner, T.J., Goldsman, D.M.: Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons. Wiley, New York (1995)
7. Swisher, J.R., Jacobson, S.H., Yücesan, E.: Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: a survey. ACM Trans. Model. Comput. Simul. **13**, 134–154 (2003)
8. Kim, S.-H., Nelson, B.L.: Selecting the best system: theory and methods. In: Chick, S., Sánchez, P.J., Ferrin, D., Morrice, D.J. (eds.) Proceedings of the Winter Simulation Conference, pp. 101–112. Institute of Electrical and Electronics Engineers, Inc, Piscataway, New Jersey (2003)
9. Rinott, Y.: On two-stage selection procedures and related probability-inequalities. Commun. Stat.-Theor. Meth. **A7**, 799–811 (1978)
10. Fu, M.C., Hu, J.H., Chen, C.H., Xiong, X.: Simulation allocation for determining the best design in the presence of correlated sampling. Informs J. Comput. **19**, 101–111 (2007)
11. Chen, C.H., He, D., Fu, M., Lee, L.H.: Efficient simulation budget allocation for selecting an optimal subset. Informs J. Comput. **20**(4), 579–595 (2008)
12. Lee, L.H., Chew, E.P., Teng, S., Goldsman, D.: Optimal computing budget allocation for multi-objective simulation models. In: Ingalls, R.G., Rosetti, M.D., Smith, J.S., Peters, B.A. (eds.) Proceedings of the 2004 Winter Simulation Conference, pp.586–594. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey (2004)
13. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, United States (1991)
14. Al-Battaineh, Hussien, AbouRizk, Simaan: Optimization of intermediate-level budget allocation using a genetic algorithm. Int. J. Archit. Eng. Constr. **2**(3), 144–157 (2013)
15. Rubinstein, R.Y., Shapiro, A.: Discrete Event Systems: Sensitivity Analysis and Stochastic Approximation using the Score Function Method. Wiley (1993)
16. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on 12 Oct 1997, vol. 5, pp. 4104–4108. IEEE (1997)
17. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. J. Global Optim. **11**(4), 341–359 (1997)
18. Zhang, S., Chen, P., Lee, L.H., Peng, C.E., Chen, C.-H.: Simulation optimization using the particle swarm optimization with optimal computing budget allocation. In: Jain,S., Creasey, R.R., Himmelspach, J., White, K.P., Fu, M. (eds.) Proceedings of the Winter Simulation Conference (2011)
19. Bratton, D., Kennedy, J., Neves, J., Defining a standard for particle Swarm Optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 120–127. (2007)

Gene Expression Profiling of Cervical Cancer Using Statistical Method

Deepak Kapse, Koel Mukherjee and Debadyuti Banerjee

Abstract Cervical cancer is accountable for numerous cancer-related deaths in women worldwide. Cancer causes molecular alterations in two types of genes with opposing functions, proto-oncogenes and tumor suppressor genes (TSG), respectively. Proto-oncogenes stimulate cell growth and hinder apoptosis, whereas TSGs inhibit growth and maintain the cell integrity. Deregulation of both types of genes may change the growth and division of cells, leading to a tumorigenic transformation. Thus we aim to study the gene expression of cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC). The data were selected and retrieved from TCGA data portal. A list of 12 driver genes responsible for causing cervical cancer was found. A code in R was written to find the correlation of these driver genes with the cancer genes by Spearman's method. Different statistical methods were applied to calculate the significantly co-expressed genes. Co-express pathways were also identified by DAVID.

Keywords Correlation · Spearman's method · Bonferroni correction · Multiple testing correction · DAVID R

1 Introduction

Every day the body discards the old cells by replacing the new ones. But sometimes this process becomes hampered [1] in the genome level leading to uncontrolled cell growth and metastasis [2]. In this situation the old cells are also not discarded when they should be. These additional cells can form tumor which is the accumulation of new and old ones. Occasionally these accumulations of cells can be malignant in its nature. Among several cancer types, cervical cancer is accountable for 10–15 % of deaths in women worldwide [3]. As per the Human Papillomavirus (www.

D. Kapse · K. Mukherjee (✉) · D. Banerjee
Bioinformatics Lab, Department of Bio-Engineering, Birla Institute of Technology,
Mesra, Ranchi 835215, Jharkhand, India
e-mail: koelmukherjee@bitmesra.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_16

hpvcentre.net) and Related Diseases Report, 1,22,844 women diagnosed every year and 67,477 die among them. Cervical cancer holds the rank of second position among all other cancers in Indian women where 432.20 million women are at the threat of developing cervical cancer [4]. According to GLOBOCAN statistics [5], by 2030, approximately, 2.14 million new cases will be diagnosed and 13.2 million deaths from cancer will occur.

Cervical cancer occurs in the cervix which is the lower part of the uterus [3]. Cervical cancer can often be successfully treated if it is detected at an early stage. Most cervical cancers are caused by a virus called human papillomavirus, or HPV [6].

Much importance has been given to RNA-Seq data to characterize numerous tumor samples and their cell lines, respectively, after the encyclopedia of the regulatory elements (ENCODE) and The Cancer Genome Atlas (TCGA) projects have submitted their projects. The identity and function of a cell is determined by its entire RNA component, which is called the transcriptome [7]. Snapshot of the transcriptomic status of the disease is provided by RNA-seq data by discovering the novel genes, pathways [8] and their expression levels.

Based on the known and novel driver mutations at protein coding genes and their mRNA expression levels, the patients are classified into different T-cell subtypes, which has important implication for personalized medicine. RNA-Seq similarly helps to identify biomarkers for breast cancers [9]. This work will provide knowledge about the genes related to cervical cancer and also its driver genes. To expand knowledge at the molecular level of cervical cancer, it is essential to build an integrated view of genomic data with transcription profiles, mutations, miRNA, epigenetic markers and clinical end points. The computational work helps to predict and analyze the result related to transcriptome data which may in future help to diagnose the women candidate suffering from the cervical cancer. The study can show a glimpse of the present scenario of the genes related to cervical cancer and simultaneously help the future researchers to solve the mystery of cancer.

2 Materials and Methods

2.1 Data Mining and Retrieval of Data

The Cancer Genome Atlas (TCGA) intends to analyze and interpret the various molecular information of tumor types at their DNA, RNA, protein and epigenetic level. TCGA data are classified by the data type (for example, clinical mutations, gene expression) and data level which are submitted at the Data Coordinating Center (DCC) for public access (http://cancergrnome.nih.gov/). The privacy of the patient's information is also maintained during the accession of the TCGA's data. The selected data (RNA-seq) were retrieved from TCGA portal (https://tcga-data.nci.nih.gov/).

2.2 Express Profiling by Various Statistical Methods

The co-expression between the cancer genes and the driver genes was found by Spearman's method using R which is an interface for statistical computing and graphical representation. Simultaneously, different statistical methods were also applied on the retrieved data like Bonferroni correction, multiple testing correction. In Bonferroni correction, p value is adjusted for several dependent or independent statistical tests and multiple comparisons can be performed. The modified P value is then calculated by dividing the previous P value (α) by the number of comparisons being made. In this study, Bonferroni correction was performed and p ($p < 0.05$) values were adjusted using *R*. To reduce the chances of obtaining false-positive results (Type I errors), Bonferroni correction was done and multiple testing correction was performed in parallel to reduce Type II error.

2.3 Co-express Pathways Detection

The action of genes and their expression can be represented as pathway diagrams for metabolic or signaling cascades. The present study was arranged by using DAVID Bioinformatics (The Database for Annotation, Visualization and Integration Discovery). All tools in the DAVID Bioinformatics provide useful analysis of large lists of genes derived from genomic studies.

3 Results and Discussion

The cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC) data was downloaded from TCGA data portal of 310 patients. 15.2 GB of data were collected having I.D of 20,532 genes. These data are classified by data type and data level consisting of six files in the data archive. Code in R was written to retrieve data from the archive files.

We tried to find how the genes are correlated with each other, whether they are positively correlated, negatively correlated or are not correlated. The correlation values are between -1 and 1. The values between 0 and 1 specify that the genes are positively correlated; the values between 0 and -1 indicate the genes are negatively correlated and the value 0 indicates not correlated. Only the top 12 significant genes were selected based on the criterion of correlation (r) values. 6 genes having $r > 0.5$ and 6 genes having $r < -0.5$ were selected. The significantly co-expressed genes with the driver genes are represented by level plots. A scale of colors indicates the correlation value where 2 colors blue and yellow indicate that the genes are positively or negatively correlated, respectively. A total number of 12 co-express genes were identified as follows: TP53, PIK3CA, CDK12, ATM, PIK3R1, EP300,

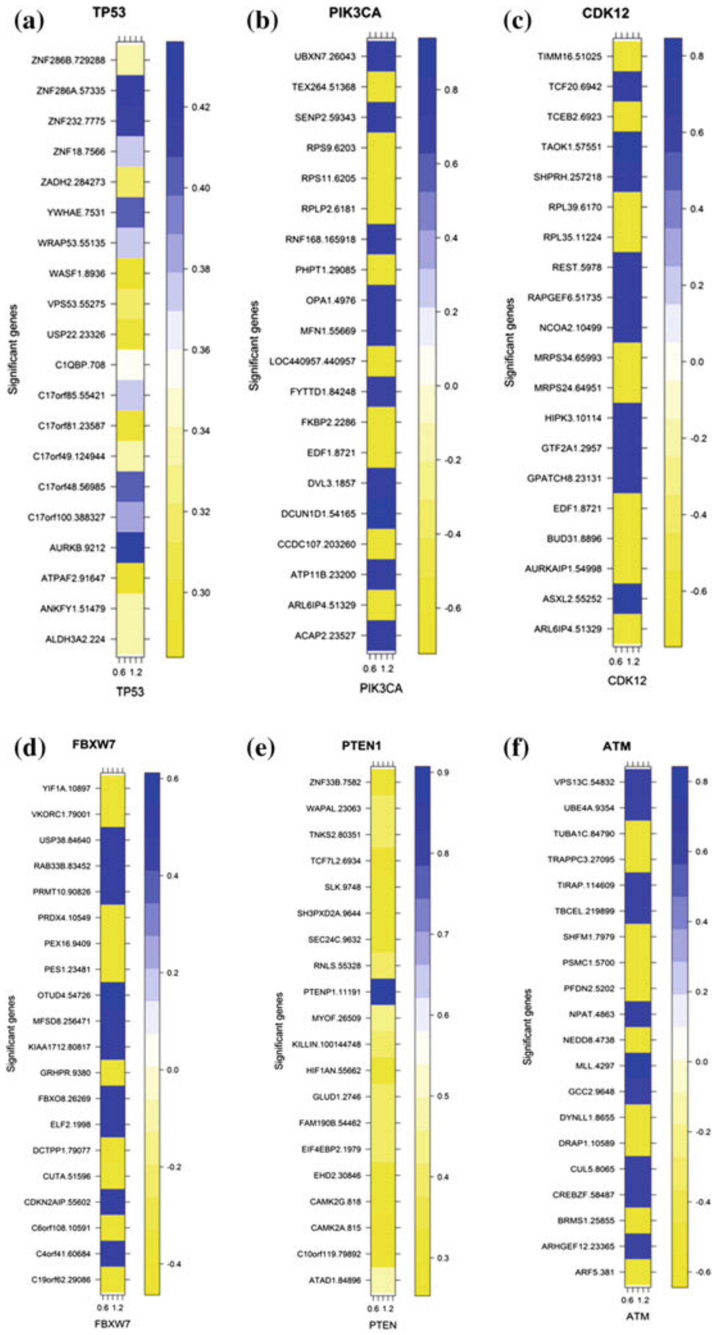
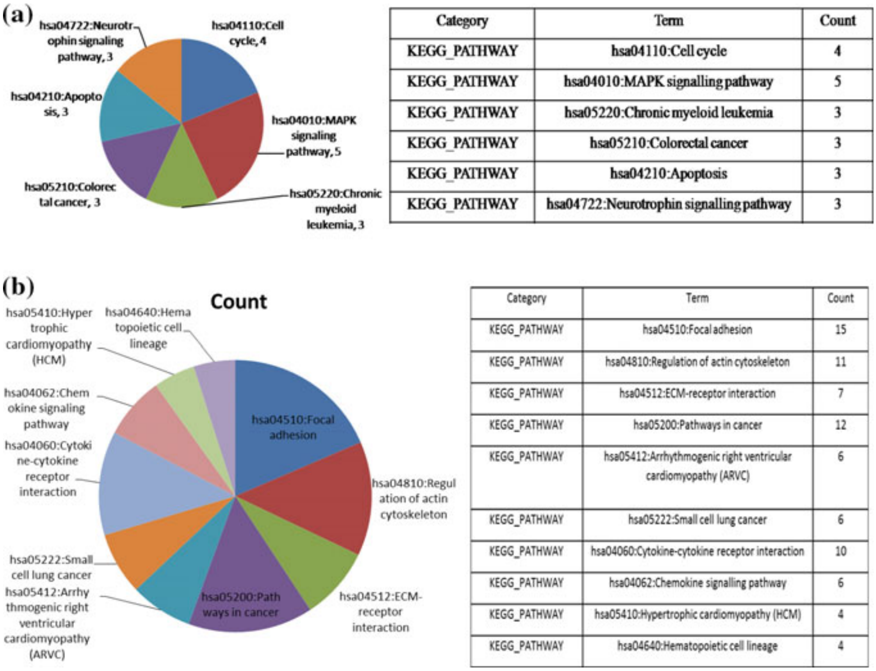


Fig. 1 A scale of colors indicates the correlation value between cancer genes and driver genes. The color plot represents (*a*)–(*f*) TP53, PIK3CA, CDK12, FBXW7, PTEN, ATM gene, respectively. The level plot shows positive (*blue*) and negative (*yellow*) correlation. Scale indicates the *r*-value

ARID1, KRAS, NFE2N2, MET, FBXW7 and PTEN. Out of 12, 6 co-express genes are discussed in this article showing the correlation by level plots (Fig. 1). Multiple testing corrections were done and most significant genes were found. Multiple correction method depends on how many genes are tested, the larger the gene list, the more stringent the correction will be.

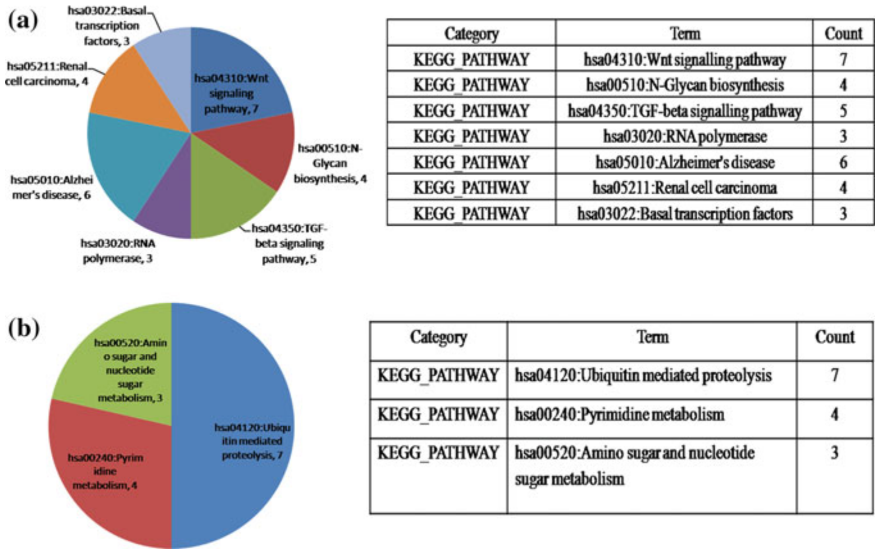
In cancer research, cancer genes [10] and their pathways provide an explanatory interpretation and understanding of cancer mechanisms [11]. Once the significant genes were identified, we found the co-express pathways. The pathways are found using the DAVID (Database for Annotation, Visualization and Integration Discovery) resource. DAVID, interpret the biological mechanisms with large gene lists. The results show (Figs. 2, 3 and 4) different pathways being affected by the co-express genes and the number of genes affecting a particular pathway. The pie charts (Figs. 2, 3 and 4) are divided into different regions and each region has a different color which specifies the different pathways and the number of significant genes playing role in those pathways.



Category	Term	Count
KEGG_PATHWAY	hsa04110:Cell cycle	4
KEGG_PATHWAY	hsa04010:MAPK signalling pathway	5
KEGG_PATHWAY	hsa05220:Chronic myeloid leukemia	3
KEGG_PATHWAY	hsa05210:Colorectal cancer	3
KEGG_PATHWAY	hsa04210:Apoptosis	3
KEGG_PATHWAY	hsa04722:Neurotrophin signalling pathway	3

Category	Term	Count
KEGG_PATHWAY	hsa04510:Focal adhesion	15
KEGG_PATHWAY	hsa04810:Regulation of actin cytoskeleton	11
KEGG_PATHWAY	hsa04512:ECM-receptor interaction	7
KEGG_PATHWAY	hsa05200:Pathways in cancer	12
KEGG_PATHWAY	hsa05412:Arrhythmogenic right ventricular cardiomyopathy (ARVC)	6
KEGG_PATHWAY	hsa05222:Small cell lung cancer	6
KEGG_PATHWAY	hsa04060:Cytokine-cytokine receptor interaction	10
KEGG_PATHWAY	hsa04062:Chemokine signalling pathway	6
KEGG_PATHWAY	hsa05410:Hypertrophic cardiomyopathy (HCM)	4
KEGG_PATHWAY	hsa04640:Hematopoietic cell lineage	4

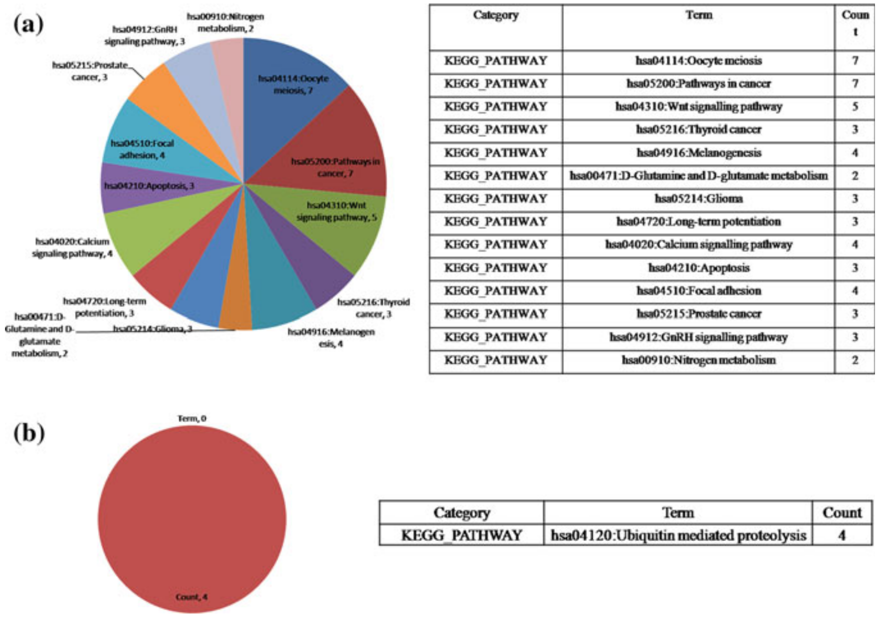
Fig. 2 A snapshot of pie chart and its corresponding table for two individual genes. The pie chart and the table represent the different pathways and the number of significant genes for **a** TP53 and **b** PIK3CA, respectively, playing roles in the pathway. The count shows the number of genes affecting a particular pathway



Category	Term	Count
KEGG_PATHWAY	hsa04310:Wnt signalling pathway	7
KEGG_PATHWAY	hsa00510:N-Glycan biosynthesis	4
KEGG_PATHWAY	hsa04350:TGF-beta signalling pathway	5
KEGG_PATHWAY	hsa03020:RNA polymerase	3
KEGG_PATHWAY	hsa05010:Alzheimer's disease	6
KEGG_PATHWAY	hsa05211:Renal cell carcinoma	4
KEGG_PATHWAY	hsa03022:Basal transcription factors	3

Category	Term	Count
KEGG_PATHWAY	hsa04120:Ubiquitin mediated proteolysis	7
KEGG_PATHWAY	hsa00240:Pyrimidine metabolism	4
KEGG_PATHWAY	hsa00520:Amino sugar and nucleotide sugar metabolism	3

Fig. 3 A snapshot of pie chart and its corresponding table for two individual genes. The pie chart and the table represent the different pathways and the number of significant genes for **a** CDK12 and **b** FBXW7, respectively, playing roles in the pathway. The count shows the number of genes affecting a particular pathway



Category	Term	Count
KEGG_PATHWAY	hsa04114:Oocyte meiosis	7
KEGG_PATHWAY	hsa05200:Pathways in cancer	7
KEGG_PATHWAY	hsa04310:Wnt signalling pathway	5
KEGG_PATHWAY	hsa05216:Thyroid cancer	3
KEGG_PATHWAY	hsa04916:Melanogenesis	4
KEGG_PATHWAY	hsa00471:D-Glutamine and D-glutamate metabolism	2
KEGG_PATHWAY	hsa05214:Glioma	3
KEGG_PATHWAY	hsa04720:Long-term potentiation	3
KEGG_PATHWAY	hsa04020:Calcium signalling pathway	4
KEGG_PATHWAY	hsa04210:Apoptosis	3
KEGG_PATHWAY	hsa04510:Focal adhesion	4
KEGG_PATHWAY	hsa05215:Prostate cancer	3
KEGG_PATHWAY	hsa04912:GnRH signalling pathway	3
KEGG_PATHWAY	hsa00910:Nitrogen metabolism	2

Category	Term	Count
KEGG_PATHWAY	hsa04120:Ubiquitin mediated proteolysis	4

Fig. 4 A snapshot of pie chart and its corresponding table for two individual genes. The pie chart and the table represent the different pathways and the number of significant genes for **a** PTEN and **b** ATM, respectively, playing roles in the pathway. The count shows the number of genes affecting a particular pathway

4 Conclusion

In cancer research, cancer genes and cancer pathways provide a holistic interpretation and understanding of cancer mechanisms. New techniques provide new evidence of gene expression or DNA structural alterations in the cervical cancer progression and new implications for cervical cancer research. The co-expression between statistically significant genes and driver genes was studied. Co-express pathways of co-express genes were studied; the genes affect different pathways. The number of genes affecting a particular pathway was also studied. The possible future scopes from this project are designing and implementation of methodology for the integration of pathway and high-throughput data and analysis of other cancers.

Acknowledgments The authors acknowledge BTISnet, Department of Biotechnology, Government of India, New Delhi, and TEQIP-II, BIT Mesra, for providing Infrastructure facility for the mentioned study.

References

1. Dunn, B.: Solving an age-old problem. Nature **483**, S2 (2012)
2. Rhee, I., Bachmann, K.E., Park, B.H., Jair, K., Yen, R.C., Schuebe, K.E., Cui, H., Feinberg, P.A., Lengauer, C., Kinzler, K.W., Baylin, S.B., Vogelstein, B.: DNMT1 and DNMT3b cooperate to silence genes in human cancer cells. Nature **416**, 552–556 (2002). doi:10.1038/416552a
3. Ojesina, A., et al.: Landscape of genomic alterations in cervical carcinomas. Nature **506**, 371 (2014)
4. Jemal, A., Siegel, R., Ward, E., et al.: Cancer statistics. CA Cancer J. Clin. **59**(4), 225–249 (2009)
5. Ferlay, J., Shin, H.R., Bray, F., Forman, D., Mathers, C., Parkin, D.M.: Estimates of worldwide burden of cancer in 2008: GLOBOCAN 2008. Int. J. Cancer (2010). doi:10.1002/ijc.25516
6. Walboomers, J.M.M., Jacobs, M.V., Manos, M., Bosch, F.X., Kummer, A., Shah, K.V., Snijders, P.U.F., Peto, J., Meijer, C.J.L.M., Muñoz, N.: Human Papilloma virus is a necessary cause of invasive cervical cancer worldwide. J. Pathology. **189**, 12–19 (1999)
7. Tang, F., Barbacioru, C., et al.: RNA-Seq. analysis to capture the transcriptome landscape of a single cell. Nat. Protoc. **5**, 517 (2010)
8. Kanehisa, M., Goto, S., et al.: KEGG for Representation and Analysis of Molecular Networks Involving Diseases and Drugs, 38: D356. Oxford University Press (2009)
9. Atak, Z.K., Gianfelici, V., Hulselmans, G., et al.: Comprehensive analysis of transcriptome variation uncovers known and novel driver events in T-cell acute lymphoblastic leukemia. PLoS Genet. **9**(12), e1003997 (2013)
10. Futreal, A.P., Coin, L., Marshall, M., Down, T., Hubbard, T., Wooster, R., Rahman, N., Stratton, M.R.: A census of human cancer genes. Nat. Rev. Cancer **4**, 177–183 (2004). doi:10.1038/nrc1299
11. Cairns, R.A., Harris, I.S., Mak, T.W.: Regulation of cancer cell metabolism. Nat. Rev. Cancer **11**, 85–95 (2011). doi:10.1038/nrc2981

Software Cost Estimation Using Cuckoo Search

Sweta Kumari and Shashank Pushkar

Abstract One of the important aspects of any software organization is to use models that can accurately estimate the software development effort. However, development of accurate estimation model is still a challenging issue for software engineering research community. This paper proposes a new model for software cost estimation that uses Cuckoo Search (CS) algorithm for finding the optimal parameter of the cost estimation model. The proposed model has been tested on NASA software project datasets. Experimental results show that the proposed model has improved the performance of the estimated effort with respect to MMRE (Mean Magnitude of Relative Error) and PRED (Prediction).

Keywords Software cost estimation • CS algorithm • COCOMO model • MMRE and PRED

1 Introduction

In modern days, software development organizations give so much importance to effective and efficient cost estimating techniques for their success. Cost estimation can be defined as the process of predicting the amount of time and persons required to develop a software system [1]. Accurate estimation of software cost is one of the most challenging tasks in software engineering as it is used for creating proper planning, budgeting and scheduling, efficient allocation of resources and monitors the progress of the project. Inaccurate estimation of software projects causes problems such as lowering the quality of the project, wastes the company's budget and can result in failure of the entire project [2]. The process of software cost estimation depends on many parameters in which size is considered to be the most

S. Kumari (✉) · S. Pushkar
Department of CSE, BIT, Mesra, Ranchi, India
e-mail: swetak44@gmail.com

S. Pushkar
e-mail: shashank_pushkar@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_17

important parameter and measured in thousands of delivered lines of code (KDLOC) [3, 4]. Now-a-days various models have been developed by many researchers for software cost estimation which are mostly based on meta-heuristics. Generally, they use these algorithms for finding the optimal parameters which are responsible for achieving the accuracy of the estimation process. In [5, 6], authors present cost estimation models using genetic algorithms and fuzzy logic. A computational intelligence technique was used to develop a model for software cost estimation [7]. Soft computing techniques were used in [8–11]. In [12, 13], the authors give the models for software cost estimation that is based on artificial neural network and neuro-fuzzy approaches.

Constructive cost model (COCOMO) is one of the traditionally known algorithmic software cost estimation model. It was proposed by Boehm [14] and based on the past experience of software projects. It is a collection of three models, namely Basic model, Intermediate model, and Semi-detached model. The basic COCOMO model uses program size for calculating efforts. It comes in the form of:-

$$\text{Effort} = a * (\text{Size})^{b} \tag{1}$$

Here, an effort is measured in person-month, Size is measured in kilo lines of code (KLOC) and a, b are constants. Generally, these constants are fixed for a specific project type. In this paper, a new model has been presented that is based on a novel meta-heuristic algorithm named CS for software cost estimation. The experiment done on a NASA software project dataset [15] has shown that the proposed model is improving the accuracy of estimation as compared to other existing models. The rest of the paper is organized as follows: Sect. 2 explains the methodology used in this work. Experiments and comparison are described in Sect. 3 and finally the conclusions are described in Sect. 4.

2 Methodology

2.1 CS Algorithm

CS is new population based algorithm proposed by Yang and Deb [16]. It is inspired by cuckoo’s aggressive breeding strategy in which they occupy other bird’s nests for breeding and the host bird’s starts taking care of cuckoo’s eggs. If a host bird finds that the eggs are not its own, it will either throw these alien eggs away or simply give away its nest and create a new nest somewhere else. Based on above cuckoo’s behavior, the CS algorithm follows three principal assumptions [16]:-

1. The cuckoo lays only one egg at a time in a randomly chosen nest. Here, each egg represents a possible solution.
2. The best nests with high quality eggs will be carried over to the next generations.
3. The numbers of available host nests are fixed and the egg laid by a cuckoo bird is discovered by the host bird with a probability p_a. Here, the host bird has two choices. It can either throw the egg away or it may abandon the nest, and create a new nest somewhere else.

The CS algorithm uses a search pattern called Lévy flight, generally followed by animals and birds for searching foods in a random or quasi random manner. Here, pattern of searching the next step is based on the current location and the probability of moving to the next location. According to Yang [17] and Gutowski [18], lévy flight is more efficient for searching than regular random walks or Brownian motions. The lévy flight is characterized by a variable step size punctuated by 90° turns.

2.2 Proposed Model

The proposed model attempts to provide better accuracy in the software cost estimation process by applying CS algorithm. Here, the main aim is to optimize the parameters a and b of cost estimation model shown by Eq. (1). The proposed model has been applied to a NASA software projects dataset [15] and the results obtained have found to be better estimating model in comparison to other recent estimation models like [5, 6, 15, 19] with respect to MMRE and PRED.

2.2.1 Fitness Function

Our objective is to minimize the MMRE (Mean Magnitude of Relative Error) between the estimated and actual efforts. It is a common evaluation criteria used for measuring the performance of models in the cost estimation process. The MMRE is calculated as:-

$$\text{MMRE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Actual Effort} - \text{Estimated Effort}}{\text{Actual Effort}}\right| \tag{2}$$

Here, n is the total number of projects.

2.2.2 Implementation of CS Algorithm for Software Cost Estimation

The CS algorithm implemented as follows:-

1. The initial population of nests with the size n is randomly generated. It is defined in the search space by the lower and upper boundaries.
2. To get the new nest or solution x^{t+1} for a cuckoo i, a lévy flight is used using the following equation:-

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \, \text{lévy}(\lambda) \tag{3}$$

   Where, $\alpha > 0$ represents the step size and t represents current generation. The ⊕ means an entry-wise multiplication. The random step length is drawn from a levy distribution which has an infinite variance with an infinite mean.

$$\text{lévy} \sim \mathrm{u} = t^{-\lambda}, \; 1 < \lambda \leq 3 \tag{4}$$

3. The fitness value is calculated and evaluated using Eq. (2).
4. A fraction of worst nests is discarded while preserving the best nest and again through Lévy flight, new nest by the same fraction are generated.
5. Calculate the fitness value of the new nest.
6. Rank all nests and the find best nest from the current ones.
7. If stop condition is satisfied, stop, if not, go to 2.

The parameter settings for the CS algorithm for finding the optimal parameter of cost estimation models are given in Table 1.

3 Experiments and Comparison

The data are taken from [15] for experimentation and comparison purpose. It is given in Table 2. The parameters obtained using the above methodology is a = 2.2448 and b = 0.8653. Now, this a and b are used for estimating effort for proposed model which is shown in Table 3.

Table 1 Parameter settings for the CS algorithm

Parameter	Value
Lower and upper bounds for a	1.00:4.00
Lower and upper bounds for b	0.01:2.00
Number of nests	25
Alpha	0.01
Discovery rate	0.25
Number of iterations	100

Table 2 NASA software projects data

Project No.	KDLOC	ME	Measured effort
1	90.2	30	115.8
2	46.2	20	96
3	46.5	19	79
4	54.5	20	90.8
5	31.1	35	39.6
6	67.5	29	98.4
7	12.8	26	18.9
8	10.5	34	10.3
9	21.5	31	28.5
10	3.1	26	7
11	4.2	19	9
12	7.8	31	7.3
13	2.1	28	5
14	5	29	8.4
15	78.6	35	98.7
16	9.7	27	15.6
17	12.5	27	23.9
18	100.8	34	138.3

3.1 Evaluation Criteria

The performance of the proposed model is evaluated here using MMRE and PRED. A detail description of MMRE can be found in the previous section. PRED can be defined as below:-

PRED (0.25):- It is defined as the percentage of predictions falling within 25 % of the actual known value. The PRED is calculated using the following equation:-

$$\text{PRED} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Actual Effort} - \text{Estimated Effort}}{\text{Actual Effort}}\right| \leq 0.25 \tag{5}$$

The experimental results are given in Table 3. The results of the proposed model are also compared with the results of existing models and it has been seen that the proposed model is able to provide closer estimated effort to the actual effort. Figure 1 shows the comparison between the estimated efforts of the proposed model to the measured efforts.

Table 4 shows that the MMRE and PRED value of various models. From the Table 4, it has been observed that the value of MMRE applying CS was lower than MMRE value of other existing models and the PRED value increases significantly. Figure 2 shows the performance comparisons of various models in terms of MMRE and PRED value.

Table 3 Estimated efforts of various models

Project No.	Measured effort	Bailey- basil effort	Doty Effort	Halstead effort	Alaa F. Sheta effort	Harish Model 1 effort	Prasad and Hari effort	proposed model effort
1	115.8	140.8202	589.3763	4454.6455	131.8988	126.3341	125.7302	110.4124
2	96	67.7741	292.5308	1632.9253	80.8741	74.2201	70.8350	61.8860
3	79	68.2434	294.5199	1648.8562	81.2577	74.6029	71.2293	62.2336
4	90.8	80.9297	347.7752	2092.1754	91.2576	84.6382	81.6179	71.3973
5	39.6	44.8482	193.2910	901.8700	60.5545	54.1846	50.4476	43.9403
6	98.4	102.1753	435.0834	2883.7593	106.7071	100.3293	98.0537	85.9160
7	18.9	19.5503	76.3028	238.1323	31.6425	26.7524	23.5607	20.3820
8	10.3	16.6662	62.0121	176.9242	27.3767	22.8547	19.8799	17.1717
9	28.5	31.1419	131.3271	518.3953	46.2312	40.4036	36.7575	31.9254
10	7	8.2121	17.2881	28.3822	11.2208	8.6649	6.9828	5.9752
11	9	9.3574	23.7593	44.7587	14.0103	11.0309	9.0602	7.7710
12	7.3	13.4096	45.4270	113.2779	22.0293	18.0445	15.4064	13.2772
13	5	7.2262	11.4989	15.8246	8.4404	6.3576	5.0000	4.2657
14	8.4	10.2221	28.5176	58.1378	15.9150	12.6711	10.5215	9.0364
15	98.7	120.8488	510.2686	3623.5742	119.2705	113.2379	111.7296	98.0137
16	15.6	15.6854	57.0744	157.0945	25.8356	21.4593	18.5737	16.0336
17	23.9	19.1691	74.4314	229.8097	31.0986	26.2527	23.0863	19.9680
18	138.3	159.4349	662.0863	5262.5246	143.0604	138.0010	138.3002	121.5548

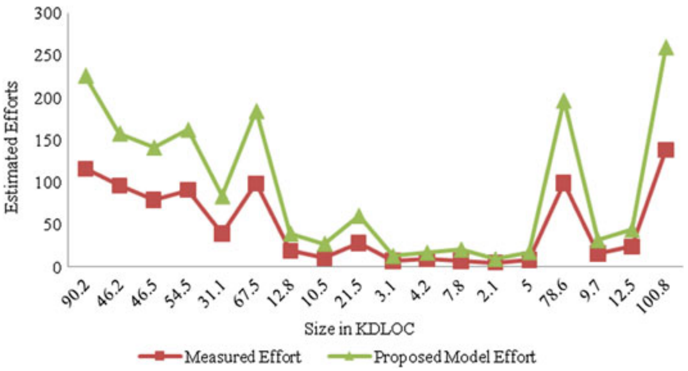


Fig. 1 Measured effort versus estimated efforts of proposed model

Table 4 Performance comparisons of various models

Models	MMRE	Prediction (0.25)
Bailey-Basil	0.2202	0.7777
Doty	3.0789	0
Halstead	17.2511	0
Alaa F. Sheta	0.5477	0.3889
Harish model 1	0.3455	0.5555
Prasad and Hari	0.2233	0.6667
Proposed	0.1986	0.8333

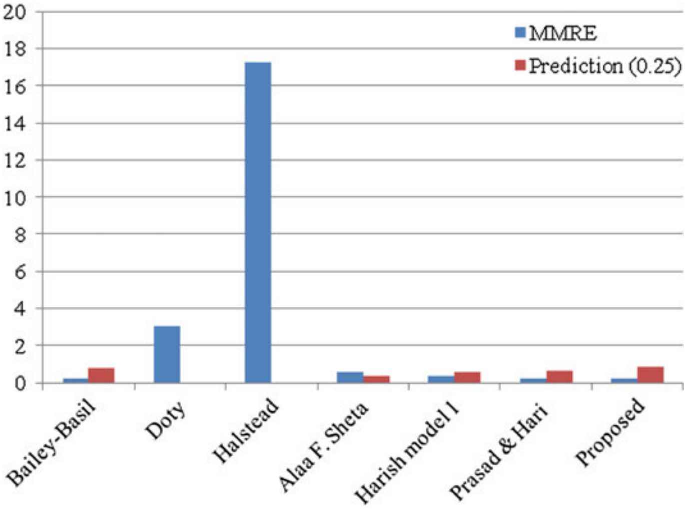


Fig. 2 Performance comparisons of various models in terms of MMRE and PRED value

4 Conclusions

Here, we propose a new model for software cost estimation that is based on a new meta-heuristic algorithm called Cuckoo Search. This algorithm has been used to discover the optimal parameters of the cost estimation model. The experimental results obtained on a standard dataset [15] have shown the superiority of the proposed model over the currently used estimation models. The future direction of the research can be to incorporate multiple evaluation estimation criteria to optimize the parameters and to investigate the suitability of the procedure for the accurate cost estimation.

References

1. Kumari, S., Pushkar, S.: Performance analysis of the software cost estimation methods: a review. Int. J. Adv. Res. Comput. Sci. Soft. Eng. **3**, 229–238 (2013)
2. Kumari, S., Pushkar, S.: A genetic algorithm approach for multi-criteria project selection for analogy-based software cost estimation. Int. Conf. Comput. Intell. Data Min. **3**, 13–24 (2014)
3. Kumari, S., Ali, M., Pushkar, S.: Fuzzy clustering and optimization model for software cost estimation. Int. J. Eng. Technol. **6**(6), 2531–2545 (2015)
4. Kumari, S., Pushkar, S.: Comparison and analysis of different software cost estimation methods. Int. J. Adv. Comput. Sci. Appl. **4**(1), 153–157 (2013)
5. Sheta, A.F.: Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects. J. Comput. Sci. **2** (2), 118–123 (2006). ISSN 1549-36362006
6. Anish, M., Kamal, P., Harish, M.: Software cost estimation using fuzzy logic. ACM SIGSOFT Soft. Eng. Notes **35**(1), 1–7 (2010)
7. Pahariya, J.S., Ravi, V., Carr, M.: Software cost estimation using computational intelligence techniques. In: IEEE Transaction, pp. 849–854. IEEE (2009). doi:10.1109/NABIC.2009.5393534
8. Sheta, A., Rine, D., Ayesh, A.: Development of software effort and schedule estimation models using soft computing techniques. In: 2008 IEEE Congress on Evolutionary Computation (CEC 2008) (2008). doi:10.1109/CEC.2008.4630961
9. Attarzadeh, I., Ow, S.H.: Soft computing approach for software cost estimation. Int. J. Soft. Eng. IJSE **3**(1), 1–10 (2010)
10. Sandhu, P.S., Bassi, P., Brar, A.S.: Software Effort Estimation Using Soft Computing Techniques, pp. 488–491 (2008)
11. Gonsalves, T., Ito, A., Kawabata, R., Itoh, K.: Swarm Intelligence in the Optimization of Software Development Project Schedulepp, pp. 587–592 (2008). doi:10.1109/COMPSAC.2008.179
12. Attarzadeh, I., Mehranzadeh, A., Barati, A.: Proposing an enhanced artificial neural network prediction model to improve the accuracy in software effort estimation. In: Computational Intelligence, Communication Systems and Networks Conference, pp. 167–172 (2012)
13. Huang, X., Ho, D., Ren, J., Capretz, L.F.: Improving the COCOMO model using a neuro-fuzzy approach. Appl. Soft. Comput. (2007) **7**, 29–40 (2005). doi:10.1016/j.asoc.2005.06.007. Elsevier
14. Boehm, B.: Software Engineering Economics. Englewood Cliffs, NJ, Prentice-Hall (1981)
15. Bailey, J.W., Basili, V.R.: A meta model for software development resource expenditure. In: Proceedings of the International Conference on Software Engineering, pp. 107–115 (1981)

16. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights, pp. 210–214. Proceeings of World Congress on Nature and Biologically Inspired Computing, India (2009)
17. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J. Math. Model. Numer. Optim. **1**(4), 330–343 (2010)
18. Gutowski, M.: Lévy flights as on underlying mechanism for global optimization algorithms, vol. 8 (2011)
19. PVGD, P.R., Hari, C.V.: Software Effort Estimation Using Particle Swarm Optimization with Interia Weight, vol. 2(4), pp. 87–96 (2011)

The Insects of Innovative Computational Intelligence

Sweta Srivastava and Sudip Kumar Sahana

Abstract The desirable merits of intelligent algorithm and the initial success in many domains have inspired researchers to work towards advancement of these techniques. A major plunge in algorithmic development to solve increasingly complex problems turned out as breakthrough towards the development of innovative computational intelligence (CI) techniques. Here this paper discusses innovative computational intelligence techniques that are inspired by insect families. These techniques utilize the skills of intelligent agents for required problem.

Keywords Swarm intelligence • Computational intelligence • Bee colony • Mosquito host seeking • Wasp colony • Insect colony • Termite • Super-bug

1 Introduction

The term “computational intelligence’ was coined by Bezdek [1] based on intelligent behavior of computer inspired from nature. CI [2] is the study of design of intelligent agents. The agents can be an inspiration from insects, worms, animals, airplane, human, organizations, society and many more. It utilizes the skills of intelligent agents for required results as illustrated in Fig. 1. CI had opened many brand new dimensions for scientific research in past two decades.

S. Srivastava (✉) · S.K. Sahana
Department of Computer Science, BIT, Mesra, Ranchi, India
e-mail: ssrivastava.rnc09@gmail.com

S.K. Sahana
e-mail: sudipsahana@bitmesra.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_18

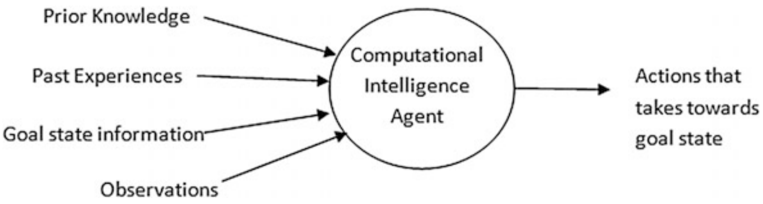

Prior Knowledge
Past Experiences
Goal state information
Observations
Computational Intelligence Agent
Actions that takes towards goal state

Fig. 1 Computational intelligence agents and driving skills

CI can be broadly classified into two categories [2]—traditional and innovative. Traditional CI [3, 4] mostly concentrates on artificial neural network (ANN), fuzzy logic (FL), evolutionary algorithms (EA) [e.g., genetic algorithm (GA), genetic programming (GP), evolutionary programming (EP), and evolutionary strategy (ES)], simulated annealing (SA), artificial immune systems (AIS), Tabu search (TS), and two variants of swarm intelligence (SI), i.e., ant colony optimization (ACO) and particle swarm optimization (PSO).

Innovative CI are highly developed and refined than CI. It can be grouped into several families depending on its functionality and the source of inspiration. It can be inspired by physics [5] like Big Bang-Big Crunch (BBC), Central Force Optimization (CFO), Particle Collision Algorithm (PCA). There are several techniques inspired by chemistry [5] like Artificial Chemical Process (ACP), Chemical Reaction Algorithm (CRA), and Gases Brownian Motion Optimization (GBMO). Techniques inspired by mathematics are Base Optimization Algorithm (BOA), Matheuristics.

There are several biological inspirations [5] for innovative CI. These can be further grouped into multiple classes like animals, plants, birds, tribes etc. depending upon its nature. There are many algorithms bagged into each categories like techniques inspired from animal includes Cat Swarm Optimization (CSO), Monkey Search, Wolf Pack Search (WPS). Techniques like Cuckoo Optimization Algorithm (COA), Dove Swarm Optimization (DSO), and Migrating Birds Optimization (MBO) are inspired by birds. Bacterial Foraging Algorithm (BFA), Amoeboid Organism Algorithm (AOA) and several others are inspired by microorganisms. Artificial Fish Swarm Algorithm (AFSA), Fish School Search (FSS), Shark-Search Algorithm (SSA) and many more techniques are based on aquatic animals. Frog Calling Algorithm (FCA). Shuffled Frog Leaping Algorithm (SFLA) are inspired by amphibians. Plants inspired algorithms include Paddy Field Algorithm (PFA), Invasive Weed Optimization (IWO). Saplings Growing Up Algorithm (SGUA). There are quite a lot of other categories and several techniques under them which are beyond the scope of this paper. A family tree of CI is shown in Fig. 2.

The prime focus of this paper is CI techniques that are inspired by insect families which will be discussed in further sections.

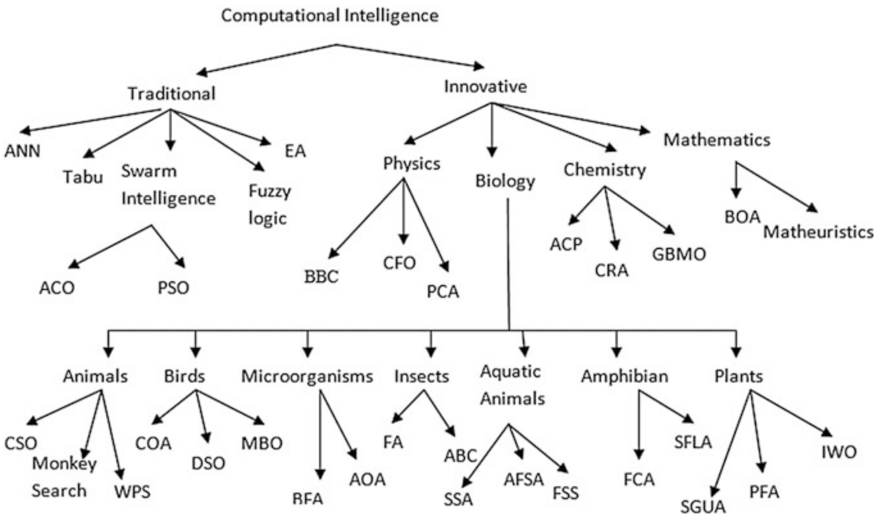


Fig. 2 Family of computational intelligence

2 Insect Family

The word insect comes from a Latin word ‘insectum’ [6]. Insects are among the most assorted set of animals in the world with more than a million described species. Their body is divided into three-parts (head, thorax and abdomen). They also have three pairs of jointed legs, compound eyes and one pair of antennae. These small living beings had always inspired researchers with their interesting organizational and food foraging behaviors. Figure 3 shows the growth of insect inspired techniques in last two decades. It can be clearly interpreted from the techniques discussed here that gradually the growth in innovative techniques inspired from insects are stepping up.

3 Insect Behavior

Insects habitually lives into solitude but some such as bees, ants, termites etc. are found in large and well organized colonies. They communicate [7–12] with each other in a verity of ways like sound, light, smell, or by rubbing wings. The prime conduct that attracted researchers towards insets are their ability to communicate with each other in a verity of ways like sound, light, smell, waggle dance or by rubbing wings.

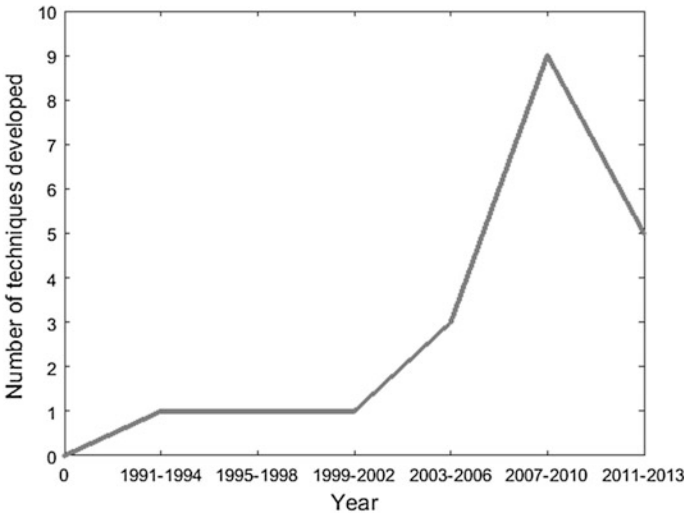


Fig. 3 Growth of insect inspired techniques in last two decades

Food foraging [7, 12–14], is another major themselves to explore promising food sources, inform the colony members and exploit it in an optimal manner.

The selection of partner for mating [15] of some insects can also be framed by computational intelligence scientists.

Another major attributes found in insects are division [16, 17] of the colony depending on its size and resource availability. In families like bees, they can have only one queen.

4 Insect Oriented Algorithm Model

There are several CI techniques developed by researchers inspired by attributes of insects such as self organization, food foraging, ability to adapt according to changing environment, and so forth. These algorithms are discussed in Table 1.

In the above section we have seen that several insect families are well thought-out for solving complex computational problems. Different character of features of the insect brings out a new methodology of solution. Most of the problems are based on selection of best solution state and discard the non optimal solutions (WSO, VBA etc.) and some with very few computational functions (ABC, HBMO etc.) and thus comparatively simpler to work. Several algorithms like Termite-hill Algorithm, Cockroach Swarm Optimization and bee hive algorithms are implemented for some specific problem.

Table 1 Insect oriented algorithmic model

Algorithm	Agents	Inspiration	Implementation and Remark
Artificial Bee Colony [7], [16]	Employed bees, Unemployed bees (scouts bees, onlooker bees)	Foraging behavior of the honey bees	Implementation: Structural optimization, [18], Traffic optimization [19], Continuous Optimization, Network routing [20] Employ fewer control parameters than DE, GA and PSO Better convergence speed than GA and DE. Employ a greedy selection process between the candidate and the parent solution Unlike DE and GA, ABC avoids premature convergence Solves multimodal and multidimensional optimization problem
BeeHive [8, 21]	Short distance bee agent, long distance bee agents	Dance language and foraging behavior of honey bee	Implementation: Network routing [22] Implemented for Japanese internet backbone scenario
Bees Life Algorithm [9]	Queen bees, drone bees and newly born bumblebees	Reproduction and foraging behavior of bees	Implementation: Vehicular ad hoc network (VANET) problem [9] Found to be more efficient than GA and HBMO Uses a low number of nodes Efficient, less complex
Cockroach Swarm Optimization [23]	Cockroach Swarm	Chase swarming, dispersing and ruthless behavior of cockroach	Implementation: Vehicle routing problem [23] Higher optimal rate and shorter time than PSO
Firefly Algorithm [24]	Firefly	Social behavior of firefly and the phenomenon of bioluminescent communication.	Implementation: Scheduling and TSP [24, 25], Feature selection [26] Decreasing random step size. Outperform Particle Swarm. Optimization [24] in 11 bench mark problem

(continued)

Table 1 (continued)

Algorithm	Agents	Inspiration	Implementation and Remark
Fruit Fly Optimization Algorithm [14]	Fruit fly	Oshresis and sensitive vision of fruit flies for food foraging	Implementation: Traffic flow control [27], Multidimensional knapsack problem [28], Control Optimization [29], Clustering [30] Ranking of solution is needed Solves combinatorial optimization problems Computational process is simple and easy implementation Tested for maximum and minimum value problem in [14]
Glowworm Swarm Optimization [10]	Glowworm	Luciferin-update phase and movement of glowworm	Implementation: Deployment of wireless sensor nodes [31], Sensor Deployment [32] Compared with PSO, artificial fish swarm algorithm (AFSA), and AFSA with chaos (CAFSA), the experimental results [33] showed that MNGSO is an effective global algorithm for finding optimal results
Honeybee Social Foraging [34]	Queen, drone, workers, brood	Foraging behavior of bees	Implementation: VANET [9] Ranking of solution. It outperforms GA and HBMO [9]
Honeybees Mating Optimization [15, 35]	Queen bee, male honey bees (drones), neuters/underdeveloped honey bees	Probabilistic mating of drones with queen bee	Implementation: Vehicle routing problem [36] Queen represents best solution Weaker solutions are replaced using fitness function In comparison to several nature inspired metaheuristic method it is one of the best method [5]

(continued)

Table 1 (continued)

Algorithm	Agents	Inspiration	Implementation and Remark
Mosquito Host-Seeking Algorithm [13]	Artificial female mosquitoes	Host seeking behavior of mosquitoes by smelling odor like CO_2	Implementation: Travelling sales person problem [13] Performs well and easy to jump into local minima. Easy to adapt into wide range of travelling sales person problem
Superbug Algorithm [37]	Bacteria	Evolution of bacteria in a culture	Implementation: Large scale optimization problems such as scheduling, transportation and assignment problems. (FMS) [37] Gives better results in comparison to Sliding Time Window (STW), Abdelmaguid Genetic Algorithm (AGA), Ulusoy Genetic Algorithm (UGA), Proposed Genetic Algorithm (PGA) and Sheep Flock Heredity Algorithm (SFHA) [38]
Termite-hill Algorithm [17]	Termite	Behavior of termite to allow simulation in restricted environment	Implementation: Wireless sensor networks routing problem [17] Originally proposed for dealing with wireless sensor networks routing problem Competitively efficient routing algorithm in terms of energy consumption, throughput, and network lifetime
Virtual Bees Algorithm [39]	Bees associated with memory bank	Foraging and communication of bees with memory bank	Implementation: Controller design problem [40], Damping control system [41] Ranking of solution Ability to solve multi-objective problems involving many local minima

(continued)

Table 1 (continued)

Algorithm	Agents	Inspiration	Implementation and Remark
Wasp Swarm Optimization [6]	Wasp	Importance of individual wasp in the colony and resource allocation to them	Implementation: Vehicle routing problem [42], Image processing [43], Scheduling optimization [20], Image processing [43] Tournament selection process for resource allocation amongst the wasp Perform well for problems with more dynamic properties

5 Conclusion

Here in this paper we have discussed some of the computational intelligence algorithm inspired by insect family. A comparative study was carried out and merits demerits and application areas are highlighted. The source of inspiration in not limited to these only. There are several other notable techniques as mentioned in this paper. As per the nature of the problem most appropriate algorithm can be selected. Another interesting observation is that till the year 2007–08 most of the scientists studied bees and wasps' behavior but gradually different species like cockroaches, termites, firefly, fruit-fly and spiders also came into picture with their unique attributes. The major fascination of these techniques is flexibility in the application part. The algorithms can be merged with other techniques and modified as desired by the researcher so that better results can be obtained.

References

1. Bezdek, J.C.: What is computational intelligence? In: Computational Intelligence Imitating Life, pp. 1–12. IEEE Press, New York (1994)
2. Poole, D., Mackworth, A., Goebel, R.: Computational Intelligence-A Logical Approach. Oxford University Press (1998)
3. Karci, A., Alatas, B.: Thinking capability of saplings growing up algorithm. In: Intelligent Data Engineering and Automated Learning (IDEAL 2006). LNCS, vol. 4224, pp. 386–393. Springer, Berlin (2006)
4. Andries, P.: Engelbrecht: Computational intelligence- An introduction. John Wiley & Sons Ltd. ISBN: 978-0-470-03561-0 (HB) (2007)
5. Xing, B., Gao, W.-J.: Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer International Publishing, Switzerland (2014). ISSN: 1868-4394. doi:10.1007/978-3-319-03404-1

6. Chapman, A.D.: Numbers of Living Species in Australia and the World. Australian Biological Resources Study, Canberra. ISBN: 978-0-642-56850-2 (2006)
7. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. **214** (2009)
8. Wedde, H.F., Farooq, M.: A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks. J. Syst. Architect. **52**, 461–484 (2006)
9. Bitam, S., Mellouk, A.: Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks. J. Netw. Comput. Appl. **36**, 981–991 (2013)
10. Krishnanand, K.N., Amruth, P., Guruprasad, M.H., Bidargaddi, S.V., Ghose, D.: Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 958–963. IEEE, Orlando, Florida, USA (2006)
11. Krishnanand, K.N., Ghose, D.: Detection of multiple source locations using glowworm metaphor with applications to collective robotics. In: IEEE Swarm Intelligence Symposium (SIS), pp. 84–91. IEEE (2005)
12. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm'. Appl. Soft Comput. 687–697 (2007)
13. Feng, X., Lau, F.C.M., Gao, D.: A new bio-inspired approach to the traveling salesman problem. In: Zhou, J. (dd.) Complex 2009, Part II, LNICST, vol. 5, pp. 1310–1321. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (2009)
14. Pan, W.-T:. A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl. Based Syst. **26**, 69–74 (2012)
15. Abbass, H.A.: MBO: marriage in honey bees optimization. A haplometrosis polygynous swarming approach. In: IEEE Proceedings of the Congress on Evolutionary Computation, pp. 207–214. Seoul, South Korea (2001a)
16. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**, 459–471 (2007)
17. Zungeru, A.M., Ang, L.-M., Seng, K.P.: Termite-hill: performance optimized swarm intelligence based routing algorithm for wireless sensor networks. J. Netw. Comput. Appl. **35**, 1901–1917 (2012)
18. Hadidi, A., Azad, S.K., Azad, S.K.: Structural optimization using artificial bee colony algorithm, In: 2nd International Conference on Engineering Optimization, 2010, Lisbon, Portugal. Accessed 6–9 Sept 2010
19. Baskan, O., Dell'Orco, M.: Artificial bee colony algorithm for continuous network design problem with link capacity expansions'. In: 10th International Congress on Advances in Civil Engineering, Middle East Technical University, pp. 17–19. Ankara, Turkey (2012)
20. Karaboga, D., Okdem, S., Ozturk, C.: Cluster based wireless sensor network routing using artificial bee colony algorithm. Wireless Netw. **18**(7), 847–860 (2012). ISSN: 1022-0038
21. Farooq, M.: From the wisdom of the hive to intelligent routing in telecommunication networks: a step towards intelligent network management through natural engineering, Unpublished doctoral thesis, Universität Dortmund (2006)
22. Farooq, M.: Bee-Inspired Protocol Engineering: From Nature to Networks. Springer, Berlin, Heidelberg ISBN: 978-3-540-85953-6 (2009)
23. Chen, Z., Tang, H.: Cockroach swarm optimization. In: 2nd International Conference on Computer Engineering and Technology (ICCET), pp. 652–655. IEEE (2010)
24. Łukasik, S., Zak, S.: Firefly algorithm for continuous constrained optimization tasks. In: Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. LNCS, vol. 5796, pp. 97–106. Springer, Berlin (2009)
25. Jati, G.K., Suyanto, S.: Evolutionary discrete firefly algorithm for travelling salesman problem, ICAIS2011. Lecture Notes in Artificial Intelligence (LNAI 6943), pp. 393–403 (2011)
26. Banati, H., Bajaj, M.: Firefly based feature selection approach. Int. J. Comput. Sci. Issues **8** (2), 473–480 (2011)

27. Zhu, W., Li, N., Shi, C., Chen, B: SVR based on FOA and its application in traffic flow prediction. Open J. Transp. Technol. **2**, 6–9 (2013)
28. Wang, L., Zheng, X.-L., Wang, S.-Y.: A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. Knowl. Based Syst. **48**, 17–23 (2013)
29. Liu, Y., Wang, X., Li, Y.: A modified fruit-fly optimization algorithm aided PID controller designing. In: IEEE 10th World Congress on Intelligent Control and Automation, pp. 233–238. Beijing, China (2012)
30. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study, swarm and evolutionary computation, June (2011)
31. Pradhan, P.M., Panda, G.: Connectivity constrained wireless sensor deployment using multi objective evolutionary algorithms and fuzzy decision making. Ad Hoc Netw. **10**, 1134–1145 (2012)
32. Liao, W.-H., Kao, Y., Li, Y.-S.: A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. Expert Syst. Appl. **38**, 12180–12188 (2011)
33. Huang, K., Zhou, Y., Wang, Y.: Niching glowworm swarm optimization algorithm with mating behavior. J. Inf. Comput. Sci. **8**, 4175–4184 (2011)
34. Quijano, N., Passino, K.M.: Honey bee social foraging algorithms for resource allocation: theory and application. Eng. Appl. Artif. Intell. **23**, 845–861 (2010)
35. Abbass, H.A.: A monogenous MBO approach to satisfiability. In: Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA). Las Vegas, NV, USA (2001b)
36. Marinakis, Y., Marinaki, M., Dounias, G.: Honey bees mating optimization algorithm for the vehicle routing problem. Stud. Comput. Intell. (SCI) **129**, 139–148 (2008)
37. Anandaraman, C., Sankar, A.V.M., Natarajan, R.: A new evolutionary algorithm based on bacterial evolution and its applications for scheduling a flexible manufacturing system. Jurnal Teknik Industri **14**, 1–12 (2012)
38. Subbaiah, K.V., Rao, M.N., Rao, K.N.: Scheduling of AGVs and machines in FMS with makespan criteria using sheep flock heredity algorithm. Int. J. Phys. Sci. **4**(2), 139–148 (2007)
39. Yang, X.S.: Engineering optimizations via nature-inspired virtual bee algorithms. In: Mira, J., Álvarez, J.R. (eds.) Artificial Intelligence and Knowledge Engineering Applications. A Bioinspired Approach. Springer, Berlin, Heidelberg (2005)
40. Wang, D.Z., Zhang, J.S., Wan, F., Zhu, L: A dynamic task scheduling algorithm in grid environment. In: 5th WSEAS International Conference on Telecommunications and Informatics, pp 273–275. Istanbul, Turkey (2006)
41. Khan, L., Ullah, I., Saeed, T., Lo, K.L.: Virtual bees algorithm based design of damping control system for TCSC. Aust. J. Basic Appl. Sci. **4**, 1–18 (2010)
42. Song, J., Hu, J., Tian, Y., Xu, Y.: Re-optimization in dynamic vehicle routing problem based on wasp-like agent strategy. In Proceedings of 8th International Conference on Intelligent Transportation Systems, pp. 688–693. Vienna, Austria (2005)
43. Fan, H., Zhong, Y.: A rough set approach to feature selection based on wasp swarm optimization. J. Comput. Inf. Syst. **8**, 1037–1045 (2012)

Part IV
Applications in Cloud Computing

Effect of VM Selection Heuristics on Energy Consumption and SLAs During VM Migrations in Cloud Data Centers

Rashmi Rai, G. Sahoo and S. Mehfuz

Abstract The virtual machine (VM) provisioning in cloud computing offers a good possibility for energy and cost saving, given the dynamic nature of the cloud environment. However, the commitment of giving the best possible quality of service to end users often leads to the requirement in dealing with the energy and performance tradeoff. In this work, we have proposed and evaluated three different virtual machine selection policies (MedMT, MaxUT and HP) to achieve a better performance as compared with the existing state of art algorithms. The proposed policies are evaluated through simulation on large-scale workload data conducted over a period of 7 days in a series of experiments. The results clearly indicate how the virtual machine selection algorithms can improve upon the energy consumption by data centers as well as the overall reduction in service level agreements (SLAs), thus reducing the cost significantly.

Keywords Virtual machine · Migration · Consolidation · Cloud computing · Data center

1 Introduction

With the advancements in computing power and the rapid increase in computing services being delivered as a utility to the consumer, there is a paradigm shift towards cloud computing technologies [1–3]. Big and small organizations, businesses as well as individual users have started relying on cloud services instead of building and managing their own data centers for fetching the required services. As a result, of which there has been sharp increase in the number of large-scale data centers across globe. For example, there are more than 454,000 servers for Amazon

R. Rai (✉) · G. Sahoo
Birla Institute of Technology, Mesra, Ranchi, India
e-mail: rashmirrai@gmail.com

S. Mehfuz
Jamia Millia Islamia, Delhi, India

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_19

EC2 and it is steadily increasing every year [4]. The number of datacenters is growing at a steady pace as per the recent forecast by CISCO and the cloud workload will be almost triple from 2013 to 2018. However, the work load in traditional data centers will decline as per the survey due to increasing virtualization in cloud environments [5]. Virtualization of resources is one of the prime factors which is responsible for the growing number of workload migration from the traditional data centers to the cloud data centers. The workloads to non-virtualized cloud server's ratio will grow from 6.5 in 2012 to nearly 16.7 by 2017. In comparison, the ratio of workloads to non-virtualized traditional data center servers will grow from 1.7 in 2012 to 2.3 in 2017. These large-scale data centers consume humongous amount of energy leading to huge operating costs and also contribute significantly towards the global CO_2 emission. According to [6], the total energy consumption by data centers will persist to grow at a fast pace until unless some sophisticated energy-efficient measures are deployed. Thus, reducing the energy usage in cloud data centers has become a prime concern across the globe, both for the sake of cloud providers benefit and the greener environment.

While the problem of energy consumption and CO_2 is on rise, virtualization technique continues to dominate the cloud data centers, providing the ability to consolidate VMs between physical nodes. This provides the dynamic VM consolidation an opportunity to utilize the minimum physical nodes. This results in turning off the idle nodes for energy saving, and possibly some SLA violation. Therefore, reducing the energy consumption using VM consolidation, while also managing fewer SLA violations, becomes a challenge. In this work, we have proposed three novel heuristics for selecting the VM for migration.

2 Related Work

Building better power efficient virtual data centers, while keeping the users happy with guaranteed SLAs, has already gained momentum and become a hot topic for research. In this context, a lot of methods and algorithms have been proposed and developed for dynamic virtual machine consolidation problem. Some of the pioneer work that has been done in the last 5 years (2010–2015) is what we have tried to summarized in this work. To the best of our understanding, the work done in [7–12] has shown excellent results for energy and SLAs performance tradeoffs. They have used the distributed approach for solving the dynamic virtual machine consolidation problem. The problem of VM consolidation has been divided into four sub-problems, namely (1) Host Underload Detection, (2) Host Overload Detection, (3) VM Selection (4) VM Migration and for each sub-problem they have proposed a number of heuristics. The evaluation and performance testing of these algorithms were done by the simulation tool CloudSim developed by them. They have evaluated their work from the real data set from large-scale workload traces. Later on, Cao [13] redesigned energy-aware heuristic framework for VM consolidation to achieve a better energy-performance tradeoff and the result guarantees 21–34 %

decrease in energy consumption, 84–92 % decrease in SLA violation, 87–94 % decrease in energy-performance metric, and 63 % decrease in execution time. The latest findings by Ehsan Arianya [14] show up to 46 %, 99 % and 95 % reduction in energy consumption, SLA violation, and number of VM migrations, respectively, as compared with previous work. They have proposed a multi-criteria decision-making method for detecting underloaded hosts and placing the VMs. Another latest survey done by Raja Wasim Ahmad [15] presents an elaborate discussion on the critical facet of virtual machine migration and dynamic server consolidation.

Based on the existing literature in the last few years, several conclusions can be made. Firstly, we can deduce that the dynamic VM consolidation problem is evolving rapidly with quite promising approaches and findings; however, there is still lot needs to be done in terms of energy and SLA tradeoff given the ever dynamic nature of cloud environment. Secondly, none of the work discusses the effect of different VM selection algorithms on the energy consumption and SLA violation. It can also be noted that VM selected for migration plays a crucial role as the number of VM migrations hugely impacts the energy consumption. Therefore, in this regard, we have tried to analyze and propose the effect of various VM selection algorithms keeping other algorithms constant in the overall consolidation problem solving.

3 VM Selection Policies

According to the work done by Beloglazov [7, 8], the Minimum Migration Time (MMT) was chosen as the most efficient VM selection policy which chooses a VM with the least migration time. However, this may not be able to detect the hotspot and might call for numerous migrations. Migration process also increases the task completion time because of the downtime which is produced at the source as well as the destination. Additionally, because of the excessive load transfer through the network VM migrations also aggravate energy consumption. This creates a hotspot situation and increases the number of SLA violations. The goal is to minimize the total number of VM migrations and eventually decrease the SLA violation. Therefore, this section proposes few VM selection policies and compares them to existing state of art algorithms.

3.1 Median Migration Time (MedMT) Policy

Once the hotspot or the overloaded host is detected, the next step would be to select a VM for migration failing which would cause degradation in the overall performance. In this regard, this policy aims to find the migration time of all candidate VMs on the hotspot and the median migration time is calculated. Eventually, the VM with least median migration time is selected for migration. This policy is an improvement upon the MMT (minimum migration time) heuristic proposed by Anton B. [7, 8] as described below. The migration time of the VM is calculated as the amount of RAM utilized by the VM divided by the unused network bandwidth available for the host j.

Where RAMu (a) is the amount of RAM currently utilized by the VM a; and NETj is the spare network bandwidth available for the host j.

3.2 Maximum Utilization (MaxUT) Policy

This maximum utilization policy chooses the VM for migration which has the largest possible CPU usage. Selecting this VM can be expected to minimize the number of migrations. We have used the finite discrete Markov chain model for predicting the CPU utilization. A finite Markov chain is a discrete stochastic process with a finite number of states defined at integer values of time n = 0, 1. A discrete system is represented by a set S of "states" and transitions between the states and S are referred to as the state space [16]. A discrete stochastic process is a discrete system in which transitions occur randomly according to some probability distribution. The finite Markov chain has an initial probability distribution vector and a transition probability matrix. The initial probability distribution vector V is an n-dimensional vector that holds the probability distribution over the states (P (s1)… P(sn)) where P(si) is the probability of each state. However, the transition probabilities matrix is an n X n matrix T = (Pij) where Pij is the probability of moving from a state i to a state j; therefore, Pij is a conditional probability Pij = p(j | i). The transition probability matrix is a square matrix containing non-negative real numbers and the summation of each row in this matrix is one [16].

CPU utilization will be partitioned into number of different levels where each level represents a different state. In this work, the CPU utilization level will be partitioned into 11 different intervals ranging from 0 to 100 %, which define the different states of the system. Random variables S1, S2, Sn will be used for defining these levels as in Table 1.

To represent various states of the Markov Chain Model, the state transition diagram and the transitions between different states are drawn in Fig. 1. This figure shows the state diagram of the CPU utilization model having four states [S1, S2, S3, and S4] only. The notation P(Si | Sj) is representing the conditional probability of state Sj given the state Si. This state diagram has been used for the construction of the transition probability matrix.

The transition probability matrix has been constructed for showing the conditional probability of moving from one state to another. This matrix in n * n matrix is shown in Fig. 2 where Pij is the conditional probability of moving from state i to state j. Now, we can easily rewrite the previous matrix as represented in Fig. 3.

We have defined the initial distribution vector V1 = (P(s1), P(s2), …, P(sn)), where n represents the entire number of states in the system. In addition to that, the transition probability matrix T containing the probability of moving from one state

Table 1 CPU utilization level at different states

CPU utilization level	[0–10 %]	[10–20 %]	[20–30 %]	[30–40 %]	[40–50 %]	[50–60 %]	[60–70 %]	[70–80 %]	[80–90 %]	[90–100 %]	[100 %]
States	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11

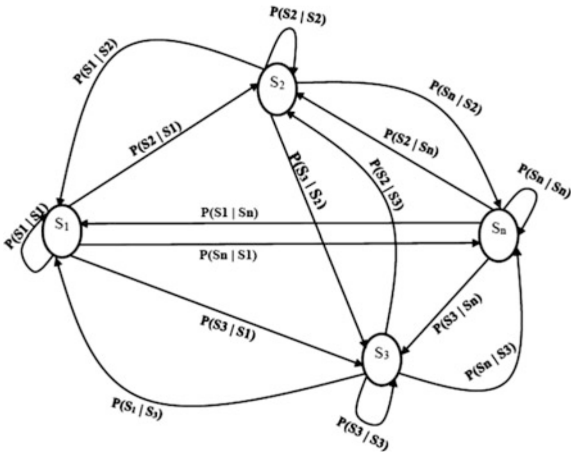


Fig. 1 CPU utilization state diagram

$$\begin{bmatrix}
State & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} \\
S_1 & P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} & P_{10} \\
S_2 & P_{21} & P_{22} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P_{29} & P_{210} \\
S_3 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_4 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_5 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_6 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_7 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_8 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_9 & P_{91} & P_{92} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P_{99} & P_{910} \\
S_{10} & P_{101} & P_{102} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P_{109} & P_{1010}
\end{bmatrix}$$

Fig. 2 Transition probablity matrix

$$\begin{bmatrix}
State & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} \\
S_1 & P(S_1|S_1) & P(S_2|S_1) & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P(S_9|S_1) & P(S_{10}|S_1) \\
S_2 & P(S_1|S_2) & P(S_2|S_2) & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P(S_9|S_2) & P(S_{10}|S_2) \\
S_3 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_4 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_5 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_6 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_7 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_8 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
S_9 & P(S_1|S_9) & P(S_2|S_9) & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P(S_9|S_9) & P(S_{10}|S_9) \\
S_{10} & P(S_1|S_{10}) & P(S_2|S_{10}) & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & P(S_9|S_{10}) & P(S_{10}|S_{10})
\end{bmatrix}$$

Fig. 3 Transition probablity matrix with different CPU states

to another has also been defined. The predicted distribution vector will be the result of multiplying the initial distribution vector by the transition probability matrix. As an illustration, the formula “V2 = V1 * T” will be used for computing the successor “predicted” state V2 based on the current distribution V1 and the transition probability matrix T. The result will be a vector with each element representing the probability of a particular state.

	Algorithm 2: Markov CPU Predication Policy
Step1:	Initialize the Utilization Vector
Step2:	Initialize the CPU states
Step3:	Utilization Vector = current utilization[] * transition Matrix[][]
Step4:	For length of utilization Vector
Step5:	Utilization Level += utilization Vector[i] * avgOfEachState
Step6:	Return utilizationLevel

3.3 *Hotspot Migration Policy*

This policy is based on the condition of hotspot where server is said to be in hotspot if it is utilizing its resources above a particular threshold point. If that threshold value is assumed to be TH and for each node Ni the load is calculated as L(Ni). Let the total sum of loads be SN. The average load present in all the nodes is calculated in variable avg and a new factor called hotspot factor α is introduced. Now, the new threshold value is determined based on the Avg value and the hot spot factor α. The node having total load greater than the threshold value will be deemed as hot spot node. Thus, all such nodes should be migrated. The Algorithm 3 details the HP Policy.

	Algorithm 3: HP Policy
Step 1	Find the load for each node L(Ni);
Step 2	Let SN = 0 and α = 1.2;
Step 3	For each node Ni // Calculate the load of each node
Step 4	SN = SN + L(Ni); // total sum of loads
Step 5	avg (avg) = SN/Ni;
Step 6	TH = α * avg; // new threshold value
Step 7	For each node Ni repeat up to step 9;
Step 8	If L(Ni) > TH then mark the node as hotspot;

4 Performance Evaluation

The policies designed in this work are for the IaaS model in cloud computing and, hence, it requires the extensive evaluation on a large-scale cloud datacenter. However, it is practically tough to conduct repeatable experiments at such large-scale setup as required by the proposed policies. Therefore, we have chosen

CloudSim [10], a very popular among cloud researchers, developed by the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne. It ropes in both behavioral aspect and system modeling of Cloud environment such as data centers, resource provisioning policies and energy-efficient management.

4.1 Experimental Setup

We have taken two configurations of servers in our experiments from the real power consumption data available from the real data given by SPECpower_ssj2008 benchmark. We have simulated a datacenter with 500 Snodes, with first half being Dell Inc. PowerEdge R620 (Intel Xeon E5-2670, 2.6 GHz, 8 Core, 24 GB) and the second half being Dell Inc. PowerEdge R720 (Intel Xeon E5-2670, 2.6 GHz, 8 Core, 24 GB).

The modeling of the four virtual machines for the experiments was done as per the Amazon EC2 Instance Types. In this modeling, each VM type has 1 core. Initially, the VMs are allocated with the maximum requested resource according to its type.

4.2 Workload Data and Framework

For more robust performance evaluation, the real workload data as used by CloudSim toolkit have been adopted in this work along with a series of experiments. For all the experiments, randomly generated cloudlets using the CloudSim toolkit are used for representing the application's workload to the VM. The length of each cloudlet is 2500 multiplied by the SIMULATION_LIMIT. The SIMULATION_LIMIT represents how long the simulation works. For all the following experiments, the simulation works for 1 day. The performance evaluation was based on three different sets of experiments with three different configurations. As this work focuses on the effect of the three VM selection policies on the energy consumption and SLA violation, while evaluating the performance, each of them would be compared with the existing state of art. The three proposed VM selection policies have been compared with the three VM selection policies, namely MMT, RS and MCC as described in the previous sections. As in the work done by Anton [7, 8], the MMT gives the best energy and SLA tradeoff; therefore, we have shown the comparison of our policies with MMT only. Similarly, out of the four algorithms for host overload detection that have been implemented in CloudSim, i.e., IQR, LR, LRR and MAD, our experiments take into account the LR (Local

Regression with threshold value 1.2). So the total combinations of six VM selection policies along with two VM overload detection policies result in twelve combinations for performance evaluation.

4.3 Results and Analysis

In this section, we have shown and analyzed the results of the proposed VM selection policies. In slightly loaded datacenters, any increase in the number of VMs accompanied with an increase in the number of hosts with the same percentage, using either the proposed VM selection policies or heuristics-based approaches, results in the following: The amount of energy consumed increases by a percentage that is slightly less than the percentage of the increase in the number of VMs. The percentage of SLA violations remains approximately at the same level without any substantial changes. In more loaded datacenters, using either the proposed VM selection policies or heuristics-based approaches, any increase in the number of VMs almost produces the same results found in slightly loaded datacenters. This means that there will be an increase in the energy consumption with a percentage that is somewhat less than the percentage of increase in the number of VMs. However, the percentage of SLA violations approximately remains in the same range. Increasing energy cost in the proposed policies by 50 % results in an increase in the percentage of SLA violations by about 45 %. Nonetheless, there is only about 10 % decrease in energy consumption. This result emphasizes on the importance of thoroughly determining the cost of energy and violations. All the foregoing experiments showed that the proposed approach is more efficient than the heuristics-based approach, on average, as it produced less energy consumption besides minimizing the percentage of SLA violations. However, there are notable variations in the values of the performance metrics among the different runs of the same experiment compared to the consistency in the results of the heuristics-based approach. The scatter plot for the three different proposed policies against the existing heuristic-based approach that is MMT has been shown in the figures (Figs. 4, 5 and 6).

Fig. 4 SLA violation to energy consumption MedMMT policy

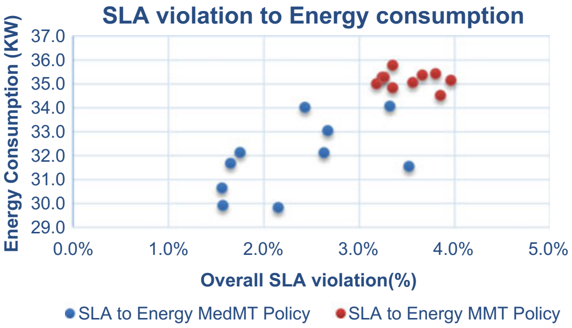


Fig. 5 SLA violation to energy consumption MaxUT policy

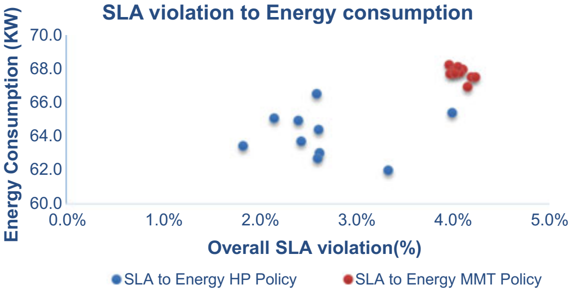
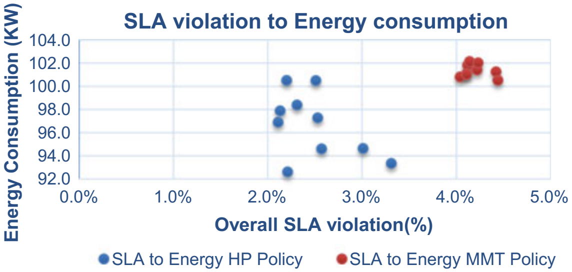


Fig. 6 SLA violation to energy consumption HP policy



4.4 Conclusion and Future Directions

To make best use of the ROI (Return on Investments) Cloud providers have to keep reinventing energy-efficient VM provisioning policies and minimize the SLAs. In this paper, we have proposed and evaluated three novel approaches for VM selection in the Dynamic VM provisioning environment which aims to mitigate the effect of hotspot. According to the findings from the experiments and the results of the analysis, we can deduce that minimizing the energy consumption by the datacenters all across globe has gained momentum. Also the SLAs have to be kept under desired level so as to please the ever demanding users and also to stay ahead in the competitive market. We have evaluated the proposed algorithms through extensive simulations on a large-scale experiment setup and the results have clearly shown the significant reduction in the overall energy consumption while minimizing the SLA violations. For future work, it is necessary to evaluate the applicability of the proposed policies in the real cloud infrastructure. A further direction for future research is the VM provisioning heuristics based on evolutionary approaches like Genetic Algorithms, Ant Colony Optimization, and Bacterial Foraging.

References

1. Armbrust, M., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010). http://doi.acm.org/10.1145/1721654.1721672
2. Bilal, K., et al.: On the characterization of the structural robustness of data center networks. IEEE Trans. Cloud Comput. **1**(1), 1 (2013)
3. Foster, I., et al.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, 2008. GCE'08, pp. 1–10. IEEE (2008)
4. Amazon data centre size: http://huanliu.wordpress.com/2012/03/13/amazondata-center-size/. Accessed 4th Aug 2015
5. Cisco Global Cloud Index: Forecast and Methodology, 2013–2018, Whitepaper. Accessed 12 Aug 15
6. Koomey, J.G.: Estimating total power consumption by servers in the US and the world. Lawrence Berkeley National Laboratory, Technical Report (2007)
7. Beloglazov, A.: Energy-efficient management of virtual machines in data centers for cloud computing (2013)
8. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency Comput. Pract. Exp. **24**(13), 1397–1420 (2012)
9. Buyya, R., Beloglazov, A., Abawajy, J.: Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. In: PDPTA 2010, Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications. CSREA Press, United States of America, pp. 6–17 (2010)
10. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. 41(1):23–50 (2011)
11. Beloglazov, A., Buyya, R.: Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, Concurrency and Computation: Practice and Experience (CCPE). Wiley Press, New York (2011)
12. Cao, Z., Dong, S.: An energy-aware heuristic framework for virtual machine consolidation in Cloud computing. J. Supercomput. 429–451 (2014)
13. Li, Z., Li, X., Wang, L., Cai, W.: Hierarchical resource management for enhancing performance of large-scale simulations on data centers. In: Proceedings of the 2nd ACM SIGSIM/PADS Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS '14), pp. 187–196. ACM, New York, NY, USA (2014)
14. Jin, H., Deng, L., Song, W., Shi, X., Chen, H., Pan, X.: MECOM: live migration of virtual machines by adaptively compressing memory pages. Future Gener. Comput. Syst. **38**, 23–35 (2014)
15. Kumar, G.S., Toosi, A.N., Gopalaiyengar, S.K., Buyya, R.: SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J. Netw. Comput. Appl. **45**, 108–120 (2014)
16. Mallick, S., Hains, G., Deme, C.S.: A resource prediction model for virtualization servers. In: 2012 International Conference on High Performance Computing and Simulation (HPCS), pp. 667–671. IEEE (2012)

Minimizing the Energy Consumption of Cloud Computing Data Centers Using Queueing Theory

Ranjan Kumar, G. Sahoo, Vikram Yadav and Pooja Malik

Abstract The servers utilize a major proportion of the energy utilized in the cloud-based companies. Through this paper, we are proposing a group of methodologies and approaches via which it would be possible to attain an energy conservation of more than 80 % of the total energy consumed by the cluster. We are using an integrated mechanism in between the servers and the jobs, consisting of combination of optimization approaches: beginning from ant colony optimization in the cloud cluster to bee colony optimization in load balancing to queueing theory in solving different job arrival patterns and some intermediate optimization algorithms like PowerNap and more. We have designed this approach without degrading the quality of service of the datacenters/clusters.

Keywords Energy optimization • Cloud computing • Ant colony optimization • Bee colony optimization • Queueing theory • PowerNap • Cloud clusters • Algorithms

R. Kumar (✉)
Department of Computer Science & Engineering, Cambridge Institute of Technology,
Ranchi, India
e-mail: ranjan.sonu2009@gmail.com

G. Sahoo
Birla Institute of Technology, Mesra, Ranchi, India
e-mail: gsahoo@bitmesra.ac.in

V. Yadav · P. Malik
IBM, Delhi, India
e-mail: vikram.yadav177@gmail.com

P. Malik
e-mail: poojapmalik@googlemail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_20

1 Introduction

With increasing demand for cloud computing and its services and its applications in other fields to share resources and minimize the cost of hardware for many enterprises, the only dominant problem comes out to be energy consumption for these giant clusters and stations. The similar power-saving approaches are applicable for datacenters and high-performance computing laboratories. In this paper, our major emphasis would be on, using approach of PowerNap on the non-pattern following job distributions, and application of queueing theory to find out an optimal resource using a mechanism to serve the jobs following certain distribution patterns, namely, Poisson, exponential, etc. The other auxiliary optimization tools which we would require to the prior optimization and classification of the server's states like idle, overloaded and under loaded would be ant colony optimization and later bee colony optimization for the load balancing among the under loaded and overloaded servers. The various factors that we could be looking forward as parameter to our research are:

- Each server's number of task operating capacity.
- Rate of arrival of jobs (r).
- Number of jobs to arrive according to last arrival pattern (n).
- Time taken by server for each job (y).
- Time taken by server to turn on from PowerNap state.
- Time taken by server to turn on from off state.
- Expected time of next job arrival (x).
- Energy consumed in resuming from off state.
- Energy consumed in resuming from off state.
- Energy consumed per second in active state.

The Organization of this paper is as follows. Related work is discussed in Sect. 2. Research methodology approaches used by us are discussed in Sect. 3. The approach's workflow is discussed in Sect. 4. The proposed procedural algorithm for queuing theory-based model and process workflow chart are discussed in Sect. 5. Section 6 gives the conclusion of our work.

2 Related Work

The major obstacle in energy conservation in cloud servers is variable workloads. There are many approaches for energy conservation among them, we have chosen a few of them and used them in different steps for our convenience. The datacenters run on 20–30 % utilization of datacenter servers [1]. Still, such poorly utilized servers consume 60 % of peak power. The solution of the author is PowerNap which suggests to put the system in a pseudosleep state where only its network and jobs receiving terminal are kept on. In [2], 7–16 % gap is achieving peak power and theoretical peak power which increases to 40 % in whole datacenter. The answer to this problem proposed by this paper was use of CPU voltage and frequency scaling [DVS].

In [3], energy optimization is done through dynamic voltage frequency scaling [DVFS] using work load decomposition the result came out to be 20–40 % energy saving to the cost of 10–30 % performance degrade. We applied a new approach of queueing theory in our work, and the text books we referred are [4] and [5]. In addition, for different queueing models analyses, we took help from [6] and [7]. Meisner [6] presented a PowerNap strategy for eliminating server idle power. Luo [8] presented a resource scheduling algorithm based on energy optimization methods. Feller [9] summarized the energy consumption and adjustment methods of standalone computer and cluster servers. In Hsu's work [10], they continuously observed the energy consumption and server utilization of server from 2007 to 2010, pointing out that server's defaults energy model has changed. In Lively and Wu's work [11], they use precise experimental probes to collect energy consumption and performance characteristics in different parallel implementations. Aydin et al. [12] have proposed the minimizing the energy consumption and subsequently the cost for the static system. Mukherjee and Sahoo [13] have proposed a framework for achieving better load balancing in Grid environment. Again, Mukherjee and Sahoo [14] proposed an ant colony optimization and Bee colony for service rescheduling. Once again, Mukherjee and Sahoo [7] proposed an algorithmic approach for Green Cloud.

3 Research Methodology Approaches

3.1 Ant Colony Optimization

Ant colony optimization technique was proposed by Marco Dorigo in the early 1990s [15]. In our problem, there are n nodes in a cluster. Some are under loaded, overloaded, and idle. Ant colony optimization will help in traversing all nodes by shortest path (TSP). After traversing all the nodes, we will find the nodes that are idle and we will not make them turn off. In our practical implementation, we have provided a color code (RGB code) to each node. We will set a threshold value like summation of RGB must be less than or equal to 300. The node will be turn off or on the basis of RGB scale. Ants (blind) navigate from source to food with the help of Pheromone. Initially, the ants navigate in all possible paths from source to food uniformly. On the long path, ants take more time to return as there will be more distance between the ants. Therefore, the evaporation will be high as compared to small path, there will be less distance between the ants and the evaporation will be less. In the next iteration, more ants will navigate in that path that has more pheromone means shortest path. Thus, shortest path is discovered via pheromone trails. Each ant moves in random, and pheromone is deposited on path. Ants detect the lead ants' path inclined to follow more pheromone path. Ant reached next node, selects the path, and continues until reaches start node. Finished tour is a solution. A completed tour is analyzed for optimality. Trail amount adjusted for favor better solution. Higher the probability of ant selecting path that is a part of better solution. A better solution receives more trails. A worse solution receives less trails. Higher the probability of ant selecting path that is a part of a better performing tour. New cycle is performed repeated until most ants select the same tour on every cycle.

3.2 Bee Colony Optimization

Bee colony optimization is used for load balancing. There are many nodes that are under loaded, and some are overloaded. We will uniformly distribute the load among all the nodes. Honey Bee is a social insect. They work in a decentralized and well-organized manner. There are two types of bees: one is forger bees, who collect the nectar and food stores, who store that in hives. Forger bees move out for searching the nectar (food). They move randomly in any direction. After finding the nectar, they come back on the hive and start dancing on the dance floor. The duration of this dance is closely correlated with the search time experienced by the dancing bees. There are two types of dance, waggle dance which implies poor quality of nectar and tremble dance (round dance) which implies good quality of nectar. If the dance is tremble dance, then the new born bee agents fly to collect the good quality nectar and store them in the hive. After this operation, the old bee agents die and the new born bee agents start to fly with the good quality nectar stored in the hive, and finally mix them with those sources which are holding poor quality nectar. This process of distribution goes on until there is a uniform quality of nectar in all the sources. Similarly, in cloud computing environment, we observe that some CPUs of IaaS are overloaded for processing consumers' services, some are under loaded, and some are totally idle. We can save the consumption of energy by turning this idle CPUs OFF and rescheduling services from overloaded CPUs to under loaded CPUs.

3.3 PowerNap

It is a state when only the system components that detect the arrival of new work are only powered except these units: all the other high-power consumption units like the CPU, DRAM and all the processing units are turned off and do not utilize power. It can be utilized to achieve enormous energy saving during idle states. It has two operating modes active mode and nap mode.

1. Wake delay: This is the time duration taken by the processing unit to turn on all its services from the state of nap till the wake delay ends. The work remains fixed.
2. Nap delay: This is the time allotted between the period when the work/job become zero to that of the nap's starting period.

Explanation: Initially, the cluster is at Nap. When the work arrives from the services, the servers are powered on and the time taken to turn it on from the Nap state that is referred as wake delay. During this wake delay, the work remains constant, and after it, all the servers start functioning and executing the jobs. Again, when the jobs end/finish, then the cluster for jobs for a definite time, after which it sends the servers, is back to Nap. This definite amount time is referred as Nap delay. After which the Nap remains till any further work is notified. The Fig. 1 shows the powerNap analytic model.

3.4 RAILS (Redundant Array of Inexpensive Load Sharing)

It is basically the replacement of the general (common) power supply units (PSUs) with a combination (parallel) of multiple inexpensive PSUs. This is very feasible for PowerNap, since in PowerNap state, a very small energy is requirement, and with need, each PSU is activated and power is taken into the consequently activated servers. Key features of RAILS are as follows:

- Efficient dynamic power output switching.
- Tolerance of PSU failure using N + K model.
- Minimal cost.

Our approach will work into two independent modules functioning simultaneously.

Module I

It will deal with the load at the servers and determining its state:- idle, under loaded, and overloaded by simple shortest path traversal using ANT COLONY

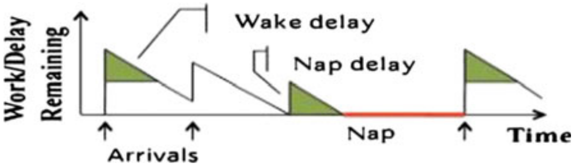


Fig. 1 PowerNap analytic model

OPTIMIZATION. The under loaded and overloaded are then balanced using the same nearest/shortest path using the BEE COLONY OPTIMIZATION, and the idle nodes are sent into PowerNap state with RAILS. An algorithm will be functioning continuously to determine the pattern of the job arrival distribution. It would be a complex algorithm to check the different (five models) possible job arrival distribution models on the data sets and give the result to the queueing theory-based mechanism in the idle nodes.

Module II

Now, the load on the set of under loaded and overloaded (or the active state servers) when comes close to their maximum load handling limit according to some particular pattern followed by the jobs, then we will apply the queueing theory model to turn on the required number of idle servers which are at PowerNap state or off state.

If,

$\mathbf{r*p \le s}$	Remain awake
$\mathbf{s < r*p < t}$	PowerNap
$\mathbf{r*p \ge r}$	Turn off

4 Proposed Procedural Algorithm

The procedural simulation would work on the underlying Algorithm. An infinite running iteration will be running to continuously performing the given operations. First, we will be traversing throughout the cluster to find out the shortest path among the nodes, and mark the nodes idle, overloaded, and under loaded according to the load present on them using ant colony optimization approach. Second, we will send all the passive/idle nodes to PowerNap using RAILS. Now, we enter the infinite iteration of load sharing and QUEUEING model. Here, we will distribute the load equally among the overloaded and under loaded nodes using bee colony optimization. Next, there will be a function to determine the job distribution pattern, whose result is passed as a parameter in the next and final system of QUEUEING model, along with list of idle and active servers. This system would perform activation of the idle servers according to the need just before the job arrival which cannot be handled by the existing active servers. The same system will send the active nodes to PowerNap also when there is no work job for them for a definite time interval. The Fig. 2 shows the life cycle model of a node.

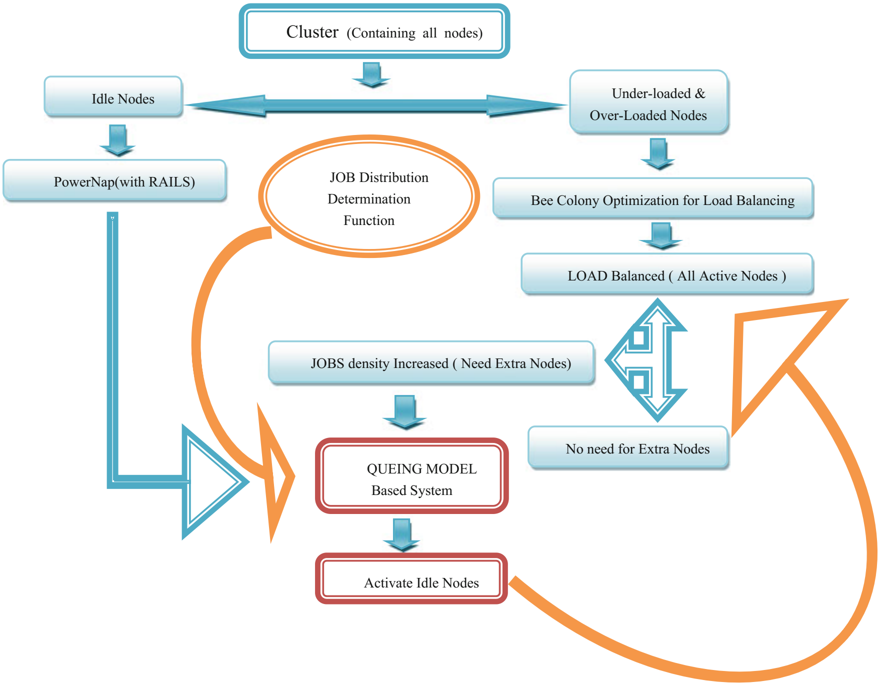

Cluster (Containing all nodes)
Idle Nodes
Under-loaded & Over-Loaded Nodes
PowerNap(with RAILS)
JOB Distribution Determination Function
Bee Colony Optimization for Load Balancing
LOAD Balanced (All Active Nodes)
JOBS density Increased (Need Extra Nodes)
No need for Extra Nodes
QUEING MODEL Based System
Activate Idle Nodes

Fig. 2 Life cycle of node

EnergyOptimizeCombo().

```
// Traverse shortest path among all nodes by ACO Optimization
ACO (all nodes)
Mark Idle, Over-loaded & Under-loaded nodes
For (Idle Nodes)
  /*Call */     RAILS()
  Apply_PowerNap()
While (true)

  For ( Overloaded & Under loaded Nodes )
    Apply_BeeColonyOptimization()
/* The above function (Apply_BeeColonyOptimization() )
Balances the Load between Overloaded and Under loaded
Nodes */
Determine_Job_Distribution(Jobs[])
/* The above function would check the Distribution pattern of Jobs & take the input Jobs as an Queue (Jobs[])
*/
If ( QOS Degrading || All Active Nodes are Nearly Overloaded)
  QUEUEING_MODEL_SYSTEM ( Idle Servers, Active Servers
Info, Job Distribution Pattern)
/* QUEUEING_MODEL_SYSTEM() function Activates the Idle
Server According to Incoming Job Demand */
```

5 Algorithm for Queueing Theory-Based System

QUEUEING_MODEL_SYSTEM ()
{

```
// the initial number of running nodes
running_nodes := INITIAL_NODES;
Load:= 1;
// 0 means that the load decreased,
// 1 that it keeps the same
// 2 that it increased

do forever {
sleep for N seconds;
  new_load := CalculateLoad ();
// the load increased
if (new_load > current_load && running_nodes< MAX_NODES
    {
    WakeupNode();
running_nodes++;
     }
// the load decreased
else if (new_load < current_load && running_nodes >
MIN_NODES)
{
node_to_be_removed := ChooseNodeWithTheLeastWorkload();
RemoveNodeFromQueue(node_to_be_removed);
PowerNapNodeWhenEmpty(node_to_be_removed);
Running_nodes--;
}
} }
CalculateLoad(Average) {
last_arrivals[] := GetNLastArrivals();
average_lambda_inverse
:=CalculateAverageLambdaInverse(last_arrivals[]);
neeed_NODES:=CalculateNeededNodesNumber(average_lambda_in
verse,MEAN_DELAY,MEAN_SERVICE_TIME);
// the load increased
if (running_nodes < needed_NODES)
return 2;
// the load decreased
else if (running_nodes > needed_NODES)
return 0;
// the load stays the same
return 1;
}

CalculateLoad(Exponential)
  {
last_exponential_lambda_inverse :=
GetLastLambdaInverse();
last_arrival := GetLastArrival();
exponential_lambda_inverse := COEFFICIENT1 * last_arrival
+COEFFICIENT2 * last_exponential_lambda_inverse;
neeed_nodes:=CalculateNeededNodesNumber(average_lambda_in
verse,MEAN_DELAY,MEAN_SERVICE_TIME);
// the Load increased
if (running_nodes < needed_nodes)
return 2;
// the Load decreased
else if (running_nodes > needed_nodes)
return 0;
// the Load stays the same
return 1;
}
CalculateLoad(Queue control)
{
waiting_threshold := SLOT_NUMBER + 1;
// SLOT_NUMBER is the number of jobs allowed to be
// executed on a single computing node
running_threshold := running_nodes* SLOT_NUMBER - 1;
waiting_jobs_number := GetNumberOfJobsInQueue();
running_jobs_number := GetNumberOfJobsInService();
if ((waiting_jobs_number - waiting_threshold) <(run-
ning_threshold-running_jobs_number))
 {
// the Load decreased
if (running_jobs < running_threshold)
return 0
// the Load increased
if (waiting_jobs> waiting_threshold)
return 2
}
// the Load stays the same
return 0;
}
```

6 Conclusion

Our work laid emphasis on the energy conservation in cloud computing clusters through combination of efficient energy optimization approaches resulting an effective energy conservation that we used ant colony optimization, bee colony optimization, RAILS, PowerNap, and queueing theory-based models in different steps to get the favorable results, thereby increasing the quality of service and service rate with energy conservation.

References

1. Meisner, D., Gold, B.T., Wenisch, T.F.: PowerNap: Eliminating server idle power. SIGPLAN Not. **44**, 205–216 (2009)
2. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: Proceedings of the 34th Annual International Symposium on Computer Architecture, ISCA'07, pp. 13–23. ACM, New York, NY, USA (2007)
3. Choi, K., Soma, R., Pedram, M.: Dynamic voltage and frequency scaling based on workload decomposition. In: Proceedings of the 2004 International Symposium on Low Power Electronics and Design, ISLPED'04, pp. 174–179. ACM, New York, NY, USA (2004)
4. Bhat, U.N.: An Introduction to Queueing Theory: Modeling and Analysis in Applications. Statistics for Industry and Technology. Springer, Dordrecht (2008)
5. Hall, R.W.: Queueing Methods: For Services and Manufacturing. Prentice Hall (1991)
6. Andersson, M., Cao, J., Kihl, M., Nyberg, C.: Performance modeling of an Apache web server with bursty arrival traffic. In: IC'03: Proceedings of the International Conference on Internet Computing. CSREA Press (2005)
7. Mukherjee, K., Sahoo, G.: Green cloud: an algorithmic approach. Int. J. Comput. Appl. **9**(9) (2010)
8. Luo, L., Wu, W.: A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods. IEEE (2010)
9. Feller, E., Leprince, D., Morin, C.: State of the art of power saving in clusters results from the EDF. Case Study (2010)
10. Hsu, C.H., Poole, S.W.: Power signature analysis of the SPECpower_ssj2008 benchmark. In: 2011 IEEE International Symposium Performance Analysis of Systems and Software (ISPASS), pp. 227–236 (2011)
11. Lively, C., Wu, X., Taylor, V., Moore, S., Chang, H., Cameron, K.: Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems. Int. J. High Perform. Comput. Appl. **25**(3), 342–350 (2011)
12. Aydin, H., Melhem, R.G., Mosse, D., Mejia-Alvarez, P.: Power-aware scheduling for periodic real-time task. IEEE Trans. Comput. **53**(5) (2004)
13. Mukherjee, K., Sahoo, G.: Mathematical model of cloud computing framework using fuzzy bee colony optimization technique. In: International Conference on Advances in Computing, Control and Telecommunication Technologies, IEEE Xplore (2009)
14. Mukherjee, K., Sahoo, G.: A framework for achieving better load balancing and job scheduling in Grid environment. Int. J. Inf. Technol. Knowl. Manage. **2**(1), 199–202 (2009)
15. Dorigo, M., Grambardella, L.M.: Ant colonies for travelling sales man problem. Biosystems. **43**, 73–81 (1997) Elsevier
16. Ghemawat, S., Gobioff, H., Leung, S.T.: The google file system. In: SOSP, 19–22 Oct, ACM (2003)

Cloud and Virtualization Based Log Management Service

Sai Rakesh Ghanta and Ayoush Mukherjee

Abstract With our computers becoming more capable and internet becoming more robust, virtualization- and cloud-based services have become prominent in their use. Virtualization technology has become a cost-effective and fast way to run simulations of models. Application virtualization helps run programs in environments different from the underlying OS. The cloud, similarly, has become an indispensable platform to deliver a variety of services, such as storage, software, network, infrastructure, and even virtualization. The aim is to integrate virtualization using open virtualization archives (OVAs) or LXC Containers with OpenStack [1] to host a cloud, along with real-time data analytics implanted using ELK (Elasticsearch [2, 3], Logstash, Kibana) stack [4] to create a platform which provides an alternate approach to the present practices of log management in companies, as the present log management technologies lack in one or more of efficiency, security, isolation, and ease of use.

Keywords Log management · Cloud computing · Virtualization technology · OpenStack · ELK stack · Elasticsearch · Logstash · Kibana · Infrastructure as a service

1 Introduction

Companies employ a large number of machines, such as servers, routers, and Linux machines. These produce large amounts of log messages each day, which are stored in dedicated log files, by default. These logs need to be analyzed and diagnosed to identify problems in the log and event data. For this purpose, the companies have

S.R. Ghanta · A. Mukherjee (✉)
Computer Science Department, Birla Institute of Technology,
Mesra, Jaipur 27 Malviya Industrial Area, Jaipur 302017, Rajasthan, India
e-mail: ayoush94@hotmail.com

S.R. Ghanta
e-mail: rakeshghanta153@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_21

system admins. These system admins use tools like Cat, Tail, and Grep, which are popular tools for log management. While this present system works, it lacks in a few areas, and does not offer much efficiency. A newer system is proposed herein, which intends to counter these issues.

1.1 Problems in the Present System

The sheer volume of logs generated by every machine, each day makes this a complex task. Upon identifying problems in logs, they are often compared with logs from previous days and records to identify a common cause to the problem, or to identify a pattern in the errors, so that further errors can be predicted and prevented. However, there is no log management service that makes this process intuitive.

The present log management solutions do not offer any way to cross-reference the present log record with previously encountered log records to make helpful analysis. The system admin generally has to scroll through the multitude of log messages, and manually search for that instance, which displayed similar characteristics. This is a time-consuming and effort intensive process. Moreover, this makes real-time log management harder, because there is no tag-based categorization of logs. In addition, there is no classification of logs based on patterns.

Another issue occurs, if the enterprise chooses to collect the logs into a server. Enterprises typically have many departments and business units. In such a case, the logs from all the units go on to the server and are stored based on their timestamp. This makes identifying logs from one source a tedious process. Moreover, putting the logs of all business units on a single server leads to security and isolation issues. What this means is that the person who has been given access to the server to view the logs of one business unit can view the logs of the other business units as well. Moreover, if the server is breached, the entire log record, which may contain sensitive data, is made vulnerable. The lack of isolation of logs from those of other business units compromises the security of the log management system.

1.2 Proposed Solution

The solution proposed, herein, is to devise a cloud-based service with virtualization to segregate the logs of each individual business unit for security and isolation purposes. Moreover, a platform is employed to search, analyze, and visualize the data, providing actionable insights in real time.

The platform will be delivered as IaaS (infrastructure as a service) using OpenStack as a cloud to provide a DevOps kind of enablement. Virtualization will be implemented using OVAs (open virtualization archive such as SUSE Linux) or LXC Containers, which will be loaded with ELK (Elasticsearch, Logstash, Kibana) stack to perform real-time analytics on the log data.

2 ELK Stack

The ELK stack [5] comprises of three open-source technologies, Elasticsearch, Logstash, and Kibana. These technologies, when used together, create an end-to-end stack to yield real-time insights into any type of structured and unstructured data. As the ELK stack is open source with a wide user base and developer community, it is a very cost-effective and reliable solution to perform analytics on log data.

The ELK stack ecosystem consists of four components:

- Shipper: The remote agents generally run this component alone. It passes the events to Logstash.
- Broker and Indexer: Receives and indexes the events.
- Search and Storage: Searches and stores events.
- Web Interface: This is Kibana, a web interface to Logstash (Fig. 1).

The log data that are entered are parsed with Logstash directly into the Elasticsearch node. Elasticsearch performs the required analytics on the data, and then Kibana is used to visualize the data for the user.

The shipper and the Logstash are implemented on the client side, and the rest of the constituents are implemented on the server side. As the server is a single entity, the server side of the framework needs to be deployed only once. However, as there will quite possibly be multiple clients, the client side (shipper and Logstash) will need to be implemented for each individual client. While this may appear to be a tedious process, especially if there are many clients, a solution can be devised to automate the process using Chef or Puppet (open-source technologies used to build, deploy and manage infrastructure).

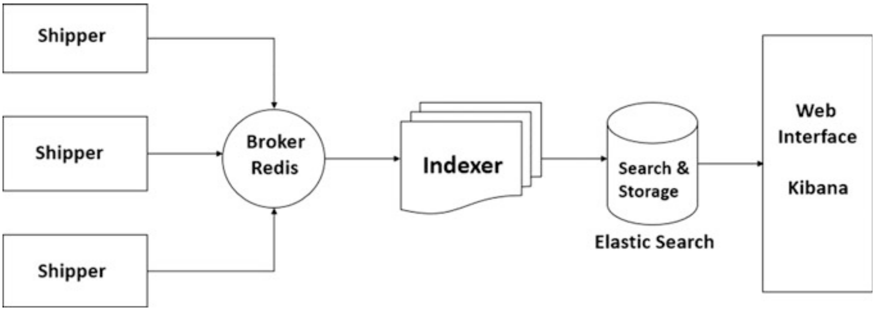


Fig. 1 ELK stack structure (*Source* The Logstash Book–Log Management Made Easy, p. 23)

2.1 Elasticsearch

Elasticsearch [6] is a search and analytics engine that is based on the RESTful architecture, and is built on top of the Apache Lucene retrieval library. It is made available under the Apache 2 open-source license and is distributed in nature. The distributed nature of Elasticsearch makes it horizontally scalable, and thus can handle clusters even when there may be a massive number of nodes present in them. Elasticsearch also adds reliability to the data, as it has automatic detection of malfunctioning nodes, and subsequently restructures the data to maintain balance.

Elasticsearch makes the process of providing data as input a convenient task, as it accepts indexed JSON (JavaScript Object Notation) files, and can automatically detect patterns and structures, and make the data searchable. The schema-free nature of Elasticsearch allows us to modify the indexing parameters based on our requirements.

The purpose of Elasticsearch, here, is to perform real-time analysis on the data, store it, and make it searchable.

2.2 Logstash

Logstash is an integrated framework for collection, centralization, parsing, and transportation of log data. Logstash has a message-based architecture, and is written in JRuby and runs on the Java Virtual Machine. It is capable of processing a variety of logging and event data, scale it, and then stream it to an analytics engine. It can efficiently process the large quantities of unstructured data that companies generate in short spans of time, and index it into JSON files for analytics by Elasticsearch. In an enterprise, different business units may have different schema and formats, which too can be parsed and normalized by Logstash.

The Logstash indexer employs the Grok filter to perform the parsing of the unstructured log data into structured and indexed JSON files.

2.3 Kibana

Kibana is an open-source, web-based interface to Logstash that helps to visualise the data. It has deep integration with elastic search and uses its search and analytics functionality to produce real-time summary of streaming data, with an intuitive user interface.

In this proposed model, the system admin can provide the timeframes, or other search parameters, based on which the Kibana interface will leverage Elasticsearch and Logstash to generate search results, perform analytics and then graphically visualize the data in real time.

2.4 Shipper

The function of the shipper is to capture the log instances and generate a stream of log data for the Logstash indexer. It is a plugin based Java configuration file, and is a Logstash agent. However, java-based configuration files require a greater amount of memory. The other alternatives are non-Logstash agents, such as HiRedis (C-Client library) and Beaver (Python-based lightweight framework).

2.5 Broker (Redis)

In real-world situations, there are multiple nodes, and thus, there might be hundreds of log files that are generated at a point in time. The shipper captures these log instances and forwards them to the indexer. However, the indexer may not be able to process the log files at the rate of arrival. The broker is essentially a buffer that is introduced to prevent the loss of data packets containing log instances. It offers two benefits:

- First, it enhances the performance of the Logstash buffer by acting as a caching buffer for log events.
- Second, it provides another level of backup, if indexing fails, as events will still be queued here.

3 Cloud and Virtualization

The problems faced by the present system were lack of efficiency, ease-of-use, isolation, and security. The ELK Stack is a simple server–client model that can solve the efficiency issues using Elasticsearch and Logstash, and make the vast amounts of data easy to use with Kibana, and the other two issues, namely, isolation and security, still remain unaddressed.

To address those issues, the ELK stack framework needs to be integrated with a cloud-based service to provide scalability, remote accessibility, and deliver the virtualization platform. This is called infrastructure as a service (IaaS). A popular platform that enables hosting of such a cloud is OpenStack [7]. The virtualization platform provides OVAs or Containers that facilitate isolation of data sets based on parameters (here, one OVA or Container in the cloud, per business unit), and by extension of this, security.

3.1 Cloud Service Using OpenStack

OpenStack [8] is an open-source set of tools to deploy public or private cloud-computing software platforms, used primarily for IaaS. It was developed by a collaboration between NASA and Rackspace. NASA contributed the technology that it uses in its Nebula high-performance computing system, and Rackspace contributed the storage, content delivery, and production platforms. It enables companies use open-source technologies to set up clouds in their own datacenters.

It helps users deploy multiple virtual environments to perform different tasks on the cloud. This has the benefit of allowing concurrent processing due to horizontal scalability. In addition, being open source, the user can access the source code, and customize the software to their requirements.

OpenStack comprises of a group of eleven technologies. The key members of OpenStack are as follows:

- Nova: This is the engine that provides a platform to host the VMs that handle computing tasks.
- Neutron: This provides the networking capability to the OpenStack components.
- Keystone: This maintains a list of all users of the cloud, with the list of components they have access to.
- Swift: This is the file storage system, where files or the data within them are referred to with unique identifiers. The storage location of the files is decided by the software itself, ensuring easy scaling.
- Glance: This maintains a repository of virtual machines that have been deployed on the cloud.
- Cinder: This provides a persistent block of storage to VMs.
- Horizon: This is the web-based user interface for OpenStack services.

3.2 Virtualization Using OVAs or LXC Containers

Linux Containers [9] and OVAs provide an efficient way to deploy applications and provide isolation and security. Suppose, Customer C1 faces issues on its routers at a certain time of the day for a month. They will need to monitor those routers during those times time to see the issues. They can enable the ELK log collection on the clients. At the same time, they will request a server (ELK container) to store the collected data. Customer C1 will be provided access to this server to search and visualize the logs. Once they are done with the server, they can request for stopping the service. The service can be provided to many customers at the same time leveraging on the security and isolation provided by the Linux Containers and OVAs. The testing team can use this to find out the bugs on the routers. The servers (Containers) can be started, frozen,, and stopped whenever the requirement arises.

This will help in efficient resource utilization on the main server hosting the containers.

The purpose of virtualization is to provide separate locations to manage the logs of each business unit.

Open Virtualization Archives (OVAs). Open virtualization format (OVF) is an open-source standard for packaging and distributing software to be run on virtual machines. OVAs are units of OVF packages that are deployed on virtual systems. OVAs do not require any host operating system, and can be directly deployed. It can act as an operating system. If a particular business unit does not require OVA, the instance can be easily killed via OpenStack's Horizon UI. An example of OVA is SUSE Linux.

LXC Containers. LXCs (Linux Containers) are an operating system level virtualization environment. It can run multiple isolated Containers on a single Linux control host. Containers are lightweight virtualization environments containing their own CPU, memory blocks, I/O and network. LXC Containers are not virtual machines, and need a host OS to be deployed, which is the Linux Control Host. LXC Containers are used to isolate applications from those running on other containers or the host.

In this solution, LXC Containers represent the virtualization technology.

4 Model for Integration of the Solution

The ELK stack [10] needs to be implemented in both the server and the client sides of the solution (Fig. 2).

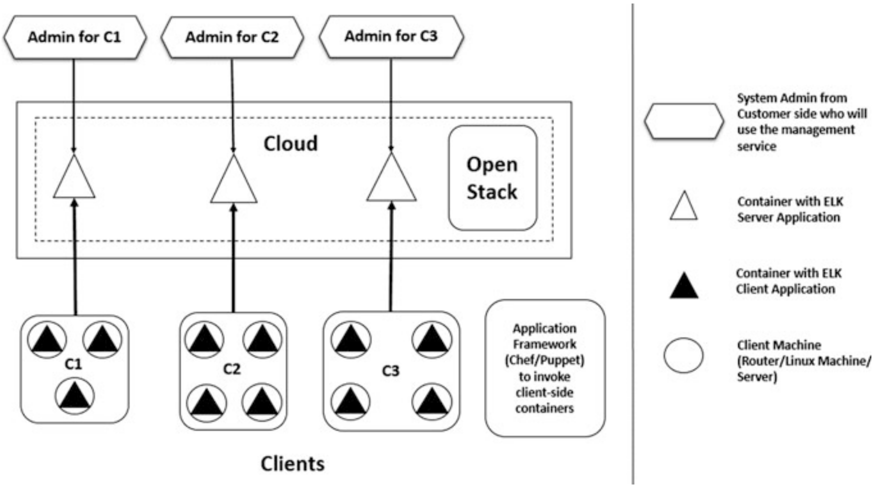

Admin for C1
Admin for C2
Admin for C3
Cloud
Open Stack
C1
C2
C3
Application Framework (Chef/Puppet) to invoke client-side containers
Clients
System Admin from Customer side who will use the management service
Container with ELK Server Application
Container with ELK Client Application
Client Machine (Router/Linux Machine/ Server)

Fig. 2 Model for integration of components

On the server side, there are Containers with ELK sever application consisting of the Elasticsearch, Logstash, and Kibana which collects, parses and stores the logs on the cloud. The containers run as individual instances on the cloud. These containers are built, deployed, managed, and demolished using OpenStack. When a new business unit requires a service from the cloud, a new instance of the container is built and deployed using OpenStack's Nova and Horizon. Here, ELK stack performs log management activity on the data. When the service needs to be retracted, the instance can just as easily be killed with the same tools.

On the client side, the Container with ELK client application consists of shipper and Logstash. These are used to form the pipeline between the client and server, and ship the logs into server container. When a new client needs to access the service, the Logstash and shipper need to be loaded on to all the machines belonging to the client. As a single client may have multiple machines, this becomes a tedious task. To make this process more intuitive, one of two open-source technologies—Chef [11] or Puppet can be utilized. As this process is similar to the one on the cloud, one may also use OpenStack, as it does not necessitate a cloud to function. However, OpenStack is optimized for cloud-based scenarios. Chef/Puppet is optimized for these scenarios. The container life cycle on the client side is managed by one of these solutions. Chef/Puppet provides an enablement similar to that of DevOps.

This solution is based on IaaS (infrastructure as a service), but can also be classified as a combination of SaaS (software as a service) and PaaS (platform as a service).

5 Conclusion

The presented solution integrates some of the newest and popular open-source technologies, to tackle the problems that many enterprises are facing in log management. To sum up, the following hold that:

- It improves the efficiency of the system using real-time analytics from concurrently operating virtual machines in the cloud. This speeds up the log management process.
- This system makes the data in the logs indexed, structured, and easily searchable.
- It also improves accessibility, as logs can even be accessed from remote locations.
- This model also greatly improves the security of the log data by providing exclusive containers for individual business units, and ensures that access to the servers can be monitored.
- Using Kibana, it provides an easy-to-use web-based interface to retrieve the logs from the server.

While all the technologies used in this solution are open-source, and thus highly customizable, a few alternatives to the software used are present:

- Splunk can be used as an alternative to ELK Stack, but Splunk is a license-based software. The choice of application can be decided based on merits and cost of each alternative.
- Containers are popular in recent times, but Docker, virtual appliances, and other virtualization techniques can also be used as alternatives.

References

1. What is OpenStack. http://opensource.com/resources/what-is-openstack
2. Gormley, C., Tong, Z.: Elasticsearch the Definitive Guide. Shroff (2015)
3. The Elastic Platform. https://www.elastic.co/products
4. Turnbull, J.: The Logstash Book—Log Management Made Easy, v1.5.1 (2015)
5. How To Install Elasticsearch: Logstash, and Kibana 4 on Ubuntu 14.04. https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-ubuntu-14-04
6. A bit on Elasticsearch. http://aarvik.dk/a-bit-on-elasticsearch-logstash-kibana-the-elk-stack/
7. Definition OpenStack. http://whatis.techtarget.com/definition/OpenStack
8. OpenStack. http://www.openstack.org/
9. LXC 1.0 blog post series. https://www.stgraber.org/2013/12/20/lxc-1-0-blog-post-series/
10. QBOX: Welcome to the ELK Stack. http://qbox.io/blog/welcome-to-the-elk-stack-elasticsearch-logstash-kibana
11. Chef. https://www.chef.io/chef/

Ranking Uncertain Distributed Database at Tuple Level

N. Lalithamani

Abstract In distributed database system, the data are located at different locations. As the data are at multiple locations, it may not be accurate. It may contain uncertain values or even some data may be missing. Due to impreciseness and uncertainty in the data, occurrence of error becomes high. This makes the processing of the data difficult. There are many ways to handle uncertain databases. To obtain required data, ranking technique is used. One such technique is the top-k query method where the data are retrieved according to user input. This paper proposes an algorithm that ranks and retrieves the data in minimum time at tuple level. In addition, the number of records traversed during this ranking and retrieval process is minimized. The time taken for retrieval of the records is also analyzed.

Keywords Distributed database • Top-k query • Tuple level • Ranking • Attribute level • Uncertainty

1 Introduction

Nowadays, the concept of uncertain data [1] is used by several applications, which is represented by probabilistic databases [2]. The amount of data stored is increasing day by day, so the management of uncertain distributed database has a major important.

The data uncertainty can be at attribute level or tuple level. In attribute level, any one of the attributes has uncertain value, and the remaining attributes have certain values. In tuple level, it is not necessary that the tuple is present. Here, the attributes of every tuple will be certain unlike in the case of attribute level.

N. Lalithamani (✉)
Department of Computer Science and Engineering,
Amrita School of Engineering, Amrita Vishwa
Vidyapeetham (University), Coimbatore, India
e-mail: n_lalitha@cb.amrita.edu

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_22

Uncertain database, the ranking or top-k [3] query is for finding the highest ranked k tuples from all the instances of the distributed sites known as possible worlds [4]. The ranking is done for getting the most relevant and consistent records related to the given query [5]. All the tuples in the distributed sites are not processed. Hence, unwanted tuples are eliminated, thereby decreasing the total number of tuples being traversed. Instead, only a part of data is delivered to the user according to his/her interest. Any query can be evaluated on every possible world. In this paper, we followed the expected rank approach [6]. Using this approach, k tuples with smallest expected ranks will be retrieved. This approach is used, because unlike in certain databases, the rank of the tuple varies in uncertain databases. Ranking in uncertain distributed database consists of two steps: first, tuples at each distributed sites are to be ranked, and second, the top-k list has to be retrieved [4].

The Shipboard Automated Meteorological and Oceanographic System (SAMOS) project aims to improve the quality of meteorological and near surface oceanographic observations collected in situ on research vessels (R/Vs) and select volunteer observing ships (VOSs) [7]. It stores the navigational, meteorological, and surface oceanographic parameters, while the vessel is at sea.

For ranking the databases, there are many techniques that are used which have its own drawbacks associated with it. For instance, the number of rounds involved in traversing the tuples will be more, thereby increasing the costs involved. Sometimes, the number of tuples traversed will be more that leads to increase in the time involved in the process.

In this paper, we developed an algorithm that overcomes the drawbacks and improves the efficiency of the algorithm. It aims to reduce the communication costs involved, and it ensures that the number of tuples that are traversed is also kept minimal.

2 Literature Review

In distributed database system, the data are located at different locations. As the data are at multiple locations, it may not be accurate. It may contain uncertain values or even some data may be missing. In distributed databases, ranking queries is an important technique used for retrieval of tuples as per the user requirement. The approach followed in ranking certain distributed databases differs from that being followed in uncertain databases.

In the distributed traditional databases, the efficient algorithm used for retrieving top-k queries is the threshold algorithm (TA) [4]. TA is applicable for queries where the scoring function is monotonic [4, 8]. There are two approaches that dealt with distributed uncertain databases ranking [7].

The first approach uses the expected rank technique [9]. In this approach, tuples present in every site are ranked and sorted accordingly. The server accesses all the tuples from every site according to their ranks. It maintains a priority queue that consists of the first tuples from every site. Then, the tuples are broadcasted to every site to compute its global rank. The execution stops when the first tuple has rank higher than or equal to that of the *k*th tuple. The second approach, which is applied in wireless sensor networks, uses any algorithm for centralized ranking after performing a tuple pruning step [7]. The framework for ranking uncertain distributed

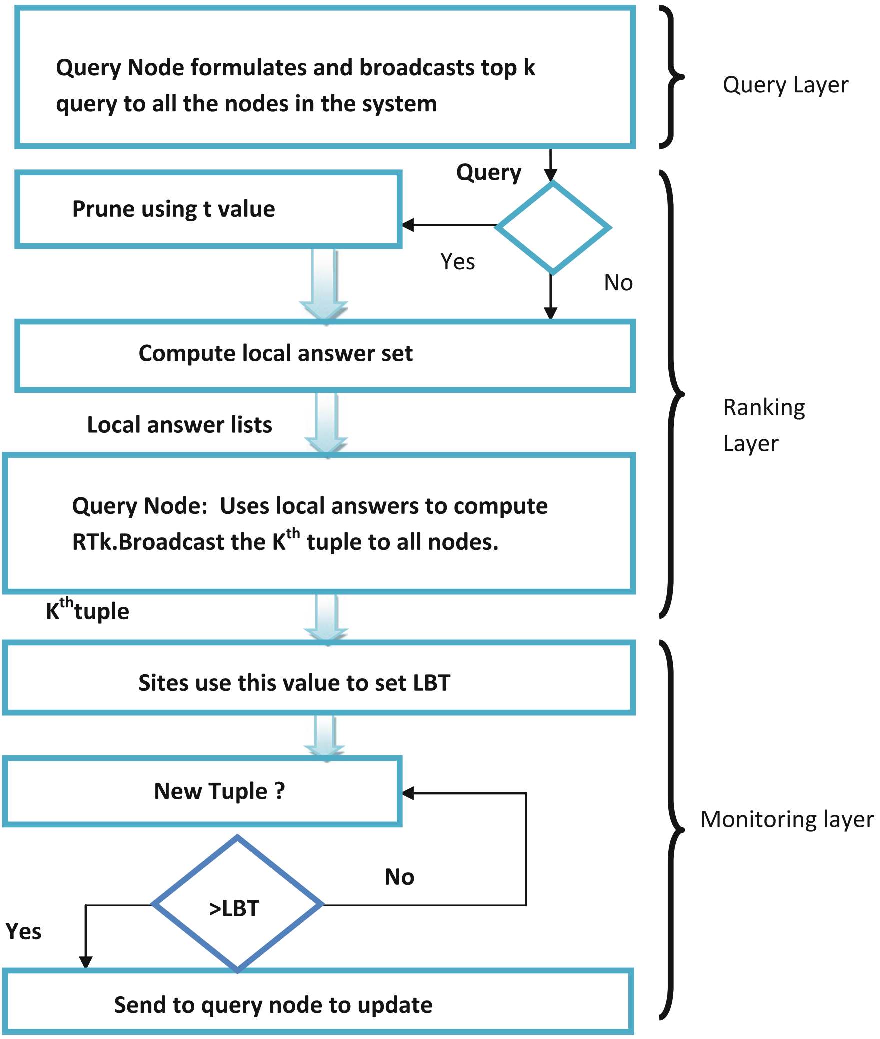

Query Node formulates and broadcasts top k query to all the nodes in the system
Query Layer
Query
Prune using t value
Yes
No
Compute local answer set
Local answer lists
Ranking Layer
Query Node: Uses local answers to compute RTk.Broadcast the K^th tuple to all nodes.
K^th tuple
Sites use this value to set LBT
New Tuple ?
No
>LBT
Yes
Send to query node to update
Monitoring layer

Fig. 1 Three-layered overview of ranking uncertain distributed database

databases consists of mainly three layers, namely, Query Layer, Ranking Layer and Monitoring Layer, as shown in Fig. 1 [4].

Figure 1 shows a framework for various characteristics that are essential while ranking uncertain distributed databases. It ensures that the communication costs that are involved during ranking process are minimized. In addition, the number of times the nodes being traversed is also fixed. Using this framework, the number of tuples that are considered for the ranking process is also reduced. Once the top-k tuples are retrieved, monitoring is done to check for the arrival of any new tuple in any node. There exist two algorithms for retrieving top-k tuples from all the nodes present in various distributed sites. The first algorithm is the threshold-tuned distributed top-k (TDTk) algorithm [4], and the second is the ceiling-tuned distributed top-k algorithm (CDTk) [4].

3 Proposed System

The framework for ranking uncertain distributed databases in the new system also comprises of three layers, namely, query layer, ranking layer, and monitoring layer [4]. In the query layer, the top-k query is sent to all the nodes in various distributed sites. In the ranking layer, ranking of the tuples takes place. The tuples are ranked to compute the most relevant tuples required for the query result. An algorithm is developed for this process.

The system is implemented at tuple level [2–4, 10] where the number of tuples stored at distributed sites varies. The data at the sites are sorted and ranked according to user's interest. This helps us to identify the most relevant tuples associated with the query. The user specifies how many tuples he wants to be retrieved. Here, the site with maximum number of tuples is considered. From that respective site, the middle tuple is sent to all the other sites along with its rank. All the tuples with a rank lower than the passed tuple will be considered for the top-k list. If the number of tuples exceeds than the required number, then the tuples that have higher rank are deleted from the sites, and the process is continued until the top-k tuples are retrieved.

Algorithm

1. Input k
2. Rank the tuples in various distributed sites
3. 3. Prune the tuples by user-defined threshold value, say, t (if any)
4. Find site with maximum tuples, say,
5. Let x be middle tuple of site

```
    Find no of tuples, say, c such that
    $r(t_i) < r(x)$
   if $c < k$
assign $t_i$ to tuple, say, DTk
   else
prune tuples having $r(t_i) > r(x)$
```

6. Repeat steps 4 and 5 till DTk has elements

Figure 2 shows the steps involved in the ranking and retrieval of top-k tuples from the distributed sites. First, if a threshold value is explicitly given by the user, the tuples are pruned and then considered for ranking process.

Otherwise, the tuples are sorted and ranked accordingly. Then, the middle tuple from the site having maximum number of tuples is passed to all the other sites. The rank of all the tuples is compared with the rank of received tuple. The tuples having lower rank than the received tuple are considered for top-k list. If the number of tuples having lower rank exceeds the k value, then the tuples are pruned from the sites accordingly until exactly top-k tuples are retrieved.

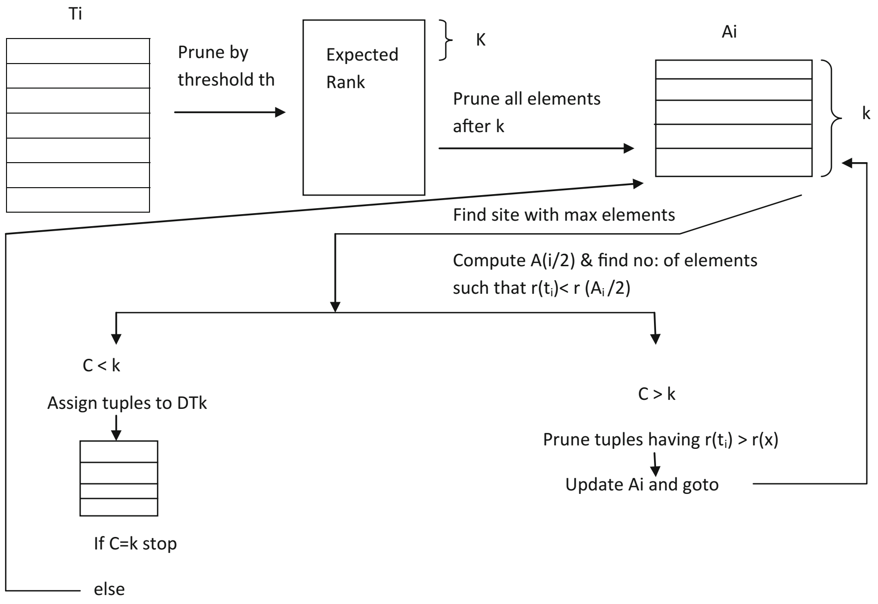

Ti
Prune by threshold th
Expected Rank
K
Ai
Prune all elements after k
k
Find site with max elements
Compute A(i/2) & find no: of elements such that $r(t_i)$< r (A_i/2)
C < k
Assign tuples to DTk
C > k
Prune tuples having $r(t_i)$ > r(x)
Update Ai and goto
If C=k stop
else

Fig. 2 Representation of steps involved in top-k query processing

If the top-k tuples are obtained as the query result, the *k*th tuple is again sent to all the nodes for the execution of the monitoring layer. In this layer, it checks for any new tuple which has been updated in any of the nodes in the system. If so, the rank of the new tuple will be compared with the received one. The new tuple will be considered to be a part of top-k list if it has a lower rank than the *k*th tuple. Those tuples will be sent to the query node to update the DTk list.

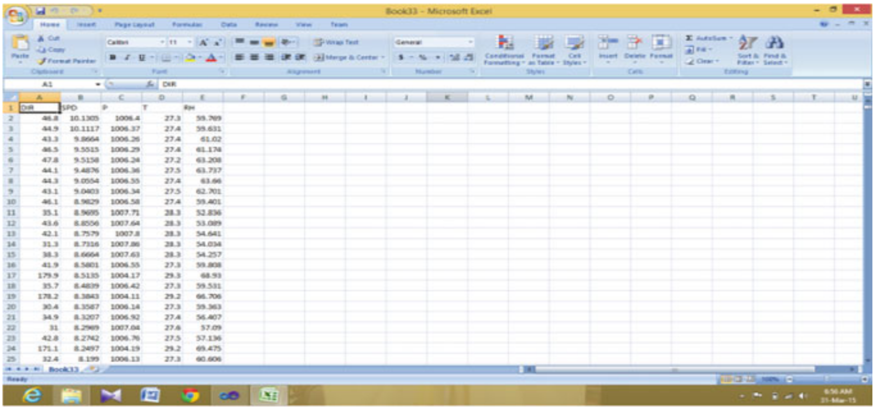
4 Results and Discussion

The data set used to test the algorithm is a real data set that is collected from Shipboard Automated Meteorological and Oceanographic System (SAMOS) project. The data collected are for 3 months which contains large number of records.

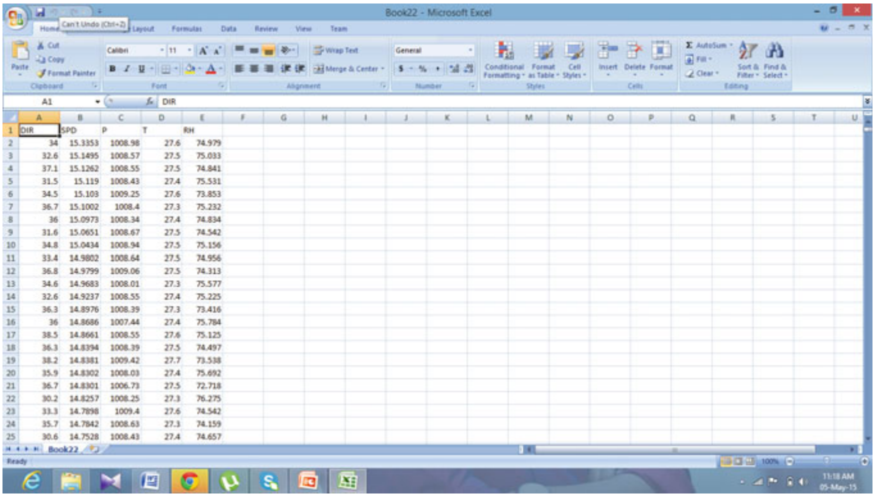
The proposed algorithm within the proposed framework has been implemented at tuple level for retrieval of the top-k tuples from distributed sites. The relevant top records are retrieved as per user requirement in a fast and efficient manner. In addition, the algorithm processes every site periodically for monitoring the arrival of new records. Accordingly, the sites get updated, thereby updating the final result if any change occurs after processing of the data.

4.1 Input Data

Input file 1 and 2



DIR	SPD	P	T	RH
46.8	10.1305	1006.4	27.3	59.769
44.9	10.1117	1006.37	27.4	59.631
43.3	9.8664	1006.26	27.4	61.02
46.5	9.5515	1006.29	27.4	61.174
47.8	9.5158	1006.24	27.2	63.208
44.1	9.4876	1006.36	27.5	63.737
44.3	9.0554	1006.55	27.4	63.66
43.1	9.0403	1006.34	27.5	62.701
46.1	8.9829	1006.58	27.4	59.401
35.1	8.9695	1007.71	28.3	52.836
43.6	8.8556	1007.64	28.3	53.089
42.1	8.7579	1007.8	28.3	54.641
31.3	8.7316	1007.86	28.3	54.034
38.3	8.6664	1007.63	28.3	54.257
41.9	8.5801	1006.55	27.3	59.808
179.9	8.5135	1004.17	29.3	68.93
35.7	8.4839	1006.42	27.3	59.531
178.2	8.3843	1004.11	29.2	66.706
30.4	8.3587	1006.14	27.3	59.363
34.9	8.3207	1006.92	27.4	56.407
31	8.2969	1007.04	27.6	57.09
42.8	8.2742	1006.76	27.5	57.136
171.1	8.2497	1004.19	29.2	69.475
32.4	8.199	1006.13	27.3	60.606

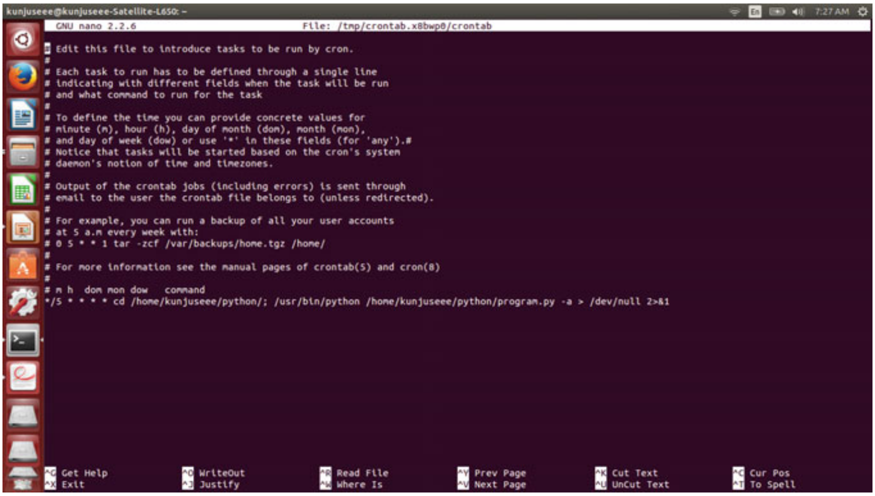


	A	B	C	D	E
1	DIR	SPD	P	T	RH
2	34	15.3353	1008.98	27.6	74.979
3	32.6	15.1495	1008.57	27.5	75.033
4	37.1	15.1262	1008.55	27.5	74.841
5	31.5	15.119	1008.43	27.4	75.531
6	34.5	15.103	1009.25	27.6	73.853
7	36.7	15.1002	1008.4	27.3	75.232
8	36	15.0973	1008.34	27.4	74.834
9	31.6	15.0651	1008.67	27.5	74.542
10	34.8	15.0434	1008.94	27.5	75.156
11	33.4	14.9802	1008.64	27.5	74.956
12	36.8	14.9799	1009.06	27.5	74.313
13	34.6	14.9683	1008.01	27.3	75.577
14	32.6	14.9237	1008.55	27.4	75.225
15	36.3	14.8976	1008.39	27.3	73.416
16	36	14.8686	1007.44	27.4	75.784
17	38.5	14.8661	1008.55	27.6	75.125
18	36.3	14.8394	1008.39	27.5	74.497
19	38.2	14.8381	1009.42	27.7	73.538
20	35.9	14.8302	1008.03	27.4	75.692
21	36.7	14.8301	1006.73	27.5	72.718
22	30.2	14.8257	1008.25	27.3	76.275
23	33.3	14.7898	1009.4	27.6	74.542
24	35.7	14.7842	1008.63	27.3	74.159
25	30.6	14.7528	1008.43	27.4	74.657

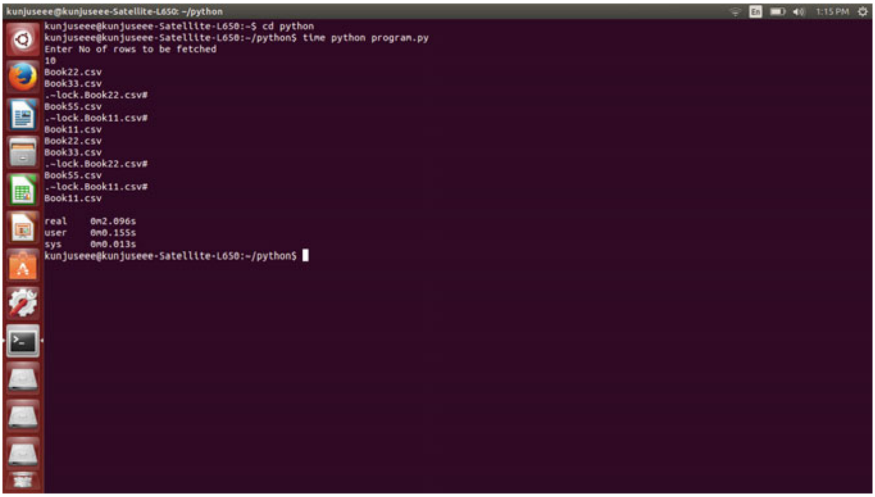
These show the sample data of a single record from the data set that is collected. The data set consists of large number of records each with the parameters direction (DIR), speed (SPD), pressure (P), temperature (T), and relative humidity (RH). The records are sorted using the SPD parameter.

The algorithm is implemented on the large number of sorted records, and the necessary number of relevant top-k tuples is retrieved as per the user requirement.

4.2 Output Data

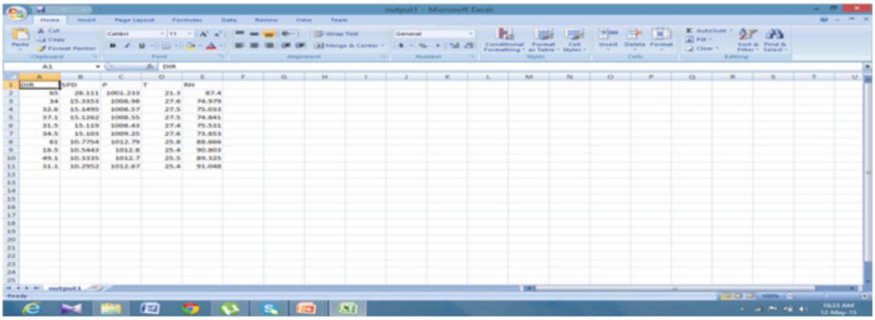


```
  GNU nano 2.2.6                    File: /tmp/crontab.x8bwp0/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * cd /home/kunjuseee/python/; /usr/bin/python /home/kunjuseee/python/program.py -a > /dev/null 2>&1
```



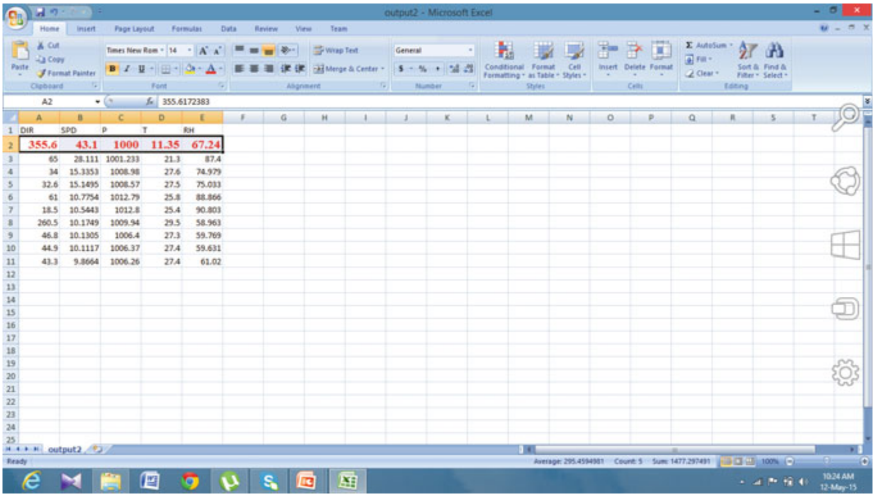
```
kunjuseee@kunjuseee-Satellite-L650:~$ cd python
kunjuseee@kunjuseee-Satellite-L650:~/python$ time python program.py
Enter No of rows to be fetched
10
Book22.csv
Book33.csv
.~lock.Book22.csv#
Book55.csv
.~lock.Book11.csv#
Book11.csv
Book22.csv
Book33.csv
.~lock.Book22.csv#
Book55.csv
.~lock.Book11.csv#
Book11.csv

real    0m2.096s
user    0m0.155s
sys     0m0.013s
kunjuseee@kunjuseee-Satellite-L650:~/python$ 
```

Top 10 records



DIR	SPD	P	T	RH
65	28.111	1001.233	21.3	87.4
34	15.3353	1008.98	27.6	74.979
32.6	15.1495	1008.57	27.5	75.033
37.1	15.1262	1008.55	27.5	74.841
31.5	15.119	1008.43	27.4	75.531
34.5	15.103	1009.25	27.6	73.853
61	10.7754	1012.79	25.8	88.866
18.5	10.5443	1012.8	25.4	90.803
49.1	10.3335	1012.7	25.5	89.325
31.1	10.2952	1012.87	25.4	91.048

Top 10 records after insertion of new tuple into input file 2



DIR	SPD	P	T	RH
355.6	**43.1**	**1000**	**11.35**	**67.24**
65	28.111	1001.233	21.3	87.4
34	15.3353	1008.98	27.6	74.979
32.6	15.1495	1008.57	27.5	75.033
61	10.7754	1012.79	25.8	88.866
18.5	10.5443	1012.8	25.4	90.803
260.5	10.1749	1009.94	29.5	58.963
46.8	10.1305	1006.4	27.3	59.769
44.9	10.1117	1006.37	27.4	59.631
43.3	9.8664	1006.26	27.4	61.02

5 Conclusion

Most of the recent ranking techniques have dealt either with uncertain data or with distributed data, and there is little literature in ranking distributed uncertain data. Uncertainty occurs both at tuple level and attribute level environment which makes it difficult for the processing of various distributed applications.

In this paper, the proposed algorithm focuses on retrieving the tuples at minimum time with maximum efficiency according to the user's interest. It focuses on minimizing the number of tuples being traversed while query processing. The algorithm also aims at reducing the communication cost and the response time involved in the ranking and retrieval process.

References

1. Li, X., Wang, Y., Li, X., Wang, X., Yu, J.: GDPS: an efficient approach for skyline queries over distributed uncertain data. Big Data Res. **1**, 23–36 (2014)
2. Flesca, S., Furfaro, F., Parisi, F.: Consistency checking and querying in probabilistic databases under integrity constraints. J. Comput. Syst. Sci. **80**, 1448–1489 (2014)
3. Cao, P., Wang, Z.: Efficient top-k query calculation in distributed networks. In: Proceedings of the Twenty-third Annual Association for Computing Machinery Symposium on Principles of Distributed Computing, PODC'04, pp. 206–215. Association for Computing Machinery, New York, NY, USA (2004)
4. AbdulAzeem, Y.M., ElDesouky, A.I., Ali, H.A.: A framework for ranking uncertain distributed database. Data Knowl. Eng. **92**, 1–19 (2014)
5. Soliman, M.A., Ilyas, I.F., Ben-David, S.: Supporting ranking queries on uncertain and incomplete data. Int. J. Very Large Data Bases **19**, 477–501 (2010)
6. Li, F., Yi, K., Jestes, J.: Ranking distributed probabilistic data. Assoc. Comput. Mach. 1–13 (2009)
7. Ye, M., Liu, X., Lee, W.-C., Lee, D.L.: Probabilistic top-k query processing in distributed sensor networks. In: Proceedings of the 26th IEEE International Conference on Data Engineering, pp. 585–588. IEEE Computer Society, Los Alamitos, CA, USA (2010)
8. Akbarinia, R., Pacitti, E., Valduriez, P.: Best position algorithms for top-k queries. Assoc. Comput. Mach. (2007)
9. Li, F., Yi, K., Jestes, J.: Ranking distributed probabilistic data. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD'09, pp. 361–374. ACM, New York, NY, USA (2009)
10. Ge, T., Zdonik, S., Madden, S.: Top-*k* queries on uncertain data: on score distribution and typical answers. In: Association for Computing Machinery (2009)

Part V
Applications in Image Processing

Texture-Based Watershed 3D Medical Image Segmentation Based on Fuzzy Region Growing Approach

Rajaram M. Gowda and G.M. Lingaraju

Abstract In this paper, a hybridization technique for multi-dimensional image segmentation algorithm is proposed. This algorithm is the combination of both the region growing and texture-based morphological algorithm of watersheds. An edge-preserving statistical noise reduction method is utilized as a preprocessing phase to calculate a perfect estimate of the image gradient. After that, a preliminary segmentation of the images into primitive regions is generated by employing the region growing. Then, watershed is applied. There are some drawbacks in medical mage study, when watershed is employed after the region growing. The main drawbacks are: over segmentation, sensitivity to noise and incorrect identification of thin or low signal to noise ratio structures. The main issue of over segmentation is controlled by texture local binary pattern (LBP). In addition, this paper has presented experimental outcomes achieved with two-dimensional/three-dimensional (2-D/3-D) magnetic resonance images. Numerical justification of the experimental outcomes is presented and demonstrated the efficiency of the algorithm for segmenting the medical image.

Keywords Local binary pattern (LBP) · Watershed · Medical image analysis

1 Introduction

Segmentation of an image is a way of splitting an image in a semantically meaningful manner. This imprecise definition indicates that the generality of the difficulty in segmentation can be observed in any image-driven process, e.g., fingerprint/text/face detection, anomalies detection in industrial electrical cables,

R.M. Gowda (✉) · G.M. Lingaraju
Department of Information Science & Engineering, MS Ramaiah Institute of Technology, Bangalore, India
e-mail: raj.gowda@gmail.com

G.M. Lingaraju
e-mail: gmlraju@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_23

tracking of moving people/cars/airplanes, etc. This methodology was broadly used in different departments, such as medical, traffic, police, satellite, and industries. An interesting source of images is the medical field. Here, imaging modalities, for example, computed tomography (CT) [1], magnetic resonance imaging (MRI) [2], and positron emission tomography (PET) [2, 3], generate a large quantity of image information. Not only the size and resolution of the images grow with improved technology, but at the same time, the total number of dimensions' increases. To obtain compatibility in dimension and to enhance workflow efficiency, Digital Imaging and Communications in Medicine (DICOM) [4] standard is used in healthcare environments worldwide [5, 6].

Medical data (image) are represented in the computer in the form of raw by the arrays of numbers like DICOM. These numbers specifying the values of appropriate physical sizes display the differences among dissimilar kinds of body tissue. The processing and examination of medical images are convenient in converting raw images into a quantifiable symbolic [2, 7] form. The main issue in the analysis of medical image is segmentation that recognizes the boundaries of body entities [8, 9]. The segmentation outcomes make it possible for shape analysis, discovering change in volume, and making an accurate radiation therapy treatment strategy. Recent medical data are not only in two-dimensional, but 3D image volumes are common in everyday practice [10, 11].

2 Related Work

Lewis et al. [12] have investigated watershed segmentation which was marker-controlled, for breast tumor candidate's identification. They did not employ watershed segmentation right on mammograms. Instead of this, they taken a morphological way to removal of low level pixels and then determined foreground and background markers. This thing overcomes the problematic side of oversegmentation and presented the watershed segmentation outcome that were more consistent.

Li et al. [13] have presented watershed segmentation to decrease oversegmentation utilizing both preprocessing and postprocessing. The preprocessing phase is deriving a gradient image directly from the original image. On the other hand, the postprocessing phase produces the texture gradient image through the gray scale co-occurrences. They avoided oversegmentation problem by merging gradient and texture gradient images. Here, they only go with 2D medical data, but for 3D medical data, they have not achieved better result this algorithm.

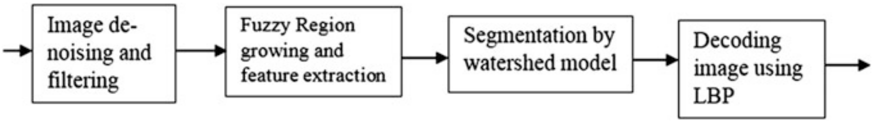


Image de-noising and filtering
Fuzzy Region growing and feature extraction
Segmentation by watershed model
Decoding image using LBP

Fig. 1 Block diagram

3 Proposed Methodology

There are a number of approaches for the segmentation of a medical image exists. In this approach, we have proposed a hybrid technique for the segmentation of the lung image. The method is based on the fuzzy region growing algorithm and watershed model based on the texture of the image. The edge detection is done by the region growing algorithm. Direct utilization of watershed segmentation habitually takes toward the oversegmentation. This thing happens because of the noise and other local irregularities that lead to huge quantity trivial regional minima. Considering this delimit, the image is to be preprocessed before the application of the proposed algorithm. Block diagram details about methodology are shown in Fig. 1.

3.1 Denoising and Enhancement of the Image

Noise that is present in the image degrades the segmentation process resulting in under segmentation or oversegmentation. Preprocessing refers about the increment in the input image quality by filtering the unwanted artifacts and decreasing the noise. This facilitates for improvement in the visibility of nodules. Numerous filters are utilized in the preprocessing stage, such as the median filters, morphological filters, and selective enhancement filter. The Digital Imaging and Communications in Medicine (DICOM) images require to be reconstructed to increase the resolution by denoising the original images.

The background noises, for example, black background and artifacts, contribute to the inefficiency in identifying the required characters in the image. The purpose of the preprocessing is to eliminate background and unwanted artifacts from the DICOM images. In preprocessing of an image, the foremost point is to detect the region of interest and crop out the unwanted portion of the image.

3.2 Integrating Features for Segmentation by Fuzzy Region Growing Method

After completing preprocessing, the fuzzy region growing method is utilized to apply for further process. There is a necessity to develop algorithms for the segmentation of the DICOM image in three dimensions. Difficulty in the segmentation rises from the variability that is inherent in the biological images and the complexity in the appearance. The main aim is to develop computationally effective algorithms that are improved versions of the previous methods. The fuzzy region growing algorithm is used for initial clustering. The main aim of the algorithm is to split the original image into the numerous homogeneous disjoint region as:

$$f = \bigcup_{j=1}^{n} R_j, \quad R_j \cap R_k = \phi \quad j \neq k, \tag{1}$$

where I represent the image and R the disjoint regions of the image, indicating that the image is composed of the numerous homogeneous disjoint regions. The grouping of pixels or the sub regions into a larger region on the basis of the homogeneity criterion is essential in the region growing process. To conserve the true edges, the unnecessary growth of the regions is restricted by employing the boundary edge. The first intensity feature is given by

$$diff = |g_{ave}(R_K) - g(i,j)|. \tag{2}$$

The difference between the average value of the intensity and the intensity value of a pixel under consideration is estimated. If the difference is small, then the intensity merges with the neighborhood else it does not merge. The gradient is calculated to achieve an accurate segmentation at lung region. The region growing algorithm avoids the smaller regions which consist of one or two pixels, since it prefers large regions.

3.3 Watershed Model for Segmentation

In the proposed methodology, watershed algorithm is implemented for object separation. The input gray level image is considered as the topographic surface. The previous algorithm does not provide cues for the separation of the touching objects.

The texture gradient of an image is obtained simply by calculating the gradient of the sub band magnitude and then summing them. The cleaned image from the previous step is obtained to generate two edge images. These images are utilized to analyze the gradient image in topographic surface. However, the texture extraction gives high energy values over the non-textured image intensity boundary. The double edge intensity boundary is resulted by the gradient of the sub-band magnitude. Therefore, the gradient of every sub-band is aimed at step detecting and not edge detection. This makes gradient extraction a simple method to execute the separable filtering on the magnitude of the images and the gradient operator can be describes as shown below:

$$\Delta gf = (gf \oplus E) - (gf \Theta E), \tag{3}$$

where $\oplus$ and Θ specify dilation and erosion correspondingly the structuring element is represented as E. Multi-scale gradient is defined for the structuring element when E_i denote a group of square structuring elements having the size (2i + 1) (2i + 1) pixels.

$$\Delta gf = \frac{1}{n}\sum_{i=1}^{n} [((gf \oplus E_i) - (gf \Theta E_i)) \Theta E_{i-1}] \tag{4}$$

The gradient image is utilized to represent the characteristics of the local regions to increase the correctness. The watershed is employed on the gradient of the image for better segmentation. Δgf is an element of the space $C(D)$ of a connected domain D and the topological distance in between the point a and b is:

$$T_{gf}(a, b) = \inf_{\gamma} \int \|(\Delta gf)\gamma(s)\| ds. \tag{5}$$

Infimum $\inf_{\gamma}$ is the overall path γ inside the domain defines the watershed. The image have minima $\{m_k\}_{k \in I}$, for some index I and to calculate the catchment basin $CB(m_i)$ of minimum m_i is defined by a set $C \in D$. The catchment basins $CB(m_i)$ is a set of points x in the domain D which are topographically closed to m_i than any other regional minimum m_j.

$$CB(m_i) = \left\{x \in D \middle| \forall_j \in I \backslash \{I\}: \Delta f(m_i) + T_f(x, m_i) < \Delta f(m_j) + T_f(x, m_j)\right\} \tag{6}$$

The watershed is then given by Eq. 7. This equation represents the set of points of an image that do not belong to any catchment basin.

$$W_{\text{shd}}(gf) = D \cap \left(\bigcup_{i \in I} CB(m_i)\right) \tag{7}$$

It is sensitive to imaging noise and may result in over segmentation. Hence, postprocessing is necessary after the implementation of the algorithm on the image. The watershed is obtained which consists of the enhanced textural properties of the original image. Finally, segmentation with smooth boundaries is obtained by the watershed transform. Once the segmentation of the image is completed, the texture of the segmented region is analyzed by the LBP operator.

3.4 LBP for Tumor Analysis

Texture is an essential characteristic of an image, and its analysis is an important aspect. An efficient nonparametric texture analysis based on the LBP has been developed. This texture examination is the measure of the grayscale invariance in a neighborhood. The image is composed of micro patterns which forms the basis for the LBP. The circular derivative of the patterns is produced thru concatenating the binary gradients in the LBP. The edge distribution and the other features in an image are contained in the histogram of these micro patterns. The invariant local grayscale variations information in the image is extracted by the LBP.

The joint distribution of the gray scale image is represented in:

$$T = t\left(gr_c, gr_0, \ldots, gr_{p-1}\right), \tag{8}$$

where T is texture of grayscale image. For $p + 1$ image pixels, $p > 0$, gr_c is center pixel value of grayscale image. $gr_p(p = 0, \ldots, p - 1)$ is the gray values of p similarly spaced on the circle f radius 'R'. (x_c, y_c) is denoted as the coordinates of the center pixel. The correlation between the pixels declines with the distance and textural info in an image can be achieved from local neighborhood. The local texture T can be represented as a joint distribution of the value of the center pixel and the differences without the loss of information is presented as follows:

$$T = t(gr_c, gr_0 - gr_c, \ldots, gr_{p-1} - gr_c). \tag{9}$$

The distribution can be factorized by assuming that the differences are independent of gr_c:

$$T \approx t(gr_c)t(gr_c, gr_0 - gr_c, \ldots, gr_{p-1} - gr_c), \tag{10}$$

where
$t(gr_c)$ total luminance of the image related to global image texture.

The independent assumption may not hold true always in practice. Maximum or minimum values of gr_c will decrease the range of possible differences due to the limited nature gr_c. The shifts in the grayscale permits one to get invariance with a possible small loss of information. The useful information for texture analysis is not derived hence a joint difference distribution has to be defined. Eq. 11 gives the joint distribution of textural characteristics.

$$T \approx t(gr_c, gr_0 - gr_c, \ldots, gr_{p-1} - gr_c) \tag{11}$$

To get the texture patterns, the dimensional difference distribution of pixel is calculated. The difference closely near to zero for constant or slow varying regions. The difference is larger at the edge compare to the slow varying region due to gray scaling shift. Monotonic transformation of the grayscale image represented in Eq. 12 based on the difference invariance:

$$T \approx t(s(gr_0 - gr_c), \ldots, s(gr_{p-1} - gr_c)), \tag{12}$$

where $s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$

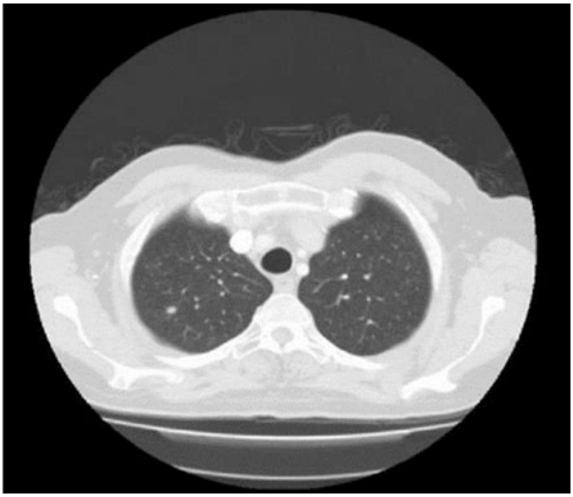


Fig. 2 Input medical image

To get a unique LBP code, binomial weight 2^p multiply with every neighborhood difference and the equation is given as:

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{p-1} s(gr_p - gr_c)2^p. \tag{13}$$

The above equation characterizes the code of the local image texture around (x_c, y_c). LBP code for uninterrupted 2^p bit discrete distribution as follows Eq. 14 after eliminating interrupted signals.

$$T \approx t(\mathrm{LBP}_{P,R}(x_c, y_c)) \tag{14}$$

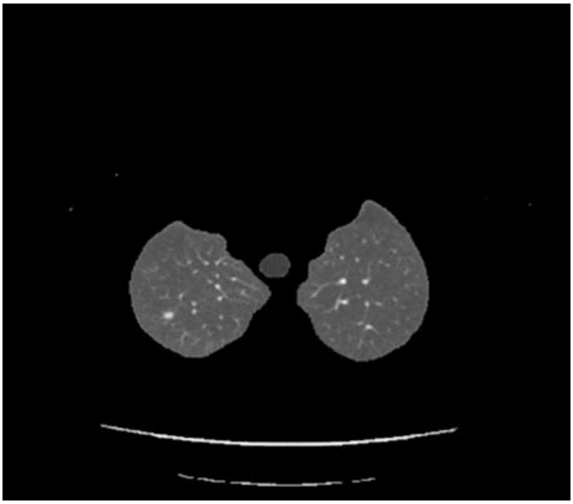


Fig. 3 Separate *Left/Right* lung portion using fuzzy region growing

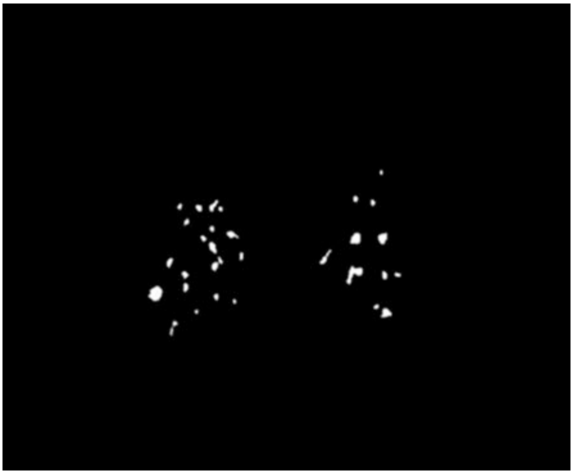


Fig. 4 Detection of the possible partition nodule using watershed

Assuming $M \times N$ image samples are given and $x_c \in \{0, \ldots, N-1\}$, $y_c \in \{0, \ldots, M-1\}$. Only central portion is considered in LBP distribution. A big neighborhood cannot be utilized in the borders due to spatial constraints. In the decided region, the LBP is computed for every pixel and the feature vector is the distribution of the codes denoted by S:

$$S \approx t(\mathrm{LBP}_{P,R}(x, y)). \tag{15}$$

The process of identifying the location of nodules and their type in the lung is known as nodule detection. It is dependent on the accuracy of segmentation methods that are employed. A nodule is almost spherical shaped surrounded by parenchyma, and its shape can be deformed by the neighboring surface.

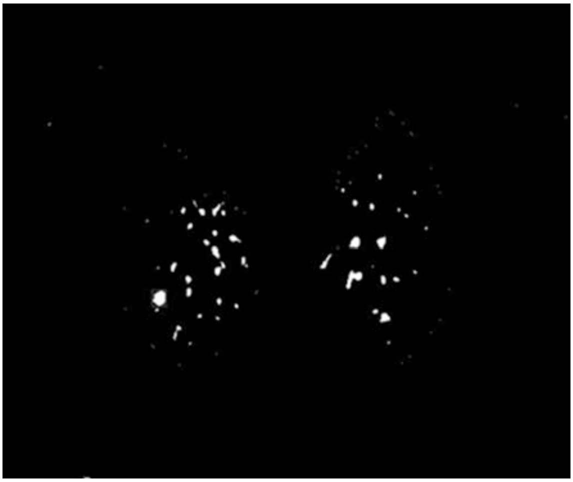


Fig. 5 Locate tumor portion using LBP

Table 1 Comparative results of detected tumor region

GLCM	Centroid	Eccentricity	Orientation	EquivDiameter	Solidity	Extent	Perimeter	Run time in (s)
Image 1	[6.6, 8.3]	0.68	7.8	1.28	0.93	0.71	4.17	4.2
Image 2	[8.1, 5.4]	0.70	−2.8	1.08	0.88	0.64	3.81	4.8
Image 3	[7.4, 6.5]	0.59	6.3	1.10	0.94	0.68	3.64	3.2
Image 4	[3.6, 4.5]	0.74	−7.1	7.2	0.95	0.65	2.21	3.9

4 Experimental Setup and Result

We implemented our proposed approach of fuzzy region growing method with texture based watershed algorithm to sample of 2-D MR lung images and got conventional segmentation maps of them. We assessed the efficiency of our proposed method by making comparison among the number of partitions in the segmentation map achieved through utilizing our proposed approach against the segmentation maps achieved through utilizing the traditional watershed algorithm. These methods identify the tumor region in 44 segmentations.

Here, Fig. 2 taken as the input image in our research. The application of fuzzy region growing before employing our texture based watershed segmentation algorithm has accomplished the aim to decrease one issue of oversegmentation when applied to MR images. The outcome image is shown in Fig. 3. After employing, only our proposed watershed method to the image in Fig. 4 has produced a final segmentation map consisting of 65 partitions. This can be detected by the visual examinations of the segmentation outcomes that there is no visible undersegmentation. Figure 5 shows the final nodule portion, which is achieved by selecting the best (minimum) set of training nodules of 65 partitions which is obtained from the watershed algorithm. The training of nodules is processed by the LBP methodology.

Table 1 shows the comparative result of detected tumor region based on centroid, eccentricity, orientation, solidity, extent and perimeter. Solidity, perimeter, and eccentricity show the size of lung nodule. If the value of perimeter is less than 5, then it is a small cell nodule, whereas it is large cell nodule if it is greater than 5. EquivDiameter shows the location of lung nodule. If the value is higher than 4, it is located in right side, else it is located in left side. In our test images, three images have nodule on the left side, and one image has it on the right side. Our proposed algorithm gives 90 % better result than the previous methods using four set of images in different segmentation map.

5 Conclusion

A region-based method through the watershed segmentation based on a texture based classification was presented. This new segmentation algorithm was employed for both the 2-D and 3-D medical data and provided adequate outcomes both with regard to segmentation performance and processing times. The requirements for memory are quite a little bit high because of the watershed detection algorithm. As a final point, even though the proposed segmentation methodology was verified quite appropriate for complex 3D anatomical structure, the utilization of further improved segmentation for complex medical data might produce better outcomes. LBP was used for analysis and correction of the resultant tumor portion.

References

1. Hua, P., Song, Q., Sonka, M., Hoffman, E., Reinhardt, J.M.: Segmentation of pathological and diseased lung tissue in CT images using a graph-search algorithm. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 2072–2075. IEEE (2011)
2. Marshall, H.R., Prato, F.S., Deans, L., Théberge, J., Thompson, R.T., Stodilka, R.Z.: Variable lung density consideration in attenuation correction of whole-body PET/MRI. J. Nucl. Med. **53**, 977–984 (2012)
3. Hofmann, M., Bezrukov, I., Mantlik, F., Aschoff, P., Steinke, F., Beyer, T., Schölkopf, B.: MRI-based attenuation correction for whole-body PET/MRI: quantitative evaluation of segmentation- and atlas-based methods. J. Nucl. Med. **52**, 1392–1399 (2011)
4. Presti, G.L., Carbone, M., Ciriaci, D., Aramini, D., Ferrari, M., Ferrari, V.: Assessment of DICOM viewers capable of loading patient-specific 3D models obtained by different segmentation platforms in the operating room. J. Digit. Imaging 1–10 (2015)
5. Subudhi, B.N., Patwa, I., Ghosh, A., Cho, S.B.: Edge preserving region growing for aerial color image segmentation. In: Intelligent Computing, Communication and Devices, pp. 481–488. Springer India (2015)
6. Li, J., Bioucas-Dias, J.M., Plaza, A.: Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. IEEE Trans. Geosci. Remote Sens. **50**, 809–823 (2012)
7. Yang, L., Georgescu, B., Zheng, Y., Wang, Y., Meer, P., Comaniciu, D.: Prediction based collaborative trackers (PCT): a robust and accurate approach toward 3D medical object tracking. IEEE Trans. Med. Imaging **30**, 1921–1932 (2011)
8. Chen, J., Li, J., Pan, D., Zhu, Q., Mao, Z.: Edge-guided multiscale segmentation of satellite multispectral imagery. IEEE Trans. Geosci. Remote Sens. **50**, 4513–4520 (2012)
9. Chan, T.E., Vese, L.: A level set algorithm for minimizing the Mumford-Shah functional in image processing. In: IEEE Workshop on Variational and Level Set Methods in Computer Vision, pp. 161–168. IEEE (2001)
10. Ao, J., Mitra, S., Long, R., Nutter, B., Antani, S.: A hybrid watershed method for cell image segmentation. In: IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), pp. 29–32. IEEE (2012)
11. Park, S.H., Lee, S., Yun, I.D., Lee, S.U.: Hierarchical MRF of globally consistent localized classifiers for 3D medical image segmentation. Pattern Recogn. **46**, 2408–2419 (2013)
12. Lewis, S.H., Dong, A.: Detection of breast tumor candidates using marker-controlled watershed segmentation and morphological analysis. In: IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), pp. 1–4. IEEE (2012)
13. Li, B., Pan, M., Wu, Z.: An improved segmentation of high spatial resolution remote sensing image using Marker-based watershed algorithm. In: 20th International Conference on Geoinformatics (GEOINFORMATICS), pp. 1–5. IEEE (2012)

Real Time Face Detection in Ad Hoc Network of Android Smart Devices

Mohammed Aljohani and Tanweer Alam

Abstract Android smart devices are ubiquitous in our daily life and becoming valuable device with the capabilities of wireless networking that are typically used with an IEEE 802.11 access point. Ad hoc network provides facilities to access devices in infrastructure less system without a centralized approach. Due to the open source of the development platform, Android was recently selected to be a supported operating system within this evolving and maturing technology delivery paradigm. Android based smart devices are dynamically joined to each other and create an ad hoc network on their own. Android based Smart devices are able to communicate with each other without requiring a dedicated wireless access point. The main goal of this paper is to identify faces and communicate in real time using ad Hoc network of Android based smart devices. The problem of communication is authentication in ad hoc network of android devices. Our study propose face detection technique to provide the solution of authentication in real time communication. In this research we implement HaarCascade technique to detect faces in real time. we develop android application to communicate in real time and detect all faces of connecting ad hoc network. The proposed system have been implemented in android ad hoc network system for authenticate the devices using face detection technique. Application has been developed and tested in ad hoc network environment of android based devices.

Keywords Wireless networking • Ad hoc network • Android smart devices • Face detection • HaarCascade technique • Communication

M. Aljohani (✉) · T. Alam
College of Computer Science and Information System, Islamic University,
Madinah, Saudi Arabia
e-mail: mohammed.aljohani@iu.edu.sa

T. Alam
e-mail: tanweer03@iu.edu.sa

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_24

1 Introduction

Android is the first open source and free platform for cell phones and is produced by individuals from the Open Handset Alliance [1]. The Open Handset Alliance is a gathering of more than 40 organizations, including Google, ASUS, Garmin, HTC and so on. These organizations have meet up to quicken and enhance the advancement of cell phones. Presently android smart devices are increasing exponentially in the world [2]. These smart devices have strong multimedia features. These features are helpful too much to the android user. The android users use these features to share multimedia quickly. Many android clients are increasing their interest to share videos and take photos by embedded camera. Also the popularity of android devices is increasing in the developing projects of entertainment. In this project clients upload their latest photographs to the system. Presently, the android smart devices are very popular with addition of its high capability [3]. The newly feature of Android devices are Wi-Fi Direct. Using this feature the wireless technologies provide support to their users to make the very good use of ad hoc network with smart devices at all time and everywhere [4, 5].

The ad hoc network communication of android smart devices can play most important role when cellular network fails [6, 7]. It works without cellular network. The smart devices will connect in the range of Wi-Fi wireless network [8, 9]. All Android smart devices within the range can communicate with each other without cellular network. I mean communication in own created network. This own created network is the special network without centralized approach i.e. Ad Hoc Network [10, 11]. Figure 1 represent the ad hoc network among some Android based smart devices without cellular networks [8, 10].

Our proposed system authenticates the communication in real time using face detection technique. The system works using android based application. The

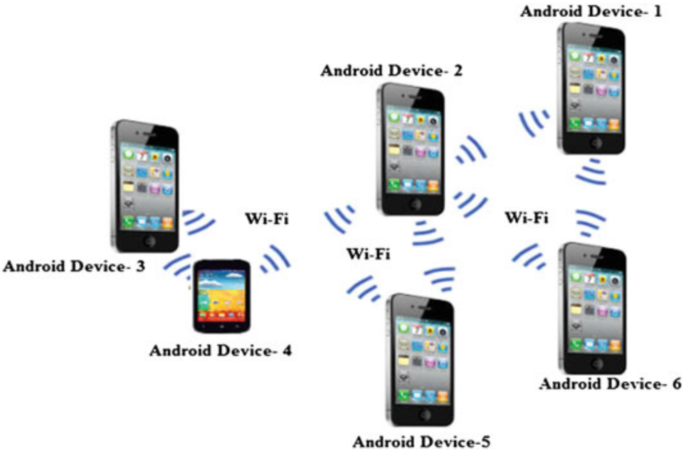


Fig. 1 Ad hoc network of Android devices

android application must be installed on all smart devices. The application needs Wi-Fi network to create the ad hoc network without centralized approach. These devices use the waves of wireless network to connect in ad hoc mode and sending their authenticated data using face detection technique embedded in android based smart devices.

2 Android Architecture in Ad Hoc Environment

Android architecture provides built-in tools to android applications for mobile smart devices. It means that the programmers need only to develop application using Android operating system and they can run these applications on different smart devices that powered by Android [12].

Android keeps running on Linux under Dalvik VM. Dalvik has an in the nick of time compiler where the byte code put away in memory is ordered to a machine code. Byte code can be characterized as ‘middle level’. JIT compiler peruses the byte code in numerous segments and accumulates progressively with a specific end goal to run the project quicker. Java performs keeps an eye on distinctive parts of the code and in this manner the code is gathered just before it is executed. When it is compiled once, it is stored and set to be prepared for later uses. We should first present some fundamental segments of the Android advancement API.

The terms are described shortly, focusing on just their role.

2.1 Activity

An Activity is a GUI segment of the application. Each movement comprises of a design, where diverse gadgets are found (different buttons, text fields, etc.). Every operation of a client is translated to a certain action in the activity.

2.2 Content Provider

A content provider is a part giving an interface to an organized arrangement of information. Content suppliers are the standard API for multi-process correspondence in the Android environment. In our application, a content provider will be utilized to access the phone’s contact list.

2.3 Service

Service component is an application that used in performing some work in the background without prompting the user. A service may run when the application itself is in the background [13]. By default, service does not create its own thread and does not run in a separate process. Nevertheless, in our application, the core services will run separate threads. For example, we use services for routing.

2.4 Intent Service

An intent service is a special type of service, running on its own thread. Intent services are used for handling asynchronous requests. For example, we use an intent service for neighbor discovery, as the Android API publishes asynchronous intents informing of device discovery [14].

2.5 Broadcast Receiver

A broadcast receiver is an application component designed to register and handle intents published in the system. The publisher for the intent may be another component in our application, or an external component from another application.

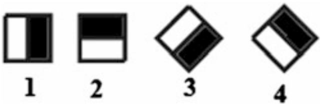
2.6 Intent

Intents are the standard communication protocol between different application components. Intents may be notifications of certain events (usually called “actions”) read by broadcast receiver, a structured set of data passed to an activity when it is started/resumed/stopped or a command to perform a certain action when passed to a service. For example, when the user wants to reply to a message he read in his inbox, the inbox activity will send an intent to the compose activity with the recipients’ ID.

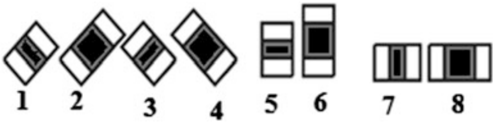
3 Implement HaarCascade Technique

The face detection technique is a technique that is used to detect facial features and convert the location and size of human face in digital image [15, 16]. It determines only face features and ignores anything else. It determines the location and size of total faces in a particular area [17, 18]. Figure 2 represents the HaarCascade technique. HaarCascade Technique is currently using the following Haar-like features which are the input to the basic classifiers:

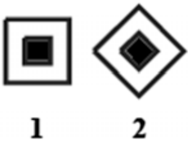
- Edge Features



- Line Features



- Centeroid Features



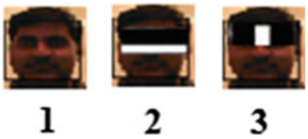


Fig. 2 HaarCascade technique

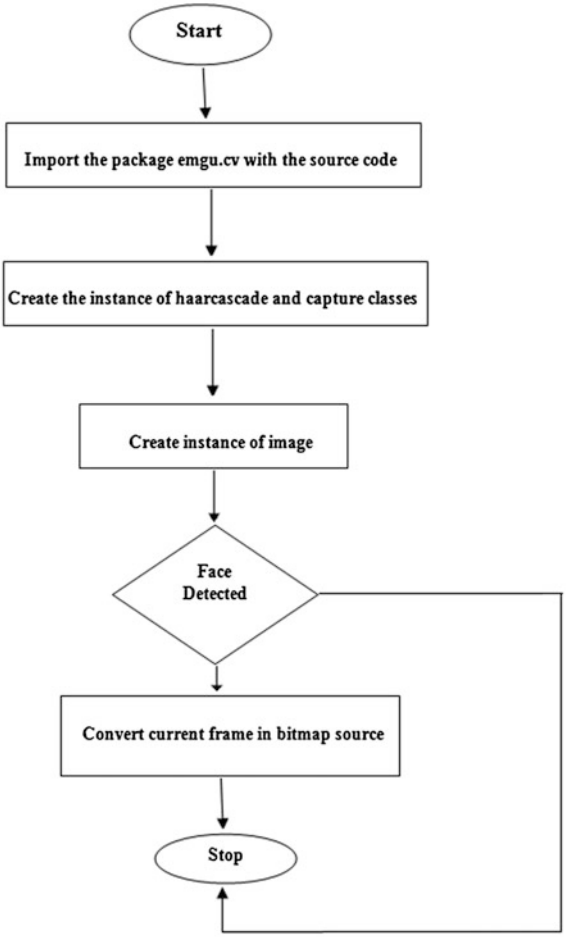
We apply all features in a window, group the features into different stages of classifiers and apply one-by-one. If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. The implementation coding steps are as follows. Figure 3 represents the flowchart of HaarCascade technique.

1. Import the package Emgu.CV with the source code.
2. Create the instance of HaarCascade and Capture classes
3. Create instance of Image as follows.

   ```
   Image Frame1 = capture.QueryFrame();
   ```

4. Faces Detected.

Fig. 3 Flowchart of HaarCascade technique


Start
Import the package emgu.cv with the source code
Create the instance of haarcascade and capture classes
Create instance of image
Face Detected
Convert current frame in bitmap source
Stop

```
            if (Frame1 != null)
            {
                Image<Gray, Byte> Frame2 =
               Frame1.Convert<Gray, Byte>();
                var detectedFaces =

grayFrame.DetectHaarCascade(haarCascade)[0];
                foreach (var face in detectedFaces)
                {
                    Frame1.Draw(face.rect, new Bgr(0,
                    double.MaxValue, 0), 3);
                 }
```

5. Length of detected faces.

```
detectedFaces.Length;
```

6. Convert current frame in Bitmap Source.

```
image1.Source = ToBitmapSource(Frame1);
```

4 Testing on Android Devices

This algorithm is implemented for face detection in ad hoc networks of android devices. The android application will detect all faces and display to all devices in the range of the network. The Fig. 4 represents the real time face detection on android based device.

All detected faces are displayed on all android devices in an ad hoc network using Wi-Fi. So this technique will help us to secure communication among real time connected devices.

5 Result Interpretation

Real time face detection in ad hoc network for android device is programmed to execute on to the android devices. Android device must have in built 802.11 Wi-Fi and Bluetooth support. The idea will implement HaarCascade face detection

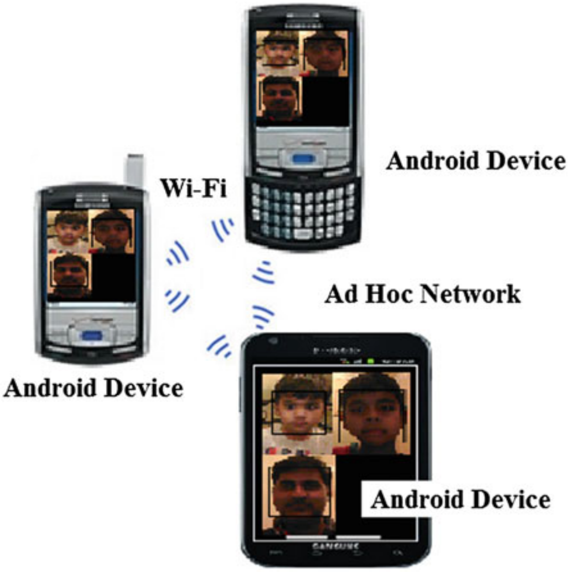


Fig. 4 Testing HaarCascade technique on android devices in ad hoc network

technique. We create an android application for communication among android devices that are in the range of Wi-Fi or Bluetooth area. These devices can communicate each other without cellular networks and each device will show faces of all connected users.

We compare the throughput of transmission in ad hoc network of Android devices in the range of Wi-Fi and Bluetooth.

In Wi-Fi ad hoc network communications, the maximum throughput is 0.42 Mbps, 2.03 Mbps, 4.22 Mbps, and 21.41 Mbps for file size 100 KB, 500 KB, 1 MB and 5 MB, respectively (Table 1).

In Bluetooth ad hoc network communications, the maximum throughput is 0.48 Mbps, 2.52 Mbps, 2.22 Mbps, and 10.5 Mbps for file size 100 KB, 500 KB, 1 MB and 5 MB, respectively (Table 2).

When Android device A sends data through an ad hoc network, each file is stored and forwarded in the intermediate Android device B, which adds a delay to the delivery of packet. This delay includes the propagation delay, transmission delay, queuing delay and processing delay. The delay has more significant impact for larger sending rates. When the intermediate device B cannot keep up, the sender will retransmit unacknowledged packets more often, leading to lower throughput. Figure 5 represents the throughput for file size 100 KB in the range of wi-fi or Bluetooth. Figure 6 represents the throughput for file size 500 KB in the range of wi-fi or Bluetooth. Figure 7 represents the throughput for file size 1 MB in the range of wi-fi or Bluetooth. Figure 8 represents the throughput for file size 5 MB in the range of wi-fi or Bluetooth.

Table 1 Throughput in ad hoc network in Wi-Fi Communication

File size	Mbps
100 KB	0.42
500 KB	2.03
1 MB	4.22
5 MB	21.41

Table 2 Throughput in ad hoc network in Bluetooth Communication

File size	Mbps
100 KB	0.48
500 KB	2.52
1 MB	2.22
5 MB	10.5

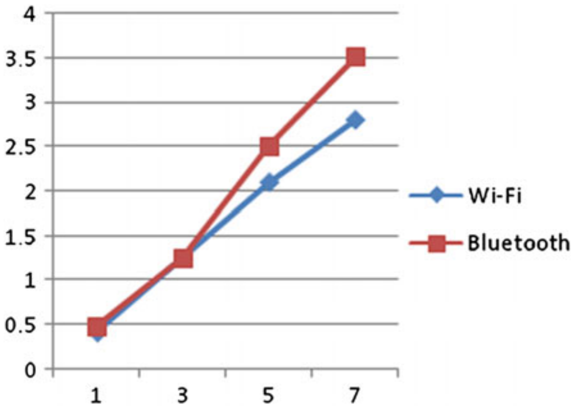


Fig. 5 Throughput for file size 100 KB in Wi-Fi and Bluetooth range

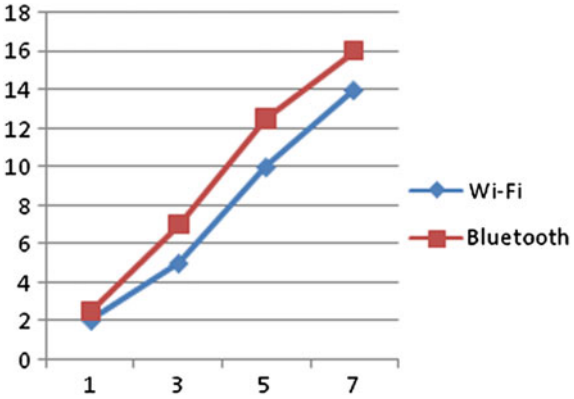


Fig. 6 Throughput for file size 500 KB in Wi-Fi and Bluetooth range

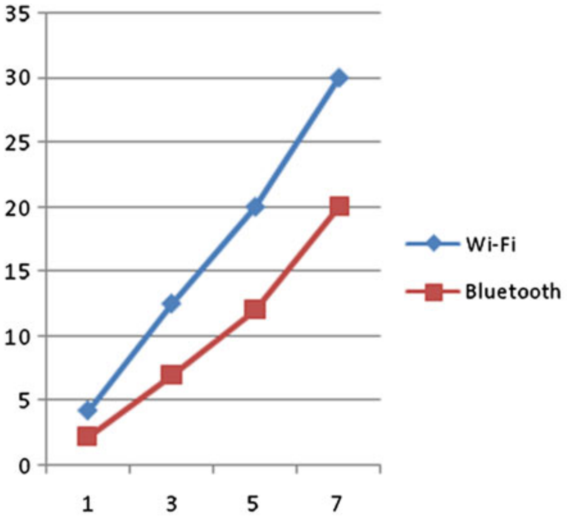
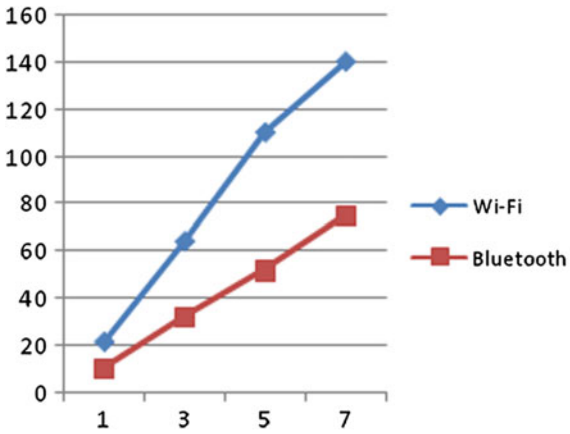


Fig. 7 Throughput for file size 1 MB in Wi-Fi and Bluetooth range

Fig. 8 Throughput for file size 5 MB in Wi-Fi and Bluetooth range



6 Conclusion

Ad hoc network is a self-organizing collection of wireless mobile nodes that form a temporary network without the aid of a fixed networking infrastructure or centralized administration. In ad hoc networks messages sent by a node may be received simultaneously by all nodes within its transmission range, i.e. by its neighbors. Messages requiring a destination outside this local neighborhood zone must be hopped or forwarded by these neighbors, which act as routers, to the appropriate target address. As a consequence of node mobility fixed source/destination paths cannot be maintained for the lifetime of the network. The need for developing face detection technique in an ad hoc network communication for Android-based mobile devices is becoming real. The face detection technique in ad hoc network will allow smart device users to communicate securely among each other in real time by using Android applications. The android application for connecting android devices and detect faces in ad hoc environment has been done and results in Wi-Fi and Bluetooth were collected. The fundamental functions of the proposed system have been introduced in android ad hoc network system. Application has been tested in Wi-Fi and Bluetooth ad hoc network environment. The results showed successful and expectation for future scope in the area of mobile ad hoc network and Image Processing.

References

1. Shabtai, A., Fledel, Y., Kanonov, U., Elovici, Y., Dolev, S., Glezer, C.: Google android: a comprehensive security assessment. Secur. Priv. 35–44 (2010)
2. Alam, T., Aljohani, M.: Design and implementation of an Ad Hoc Network among Android smart devices. In: Green Computing and Internet of Things (ICGCIoT), 2015 International Conference IEEE, pp. 1322–1327, 8 Oct 2015

3. Aljohani, M., Alam, T.: Design an M-learning framework for smart learning in ad hoc network of Android devices. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–5, 10 Dec 2015
4. Jabbar, W.A., Ismail, M., Nordin, R.: Framework for enhancing P2P communication protocol on mobile platform. In: Proceedings of the ICIA 12 Conference. Malaysia (2012)
5. Abduljalil, F.M., et al.: Integrated Routing Protocol (IRP) for integration Ad Hoc networks. In: Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06). 0-7695-2553-9/06 $20.00 © 2006 IEEE
6. Gardner-Stephen, P.: The Serval project: practical wireless ad-hoc mobile telecommunications. http://developer.servalproject.org/site/docs/2011/ServalIntroduction.html. Accessed 15 May 2013
7. Balfanz, D., Smetters, D.K., Stewart, P., Wong, H.C.: Talking to strangers: authentication in Ad-Hoc wireless networks. In: Proceedings of Network and Distributed System Security Symposium Conference (2002)
8. Saavedra, A.G., Serrano, P.: Device-to-device communications with WiFi Direct: overview and experimentation. IEEE Wirel. Commun. **20**(3) (2013)
9. Sumino, H., Ishikawa, N., Kato, T.: Design and implementation of P2P protocol for mobile phones, pp. 6–162 (2006)
10. Fodor, G., Dahlman, E., Mildh, G., Parkvall, S., Reider, N., Miklos, G., Turanyi, Z.: Design aspects of network assisted device-to-device communications. IEEE Commun. Mag. **50**(3), 170–177 (2012)
11. Jabbar, W.A.: Peer-to-Peer Communication on Android-Based Mobile Devices: Middleware and Protocols. 978-1-4673-5814-9/13/$31.00 ©2013 IEEE
12. Han, B.: Analysis and research of system security based on android. In: 2012 Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 581–584 (2012)
13. Alam, T., Aljohani, M.: An approach to secure communication in mobile ad-hoc networks of Android devices. In: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) IEEE, pp. 371–375, 28 Nov 2015
14. Aljohani, M., Alam, T.: An algorithm for accessing traffic database using wireless technologies. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4, 10 Dec 2015
15. Mahmoodi, M.R., Sayedi, S.M.: A face detection method based on kernel probability map q. Comput. Electr. Eng. (2015). http://doi.org/10.1016/j.compeleceng.2015.02.005
16. Yan, J., Zhang, X., Lei, Z., Li, S.Z.: Face detection by structural models ☆. IMAVIS **32**(10), 790–799 (2014). http://doi.org/10.1016/j.imavis.2013.12.004
17. Jin, J., Xu, B., Liu, X., Wang, Y., Cao, L., Han, L.: Signal processing : image communication a face detection and location method based on feature binding. Signal Process.: Image Commun. **36**, 179–189 (2015). http://doi.org/10.1016/j.image.2015.06.010
18. Zafeiriou, S., Zhang, C., Zhang, Z.: A survey on face detection in the wild: past, present and future R. Comput. Vis. Image Underst. **138**, 1–24. http://doi.org/10.1016/j.cviu.2015.03.015 (2015)

Detection of Copy-Move Image Forgery Using DCT

Choudhary Shyam Prakash, Kumar Vijay Anand and Sushila Maheshkar

Abstract With the advancements in computer technology digital image tampering like copy-move forgery has become frequent. In this paper, we present a novel DCT-based technique for detecting copy-move forgery. DCT is applied to each fixed-size overlapping block of image to represent its features. The dimension of the features is reduced using truncation. Then the feature vectors are lexicographically sorted and, duplicated image blocks will be neighboring in the sorted list. Thus duplicated image blocks will be compared in the matching step. To make the method more robust, a scheme to judge whether two feature vectors are similar is imported. Simulation results show that the proposed technique is capable of detecting the duplicated regions even when an image was distorted by JPEG compression, blurring or additive white Gaussian noise.

Keywords Copy-move forgery • DCT • Tampered region detection • Dimension reduction

1 Introduction

In present days, with the availability of many software (e.g., Photoshop) and applications we can edit a picture easily and modify it without leaving noticeable proofs, so it is difficult to believe that the given image is original or forged. It creates lots of problem primarily in courtroom witness, insurance claims, and scientific scams. Tampering can be of many types, however, covering any objects

C.S. Prakash (✉) · K.V. Anand · S. Maheshkar
Department of Computer Science and Engineering, Indian School of Mines,
Dhanbad 826004, Jharkhand, India
e-mail: shyamprakash2008@yahoo.com

K.V. Anand
e-mail: vijayanand.sikandar@gmail.com

S. Maheshkar
e-mail: sushila_maheshkar@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_25

from natural images is a common form of image tampering and this is known as copy-move forgery. Some researchers have developed different techniques to deal with copy-move forgery, all of them use square blocks for matching purpose. Fridrich [3] used DCT-based features instead of exhaustive search to detect region duplication, which is more effective, but their method is sensitive to variations in duplicated regions owing to additive noise. Later, Huang et al. [5] improved the performance by reducing the feature vector in dimension; however, they failed to consider the multiple copy-move forgery. In [2], Popescu proposed a new method by adopting the PCA-based feature, which can endure additive noise, but the detection accuracy is low. Luo [6] proposed color features as well as block intensity ratio to show the robustness of their method. Bayram et al. applied Fourier-Mellin transform (FMT) to each block [1], FMT values are finally projected to one dimension to form the feature vector. [7] used a method based on blur moment invariants to locate the forgery regions, and [8] took the advantage of the SIFT features to detect the duplication regions and their experiments show the robustness of their approach. Yet, the methods mentioned above have higher computational complexity, since the quantized square blocks are directly used for matching, that the dimension of feature vector is higher, as a consequence, affecting the efficiency of detection, especially when dealing with high-resolution digital images.

2 Proposed Technique

According to the region-cloning feature, two similar regions must be present in the tampered image. It is fact that a natural image hardly contains two similar regions, unless it contains two large smooth regions. So an image is said to be a copy-move tampered if there are two similar regions which has been detected in the image. It is assumed that the duplicated region is non-overlapping. Here image is divided into fixed-size overlapping blocks; the reason behind the division is to detect the duplicated region efficiently. As the shape and size of the duplicated regions are unknown and it is impractical to compute all possible pair of regions with different shape and size. To conduct a test for tampered image, it is required to represent each blocks by some features. By extracting features of all blocks, the detection of tampering is robust and effective. After extracting features of all blocks it is sorted lexicographically. As the similar regions have similar features vector. By applying the lexicographical sort, all the similar regions are sorted consecutively which make the matching process adequate. Matching process is applied on each pair of feature. If the features are matched then it is considered that the corresponding block is similar. Similarly, if many matched blocks are adjacent then there is copy-move tampering in the input image. To judge the tampering the detected region is marked in a map.

2.1 Algorithm Framework

The above discussed method for copy-move detection algorithm which is also shown in Fig. 1.

Steps for the detection algorithm are as follows: Division → Representation → Matching → Output.

In division the image is divided into sub-blocks. For division a square block of size bb slides along the image from upper left corner to the bottom right corner. Let us consider the size of image is $M \times N$. In this way $(M - b + 1)(Nb + 1)$ overlapping blocks are generated. In representation step each sub-block is represented using some suitable features. To extract the features of each block, DCT is applied on each block. The extracted DCT coefficients are reshaped in zigzag order so that all similar frequencies are grouped together. In this way, $(Mb + 1)(N - b + 1)$ row vectors are generated from all the blocks. Then, lexicographical sort is applied and a matrix A is generated of size $(M - b + 1)(N - b + 1) \times b^2$. In matching step each pair of blocks are matched. Let ith row of matrix A is denoted as a_i and the top-left corner's coordinate of the corresponding block is denoted as (x_i, y_i). Each pair of consecutive vectors is tested. If the vectors are same then the shift vector s between two corresponding blocks is calculated as:

$$s = (s_1, s_2) = (x_1 - x_2, y_1 - y_2). \quad (1)$$

Here, s and $-s$ correspond to the same shift, so for normalization it is necessary to multiplying s by -1 to make sure that $s_1 \geq 0$. Similarly algorithm search for all the shift vectors $s(1)$, $s(2)$, …, $s(n)$ whose occurrence exceeds a predefined threshold T. For output all the shift vectors found corresponding to the blocks is colored and mapped. The mapped region is considered as the duplicated region in the input image. By mapping the duplicated region, it is easy to determine whether the input image is tampered or original.

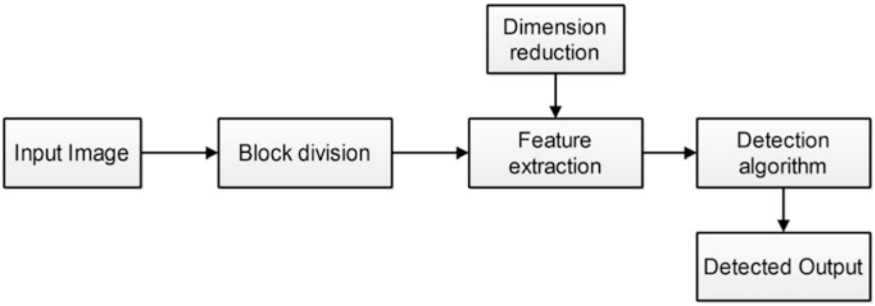


Fig. 1 Architecture of detection algorithm

2.2 Feature Dimension Reduction

In feature representation, DCT is applied on each block of size b × b to generate the feature vector of b2 dimension. According to the DCT properties, the energy of transformed coefficients is concentrated on the low frequency coefficients. Hence, the higher frequency coefficients can be pruner to reduce the dimension of the vectors. As the DCT coefficients have been reshaped to a row vector in zigzag order, pruning is easy by saving only some part of vector components. Specifically, a factor $p(0 < p \leq 1)$ is defined and only first $\lceil p \times b^2 \rceil$ DCT coefficients are saved for further processing. Due to dimension reduction, the matching process becomes faster. To achieve the robustness in our algorithm, a quantization factor q is introduced and each DCT coefficients is quantized by q then it is rounded to the nearest integer.

2.3 Matching

For matching, each row in the matrix A is tested with neighboring rows which fulfill the condition $j - i < N_f$ where N_f is a parameter that control the number of neighboring rows. To test the similarity of two vectors, the ratio of each corresponding components of two vectors is calculated. If the two vectors are correlated then the ratio will be same. If there is a case that the vector component is 0 and cannot be the divider, the test is switched to checking whether the pair of corresponding component is close enough. Here, we assumed that the duplicated region is not overlapping and the divided blocks may be overlapping, an additional judgement has to be made when a matching is detected, count only those shift vectors which are not too close. For this a parameter N_d is introduced. Only those shift vectors are counted which are generated by two similar feature vectors whose distance is larger than N_d.

2.4 Post-Processing

After calculating all the shift vectors $s_{(i)}$ whose appear frequency exceeds threshold T, a map image is initialized whose size is same as the input image and all the pixel values are 0. For each $s_{(i)}$, the matching blocks that contributed to this shift vector are marked in two special colors in the map, one color at the source and the other at destination. As we truncated some DCT coefficients for dimension reduction, there may be some false matching. For this a morphology operation [4] is applied to remove the isolated blocks.

2.5 Algorithm Flow

The algorithm flow for copy-move detection is as follows:

1. The input image is taken in gray scale image I of size M × N. If the image is color then convert it using standard formula $I = 0.299\text{R} + 0.587\text{G} + 0.11\ \text{B}$.
2. Divide the image into sub-blocks of size $b \times b$ to get $(M - b + 1)(N - b + 1)$ blocks.
3. Apply DCT on each block and reshape the $b \times b$ quantized coefficient matrix with a zigzag scan to a row vector. The vector is then pruned to only elements.
4. Apply the lexicographical sort and form a matrix A of size $(M - b + 1)(N - b + 1) \times (p \times b^2)$
5. For each row in A, test its neighboring rows which satisfy that whether they are similar.
6. If and come out to be similar, the distance between two corresponding blocks is calculated $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.
7. If $d > N_d$, the shift vector s is calculated and normalized. Then the shift vectors existing frequency plus 1.
8. If the shift vectors $s_{(i)}$ whose appear frequency exceeds threshold T, go to the next step, otherwise the algorithm exit.
9. Mark the detected region in map image for all $s_{(i)} > T$.

3 Experimental Results

In this section, we discussed the experiment procedure and parameters which is used in our algorithm and discussion of the results to prove the robustness and effectiveness of our algorithm.

3.1 Experiment Methods and Procedure

All the experiments were carried out on Matlab R2013a. All the images were 128 × 128 pixel grayscale images saved in jpg format. In our experiments, without specific specification, all the parameters in the experiment were set as: $b = 8$, $T = 35$, $N_f = 3$, $N_d = 16$, $p = 0.25$, $q = 4$, s threshold = 0.0625 by default. If our algorithm detects some tampering in input image then W would be set to 1, otherwise $W = 0$. When an input image is natural and our algorithm detects some duplicated region then it is called false alarm. Similarly, if our algorithm failed to detect any duplicated region in tampered input image then it is called miss alarm. Two counters, $C1$ and $C2$ which were initialized to 0, were also plus 1 when wrong

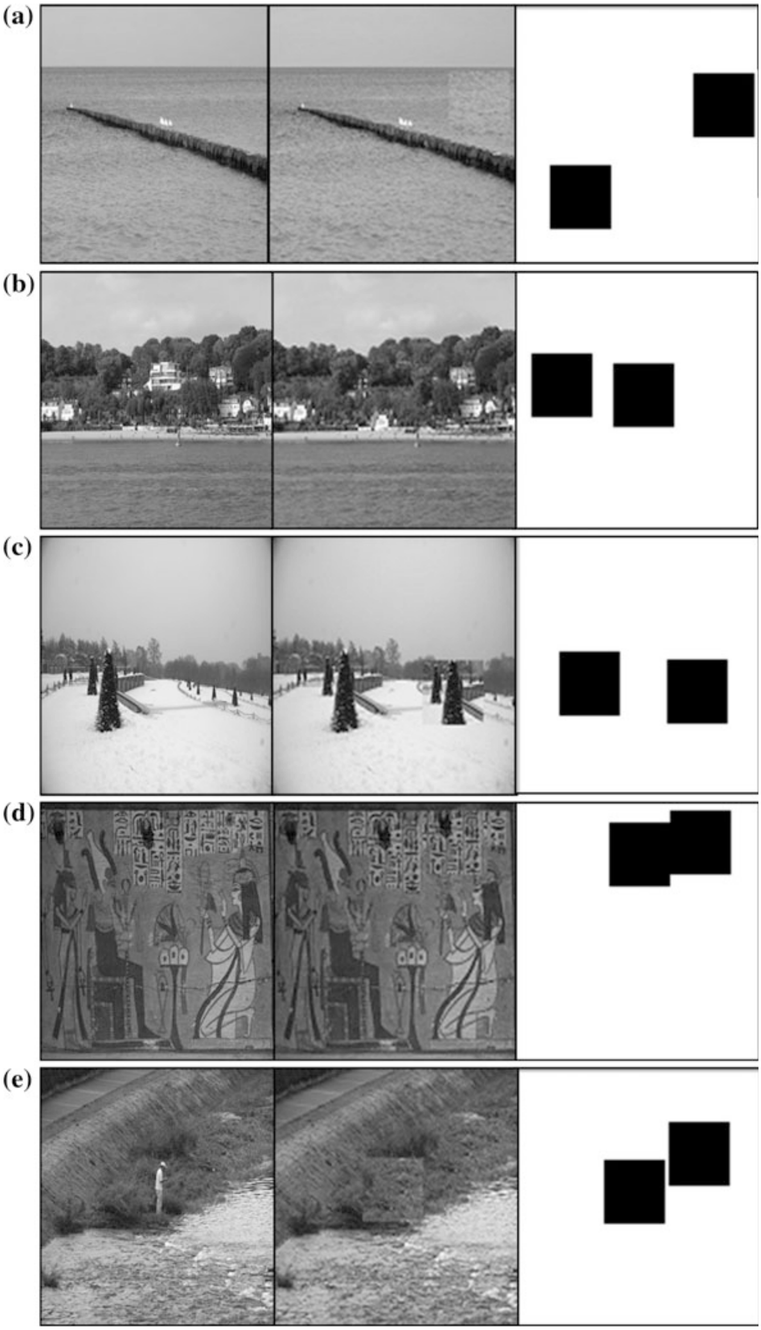


Fig. 2 Shown are the detecting results for tampered region of 32 × 32 size

detection or failure in detecting forgery happened, respectively. After the whole testing data set had been processed, false alarm rate far, miss alarm rate mar and classification accuracy rate car were calculated as the following formulas:

$$\mathrm{car} = 1 - \frac{W}{\mathrm{pics}}, \mathrm{far} = \frac{C1}{\mathrm{pics}}, \mathrm{mar} = \frac{C2}{\mathrm{pics}}, \tag{2}$$

where pics is the number of images used. These parameters can be used to evaluate the sensitivity of this algorithm.

To calculate the accuracy and robustness of our algorithm, we set duplicated region in tampered image as φ_s and detected duplicated region as $\tilde{\varphi}_s$, while the duplicated regions are detected by our method as φ_t and $\tilde{\varphi}_t$ The detection accuracy rate (DAR) and false positive rate (FPR) is calculated as

$$\mathrm{DAR} = \frac{|\varphi_s \cap \tilde{\varphi}_s| + |\varphi_t \cap \tilde{\varphi}_t|}{|\varphi_s| + |\varphi_t|}, \mathrm{FPR} = \frac{|\tilde{\varphi}_s - \varphi_s| + |\tilde{\varphi}_t - \varphi_t|}{|\tilde{\varphi}_s| + |\tilde{\varphi}_t|}, \tag{3}$$

where |.| means the area of region, ∩ means the intersection of two regions. The more DAR is close to 1 and FPR is close to zero, the more precise the method would be. The visual results of detected tampered images are shown in Fig. 2. Each image consist of three images: original image, tampered image, and detection result.

In the following examples the parameters which is defined previously is used. We perform the detection algorithm without any post-processing operation, results are shown in Table 1, where DAR, FPR and W (wrong classification rate) all are average value which has been tested on several images. In Table 2 the far, mar and car is shown on different post-processing operation performed on image. To test the robustness and effectiveness of our algorithm we perform some post-processing such as jpeg compression and Gaussian blurring operation on images and perform the detection operation. The results are shown in Tables 3 and 4. In most of the

Table 1 Detection results of the duplicated images without any post-processing, where DAR, FPR are defined in Eq. 3 and *W* is the wrong classification rate

Image size	16 × 16	24 × 24	32 × 32	40 × 40	48 × 48
DAR	1	0.9987	0.9945	0.9951	0.9964
FPR	0.035	0.029	0.016	0.009	0.005
W	0.011	0	0	0	0

Table 2 Detection results of the proposed method

	BMP	JPEG	PNG	Blur	AWGN
far	0.2	0.16	0.2	0.17	0.1
mar	0.1	0.14	0	0.019	0.1
car	0.88	0.85	0.89	0.82	0.8

Table 3 Detection results when images are distorted by JPEG compression ($Q = 90$)

Imagesize	24 × 24	32 × 32	40 × 40	48 × 48
DAR	0.8475	0.8519	0.8492	0.8488
FPR	0.2393	0.2387	0.2382	0.2374
W	0.0020	0.0017	0	0

Table 4 Detection results when images are distorted by Gaussian blurring

Imagesize	$w = 3, \sigma = 0.5$	$w = 3, \sigma = 1$	$w = 5, \sigma = 0.5$	$w = 5, \sigma = 1$
DAR	0.7092	0.6992	0.6887	0.6879
FPR	0.2985	0.2977	0.2956	0.2975
W	0	0	0	0

cases, the accuracy rate is 80 % which show that our algorithm detects the tampered region successfully. Hence, our algorithm is robust and efficient in detection of tampered images.

4 Conclusion

Here, we have presented a robust and effective algorithm for copy-move tamper detection. It is a passive method for tamper detection that means there is no any preceding knowledge about the tested image. Our method use less feature to represent each block than the existing methods [2, 3, 5] and the computational complexity is less. It is observed from the table that the DAR is more than 80 % in average cases, which shows the robustness and effectiveness of our method is acceptable. Hence, we believe our method could be useful in some area of forensic science.

References

1. Bayram, S., Sencar, H.T., Memon, N.: An efficient and robust method for detecting copy-move forgery. In: ICASSP 2009. IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 1053–1056. IEEE (2009)
2. Farid, A., Popescu, A.: Exposing digital forgeries by detecting duplicated image regions. Technical Report, TR2004-515, Department of Computer Science, Dartmouth College, Hanover, New Hampshire (2004)
3. Fridrich, A.J., Soukal, B.D., Lukáš, A.J.: Detection of copy-move forgery in digital images. In: Proceedings of Digital Forensic Research Workshop. Citeseer (2003)
4. Gonzalez, R.: Re Woods, Digital Image Processing. Addison (1992)
5. Huang, Y., Lu, W., Sun, W., Long, D.: Improved DCT-based detection of copy-move forgery in images. Forensic Sci. Int. **206**(1), 178–184 (2011)

6. Luo, W., Huang, J., Qiu, G.: Robust detection of region-duplication forgery in digital image. In: 18th International Conference on Pattern Recognition, 2006. ICPR 2006, vol. 4, pp. 746–749. IEEE (2006)
7. Mahdian, B., Saic, S.: Detection of copy–move forgery using a method based on blur moment invariants. Forensic Sci. Int. **171**(2), 180–189 (2007)
8. Pan, X., Lyu, S.: Detecting image region duplication using sift features. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1706–1709. IEEE (2010)

A New Learning-Based Boosting in Multiple Classifiers for Color Facial Expression Identification

Dhananjoy Bhakta and Goutam Sarker

Abstract A new method for automatic color facial expression identification using multiple classifier classification system has been developed. The system is primarily composed of three classifiers. The same color face features are then trained independently using three different classifiers. Now, a super classification technique integrates the decisions coming from each single classifier which outcomes as a final identified expression. To fuse the conclusion drawn by different classifiers, we apply a new technique of learning-based boosting which improves the complete system performance meaningfully in terms of recognition accuracy.

Keywords Facial expression identification · Multiple classifier · Knowledge-based method · MOCA · HBC · RBFN

1 Introduction

Apart from the face identification, automatic facial expression identification is also a current research area due to a number of applications, for example, human–computer communication, patient observing, audiovisual conferencing and consumer satisfaction readings. According to Mehrabian [1], the verbal part of a speech contributes only 7 % of the effect of any message; the vocal part pays about 38 % while facial expression contributed for 55 % of the effect of the message. So, facial expression plays a significant role in our social communication. Through revising person's emotion we can assume about attention, behavior, intention and physiological state of the person. Many researchers deal with seven basic facial expressions: neutral, happy, sad, fear, disgust, surprise and anger [2]. Facial expressions

D. Bhakta (✉) · G. Sarker
Computer Science and Engineering Department, NIT, Durgapur,
Mahatma Gandhi Avenue, Durgapur 713209, West Bengal, India
e-mail: bhaktadhananjoy@gmail.com

G. Sarker
e-mail: g_sarker@ieee.org

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_26

[3, 4] are noticed when major changes arise in forehead, eyebrows, eyes, nose and lip rather than in complete face. Generally, a person's eyes tell about in what way they are feeling, or what they are thinking. If we notice blink rate of person's eyes, we know about the nervousness of the person. Real smile always consists of crow's feet wrinkles, pushed-up cheeks, and movements from muscle that orbits the eye. So, if we noticed no such thing, we conclude that the smile of the person is fake. In the same way, in sad expression, eyebrows will be up, upper eyelids will be drooping and trailing focus in eyes. These types of facial expressions are displayed on the basis of facial muscle movements [5].

Numerous researchers have put efforts in recognition of facial expression. Most of the researchers follow the Facial Action Coding System (FACS) for automatic facial expression recognition developed by Ekman and Friesen [2, 6]. FACS is a combination of a set of facial muscle movements which can be describing the six basic emotions. Other researcher finds out some facial points known as region of interest (ROI), which may help them to classify the person's emotions.

This paper presents automatic identification of person's facial expression. This system mainly considers three classifiers, where one of them is knowledge based and other two are appearance based. These classifiers are used for color facial expression identification which identifies one from the five basic emotions such as neutral, happy, angry, sad and surprise where color information of the extracted facial part is employed as an added feature. In the proposed system, when a 2D facial image is served as input, first some image pre-processing steps are applied to extract facial component and reduce dimensionality of the input image [7, 8]. Then, knowledge-based classifier, Modified Optimal Clustering Algorithm (MOCA)-based classifier and Heuristic-based clustering (HBC) [3, 9, 10] based classifier are used to assign class lebel to these processed color features. Now, outcomes of these three classifiers are integrated through a new learning-based boosting technique to form robust decision as classified facial expression which is our desired output. Performance of the system is measured through Holdout Method [8], in which the classifier is build with the labeled instances (examples used during training) and performance of the constructed classifier using unlabeled instances.

The paper is organized as follows: Sect. 2 describes different types of classifier used in our system. Section 3 discusses the method of boosting multiple classifiers. Section 4 discusses the outline of the proposed system. In Sect. 5, KDEF [11] (http://www.emotionlab.se/resources/kdef) facial expression database is used in our experiment is discussed and Sect. 6 concludes the paper.

2 Multiple Classifiers

Generally, decision coming from a group of experts is more reliable than decision of a particular expert. Decision made by a sole classifier may not be accurate at all times, for what, we deal with many classifiers instead of taking a sole classifier. In multiple classifier system, every single classifier works on different features of the input. To

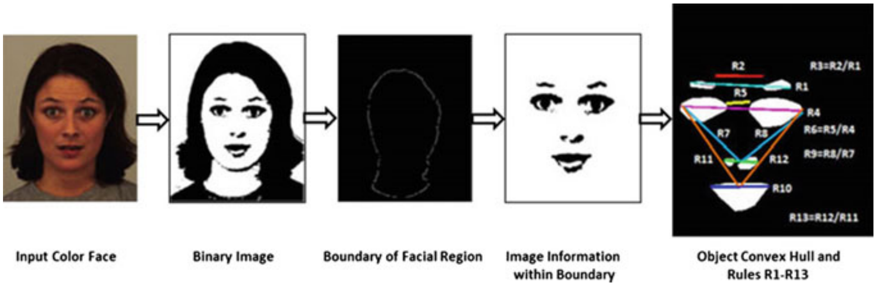

R2
R3=R2/R1
R1
R5
R4
R7
R8
R6=R5/R4
R9=R8/R7
R11
R12
R10
R13=R12/R11
Input Color Face
Binary Image
Boundary of Facial Region
Image Information within Boundary
Object Convex Hull and Rules R1-R13

Fig. 1 Feature extraction for knowledge-based classifier

build an effective system, we consider a set of three different classifiers each of which classifies facial expression irrespective of person. To obtain strong evidence, these three decisions are combined using a novel learning-based boosting process which is discussed in the next section. These classifiers are constructed as follows:

The first classifier of our system is build using knowledge-based approach [12]. This rule-based method encodes human knowledge of what constitutes an expression of typical face to identify facial expression. Usually, the rules capture the relationships between facial features by their relative distance and position. The feature extraction steps for knowledge-based classifier are shown in Fig. 1. Here, we consider some rule to identify facial expression of a facial image. Those rules are described as follows:

R1: distance between two outermost corners of eyebrows
R2: distance between two innermost corners of eyebrows
R3: ratio between R2 and R1
R4: distance between two outermost corners of eyes
R5: distance between two innermost corners of eyes
R6: ratio between R5 and R4
R7: distance between nostril points and left outermost eye corner
R8: distance between nostril points and right outermost eye corner
R9: ratio between R8 and R7
R10: distance between two mouth corners
R11 distance between center of mouth points and left outermost eye corner
R12: distance between center of mouth points and right outermost eye corner
R13: ratio between R12 and R11
All these rules constitute a single expression.

The second classifier of our system is a modified OCA (MOCA)-based classifier. We know, if intra-cluster distance is minimum and inter-cluster distance is maximum then the number of clusters formed will be optimum. So, to recover the performance of ordinary OCA, MOCA clustering considers these two distances. The separation within clusters S_w and separation between clusters S_b [13] are determined by following Eq. (1) and (2), respectively.

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{N_j} \left| \left(x_i^j - \mu_i \right) \right| \tag{1}$$

where C is the number of cluster formed,

N_j the number of patterns in a cluster,

x_i^j the jth component of the ith cluster,

μ_i mean of the ith cluster.

$$S_b = \sum_{i=1}^{C} \left| (\mu_i - \mu) \right| \tag{2}$$

where μ is the mean of all the means.

Now, we continuously perform OCA clustering until we achieve the optimal value of $\left|\frac{S_b}{S_w}\right|$. The number of cluster formed will be optimum in this ratio. The pseudo code of MOCA algorithm is given as follows.

Input: set of pattern point, S and threshold value, T (setting by well-tuned)
Output: Cj, $\forall j = 1,2,\ldots,M$ and M is total number of clusters to be formed

```
      do
      initialize cluster $C_j$ with any pattern point, say, $P_k$ //$\forall k = 1,2,\ldots,N$
      foreach $P_i$ in S, // $P_i$ is $i^{th}$ pattern point, $\forall i = 1,2,\ldots,N$ and $i \neq k$
            find closest $P_{i,}$from $P_k$
            if ( distance_of ($P_i$, $P_k$) < T)
                  include $P_i$ in $C_j$;
            else
                  include Pi in $C_j$+1;
            end if
   end for
   calculate$\left|\frac{S_b}{S_w}\right|$ from eq.1 and eq. 2
   untilmaximum_of ($\left|\frac{S_b}{S_w}\right|$) achieves
```

MOCA [3, 8] clustering is used in first stage of learning which returns the set of center and standard deviation. These are served as input to the intermediate level of RBFN. Finally, the output of RBFN is used as input to the back propagation network which is the decision of second classifier.

The third classifier of our system is HBC-based RBFN classifier. Here, HBC [9, 10] clustering technique is used to form "person-expression" group. In this technique, expected numbers of cluster will be formed by self-computing intra-cluster similarity which varies from one cluster to another. Within each cluster, the same procedure is applied recursively to get the subsets until the algorithm is converged. The pseudo code of HBC algorithm is given as follows.

```
Input: set of pattern point, S
Output: Cj, $\forall j = 1,2,\ldots,M$//M is number of clusters formed

        initialize cluster Cj with any pattern point, say, P1
        [T]= set_Threshold(); //set_Threshold() function returns threshold value for the current cluster
        foreach Pi in S, // Pi is ith pattern point, $\forall i = 1,2,\ldots,N$
                find closest Pi from (P1,P2,P3)
                if ( distance_of (Pi, (P1,P2,P3)) < T)
                        include Pi in Cj;
                else
                        include Pi in Cj+1;
                end if
        end for

        set_Threshold()
        {
             foreach Pi in S,
             find closest Pi from P1, let this new Pi is P2 and assign this distance to d1;
             find closest Pi from P2, let this new Pi is P3 and assign this distance to d2;
             find distance of P3 from P1 and assign this distance to d3;
             include P2 and P3 in Cj;//remove these P1, P2 and P3 from S;
             returnmaximum_of( d1, d2, d3);
     }
```

A suitable example of OCA clustering technique is shown in Fig. 2 and HBC clustering technique is demonstrated in Fig. 3. In this example, select ax randomly, and ay as nearest neighbor of ax and az as nearest neighbor of ay. Now compute each distance d1, d2 and d3 between ax, ay and az. Set threshold 'T' as maximum (d1, d2, d3). Randomly select a point 'b' and calculate distance di from ax, ay and az and check whether minimum distance of di is within threshold 'T', then 'b' is included within cluster C1 otherwise create another cluster C2 with the object 'b'. This procedure will apply until no more objects are left.

These set of mean and standard deviation are served as input to the intermediate layer unit of RBF network. Finally, output of this layer is used as input to the back propagation network which is the decision of third classifier.

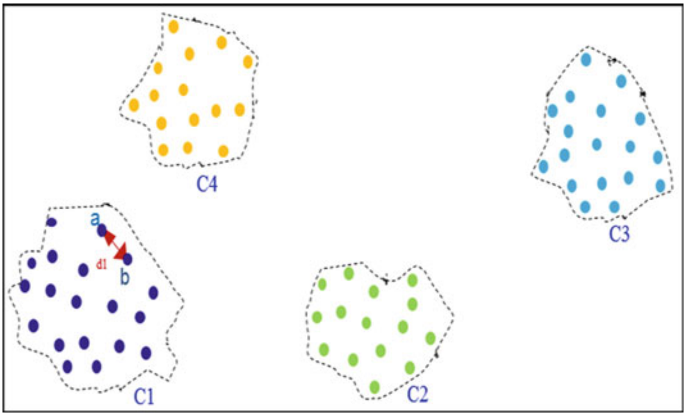


Fig. 2 An example of OCA clustering technique

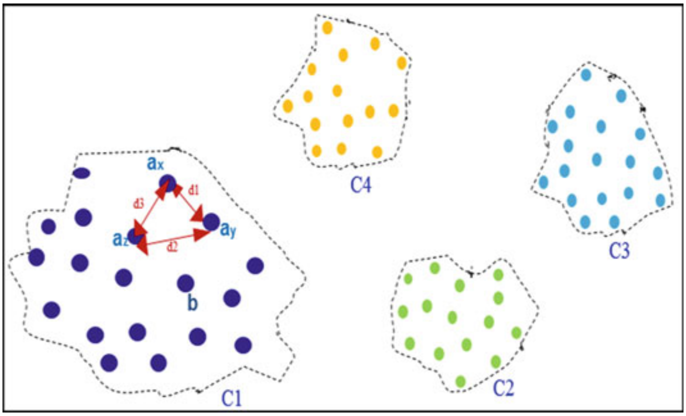


Fig. 3 An example of HBC clustering technique

3 Boosting Technique

Once the multiple classifiers are developed and trained, their individual decisions must be combined to form the final result. Recognition accuracy should be improved by integrating decisions of different classifiers. Numerous approaches exist to integrate classification decisions of different classifiers. Among them bagging [14] and boosting [15] are mostly used. These techniques are used to create a collective system using different training instances. In bagging method, final class is selected by taking decision which is agreed by most of the classifiers. To develop the performance of the bagging, performance of each classifiers is taken as weight in boosting [7] and vote of each classifier is multiplied with this weight to form weighted majority votes. We determine aggregate vote received by each class by Eq. (3):

$$V_j = \sum_{i=1}^{N} D_i.C_j \quad \forall j = 1, 2, \ldots, C \tag{3}$$

D_i is the weight learned by ith classifier,
C_j is the predicted as jth class by ith classifier

Two types of boosting techniques, namely programming-based boosting and learning-based boosting are used in general. In programming-based boosting technique, the weight of each classifier is selected by assigning its own performance (accuracy of that classifier). But in learning-based boosting technique, the weight of each classifier is assigned by an iterative learning procedure. In this technique, weight of each classifier is not assumed, it is learned. We approach a new learning-based boosting technique where we assume the initial weight of each classifier is its own accuracy as in programming-based boosting. Then we apply iterative learning procedure [7] to gain final weight of the classifier. Finally, we get the cumulative decision as a final decision.

4 Overview of the Present System

To identify a facial expression of a facial image, image pre-processing is used as the first stage. Figure 4 shows the simple architecture of an appearance-based classifier and Fig. 5 shows the graphical representation of proposed method. The proposed method is described below.

4.1 Pre-Processing

To improve performance of the system, the color facial images are pre-processed and compressed. Extraction of color facial features involves the following image pre-processing steps:

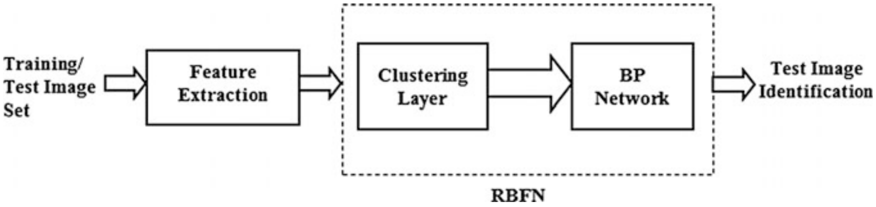


Fig. 4 Basic block diagram of an appearance-based classifier

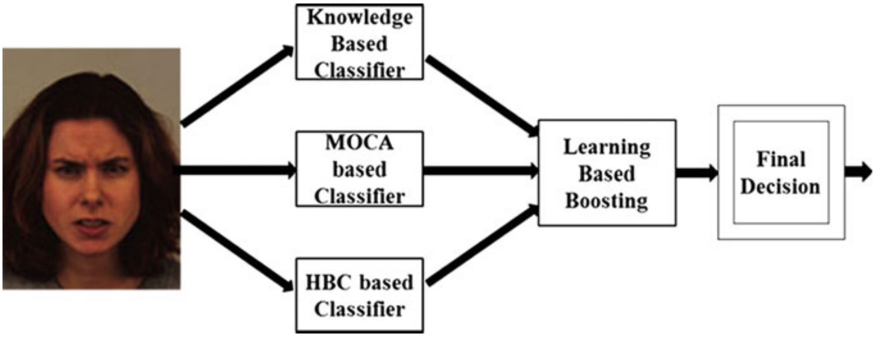


Fig. 5 Block diagram of the proposed system

- Generally, human eyes are more feasible to luminance, but not to chrominance. So, to achieve better representation of facial images, we convert this image from RGB color space to YCbCr color space [3].
- Now the skin area is searched by tuning a “skin_threshold_value” which determines whether the encountered pixel is skin part or not.
- In matlab, we used Robert filter *to* find the boundary of the skin area and *bwlabel()*for finding connected components in the processed image.

Then we extract color face region from the input facial image. The images are then compressed by taking the mode value of pixel intensities from the selected block of pixels. This normalizes all images of 640 × 480 into 128 × 96 dimensions. These compressed images are transformed into 1D column matrix which is fed as input to the system.

In case of knowledge-based classifier, we extract eyebrows, eyes, nose and mouth by applying object convex hull method. Then we form the rule R1–R13 from the extracted features (shown in Fig. 1).

4.2 *Classifications*

After taking out relevant features from the color facial images, these features are passed to an appropriate classifier amongst three distinct classifiers, two from these three classifiers perform normal grouping in the sense of human brainpower. For our system, pre-processed images are supplied as input to the knowledge-based classifier. Then this pre-processed image is input to the MOCA-based RBFN classifier and HBC-based RBFN classifier.

After, obtaining decisions of each classifier, we build the robust decision from those three weak responses using incremental learning-based boosting technique. Once the system finally learned weight of each classifier, we apply (3) to gain final predicted category that gets highest total votes. If the outputs of two or more classifiers are contradictory in nature, say, output 1 for one clustering algorithm is angry and output 2 for another clustering algorithm is happy, then the conclusion drawn by the classifier with higher probability has to be accepted during integration. So the ensemble algorithm during super classification perfectly handles this contradictory situation.

5 Experimental Results

This proposed system is simulated in the system with Intel 3.00 GHz CPU, 4 GB RAM and the database used in the experiment is KDEF [11] database.

To build the classifier, we chose faces of 5 distinct person, each with five different emotions (neutral, angry, happy, sad and surprise) with four variations of each emotion. Due to insufficient number of faces of a person present in the standard face dataset, we have made slight modification to form the training and test dataset. Number of test example sets used in our system are 320 which are not introduced during training. The performance evaluation time and accuracy of our system are shown in Table 1. Following this result, we can conclude that combining the evidence of multiple classifiers achieves greater performance instead of considering individual evidence. Due to power of boosting, the system gives nearly 99 % facial expression recognition accuracy where as it gives 96.4–98.4 % accuracy when each evidence is taken independently.

Initial weights of these three classifiers used in boosting are 0.33394, 0.33644 and 0.32960, respectively, which is normalized recognition accuracy of three classifiers. Then we apply iterative learning procedure [7] to these classifiers with training example set to gain final weight of each classifier. The final learned weight in our experiments is 0.3289, 0.3448 and 0.3263, respectively. Some salient portion of experimental result is shown below:

Table 1 Accuracy and performance evaluation time of different facial expression identification system

Identification using	Accuracy (%)	Learning time (in Sec.)		
		Training time (no. of training faces = 100)	Testing time (single test face)	Total time
Knowledge based	97.67	462.7	5.01	467.71
MOCA based	98.4	49.88	0.018	49.89
HBC based	96.4	31.55	0.016	31.56
Super classifier	99.06	544.13	5.044	549.16

```
Using Knowledge base, this is expression 5 with graded
probability: 0.6923
Using MOCA-based RBFN this is expression 2 with graded
probability: 0.9987
Using HBC-based RBFN this is expression 5 with graded prob-
ability: 0.3522
After applying boosting this is expression 2 with graded
probability: 0.3522
```

Now, we can see that if we apply simple majority voting then the result as expression 5 (surprise) will be accepted because two from three classifiers agree with this decision which is wrong. But when we apply new Boosting technique, the result is expression 5 with weight of 0.34261 and expression 2 (angry) with weight of 0.34438. So, we accept the maximum weighted result which is expression 2.

6 Conclusion

This paper presents a multiple-classification system, which identifies five basic emotions of a person. Each classifier is used to train same color facial feature and finally the decisions of individual classifier are combined together through the technique of new learning-based boosting to categorize the person's emotional states. Facial expression identification through this new system of super classification is efficient. In addition, performance evaluation of the super classifier in terms of recognition accuracy is reasonably high. Consequently, the current system of learning-based boosting in super classification is reliable to a certain extent for facial expression recognition.

References

1. Mehrabian, A.: Communication without words. Psychol. Today **2**(4), 53–56 (1968)
2. Ekman, P., Friesen, W.V.: Facial Acton Coding System. Consulting Psychologists Press, InvestigatorsGuide (1978)
3. Bhakta, D., Sarker, G.: A method of learning based boosting in multiple classifier for color facial expression identification. In: 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), pp. 319–324 (2015)
4. Ma, L., Khorasani, K.: Facial expression recognition using constructive feedforward neural networks. IEEE Trans. Syst. Man Cybern. Part B Cybern. **34**(3), 1588–1595 (2004)
5. Ryan, A., Cohn, J.F., Lucey, S., Saragih, J.: Automated facial expression recognition system. In: 43rd Annual International Carnahan Conference on Security Technology, pp. 172–177, October 2009
6. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognit. **36**(1), 259–275 (1999)
7. Bhakta, D., Sarker, G.: A multiple classifier system with learning based boosting for clear and occluded color face identification. Am. J. Adv. Comput. **1**(2), 54–59 (2014)

8. Bhakta, D., Sarker, G.: An unsupervised OCA based RBFN for clear and occluded face identification. In: First International Conference on Intelligent Computing & Applications 2014 (ICICA 2014), Advances in Intelligent Systems and Computing, 2014, pp. 19–25. Springer (2014)
9. Sarker, G.: A Heuristic Based Hybrid Clustering for Natural Classification. IJCITAE Int. J. Comput. Inf. Technol. Eng. **1**(2), 79–86 (2007)
10. Sarker, G., Sharma, S.: A heuristic based RBF network for location and rotation invariant clear and occluded face identification. In: International Conference on Advances in Computer Engineering and Application, ICACEA-2014, with IJCA, pp. 30–36
11. Lundqvist, D., Flykt, A., Öhman, A.: The Karolinska Directed Emotional Faces—KDEF. In: CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9, 1998
12. Hamouz, M., Kittler, J., Kamarainen, J.K., Paalanen, P., Kal, H., Matas, J.: Feature-based Affine-Invariant localization of faces. IEEE Trans. Pattern Anal. Mach. Intell. **27**(9), 1490–1495 (2005)
13. Martinez, A.M., Kak, A.C.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 228–233 (2001)
14. Breiman, L.: Combining Predictors, Combining Artificial Neural Nets, pp. 31–50 (1999)
15. Freud, Y., Schapire, R.: Experiments with a new boosting algorithms. In: Proceedings of the 13th International Conference on Machine Learning, pp. 148–156 (1996)

Efficient Face Detection Using Neural Networks

Amit Sinha

Abstract There is a growing need for automated face recognition systems for various purpose, mainly security within organizations. The problem of face recognition consists of mainly 2 important steps. The first step is obtaining the region of the face from a raw image (face detection) and this is followed by a face recognition step to identify the individual. This paper gives an overview of the method used to detect the location of frontal faces in a small amount of time. The method involves creating a data set from a face database which has the localized parts of face images along with corresponding scores of the sectors of the face that are being detected. Based on the scores obtained further processing is done in the likely face area to determine whether a frontal face is present in the given region. This detection can be done in a small amount of time because the neural network can identify parts of a face from a kernel and not too many kernel operations are required to successfully identify the face.

Keywords Neural networks • Frontal face detection • Gradient descent • Scale invariance • Real time detection • Local feature detection

1 Introduction

There are presently many techniques for face recognition that work well to identify face regions in raw input images. Many of these methods however involve the concept that there is a varying kernel moving about all possible locations of the image. Also because of scale invariance it is required to try out multiple kernel sizes as well which results in many iterations which can be computationally expensive for a large image. The Viola Jones method for face detection is one such example. It is effective in places where the sizes of face that are to be detected are more or less fixed in size so that the detection kernel can detect them in real time.

A. Sinha (✉)
EEE Department, BIT Mesra, Ranchi 835215, Jharkhand, India
e-mail: amitfishy@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_27

However, for such an algorithm the scale variance problem is handled by using different kernel sizes and this would result in a lot of computations which in some cases may not be achievable in real time.

The method discussed in this paper uses fewer kernels since the neural network is capable of identifying localized features of the face and it is also capable of identifying which part of the face is identified. Using this information, it can be checked if there is face region by checking for the other features of the face in the local area of the part which is first identified. Thus by selecting a few threshold parameters we can adjust the boundary between false face predictions (false positives) and actual face predictions (true positives).

The remainder of this paper is organized as follows. Section 2 consists of the details on how the database of local features of faces is created with the locality of features expressed as a score. Section 3 consists of the details involved in architecture and training of a neural network to identify the local features of a face and their locality in a face. Section 4 discusses an algorithm used to implement the neural network to get location of faces in a raw image. Section 5 discusses the results and Sect. 6, conclusions.

2 Database Creation

The image database under consideration is a very small subset which is taken from [1] consists of 96 × 96 face images (mostly frontal). From these face images we obtain a large number of images considering the various local features of the face.

This is done by considering the various parts of a general face image to be divided into 9 sectors as shown in Fig. 1 and following a simple rule which states that if part of the face is detected in the kernel, then the score of the local features for that particular kernel is determined by the area in the kernel that covers each sector of the face image. The more area that covers a certain local feature, the larger will be the score for that sector in that kernel.

The sector score for the kernel vary between 0 and 1 where the value 0 for a sector means that no part of the sector of the face image is present in the kernel and the value 1 means that the entire area of the sector of the face image is present in the kernel.

For example, if in the above 96 × 96 image we have in the top left corner a kernel of size 48 × 48. The sectors are 32 × 32. Then the scores evaluated for that particular kernel would be 1 for Sector 1, 0.5 for Sectors 2 and 4, and 0.25 for Sector 5.

The database is created by taking various sub-images from the images of faces of varying kernel sizes and calculating corresponding scores using the aforementioned method and storing these results together. The sub-images of varying sizes are down-sampled to 24 × 24 and stored. The sub-images considered have been of sizes 32 × 32 to 80 × 80 (345 sub-images) in order to give the system the ability to deal with scale invariance up to some extent.

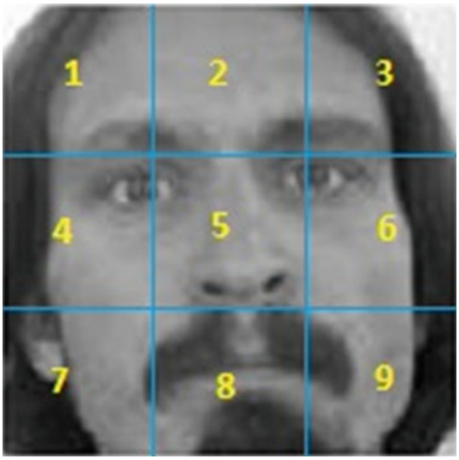


Fig. 1 Sector division

The down-sampled image data is normalized by subtracting the mean of all the sub-image pixels and then dividing by the standard deviation of all the sub-image pixels.

All of the above has been implemented in Matlab and the resulting database is the one that we require for training in the neural network.

3 Neural Networks

Neural networks have been used in the past for many papers involving pattern recognition. There have been many neural networks used for facial recognition as well and the models are seen to be quite successful in terms of recognition rate.

The most basic neural network consists of an input layer, a hidden layer and an output layer. Through forward propagation the output layer is obtained from the input layer.

In this method we have used a 3 layer network, with 576 input neurons (corresponding to the 24 × 24 grid of down-sampled sub-images) and 100 hidden neurons in the hidden layer and 9 neurons in the output layer. The value of number of faces used for training is 300. The sigmoid function is used as activation function for all the neurons. The target output is calculated in the database creation step which is used to find the errors in the network and tune the parameters of the network.

The training is done using adaptive batch gradient descent with momentum. The gradients are calculated using back-propagation algorithm which involves back-propagating errors to determine gradients with respect to each weight and bias.

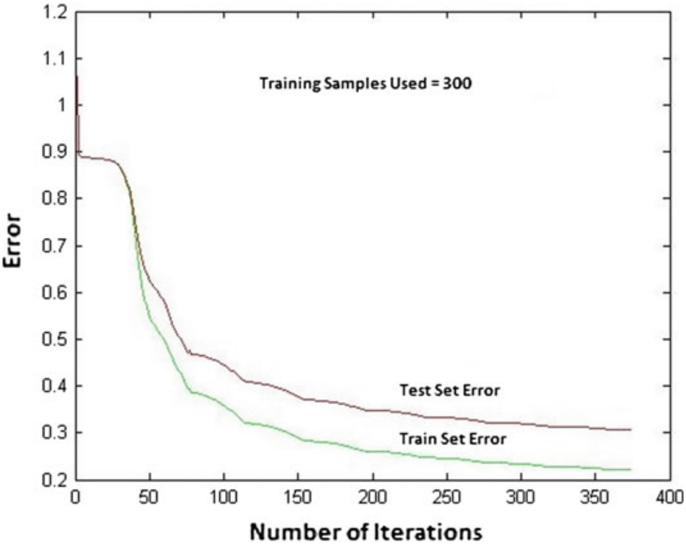


Fig. 2 Neural network error graph

The training error and test error for the finally selected network is plotted as a function of number of iterations of gradient descent in Fig. 2. After more iterations are allowed to take place the training error decreases to 0.1199 and the test error becomes 0.1281. It is seen that there is some amount of over-fitting taking place but the generalization error is fair.

4 Face Detection Algorithm

Now that the trained neural network is obtained it is ready to detect local features in a raw input image.

The input images are considered from a database called BioID Face Database [2] having frontal oriented faces amongst different backgrounds. The images are of size 384 × 286. For this particular size, search kernels of 40–130 are used in increments of 10.

The algorithm proposed is as follows. The search kernel is moved around the image in steps of half-kernels in both dimensions of the image. The image under the kernel is down-sampled to 24 × 24 and then reshaped to a 1 × 576 column-oriented vector (which is then normalized with mean and standard deviation as before) to feed into the input layer.

Firstly the possible centers of face images are identified by checking the value of the central sector obtained. A certain threshold value is given below which further checking is not done and the next kernel is checked. If the central sector has a value given by the neural network that exceeds the threshold, then further checking is done since it may or may not contain the face.

Further checking is done at 9 kernels around the existing kernel at a distance of half-kernels from the original. The scores obtained from each of the 9 kernels used is cumulatively added after multiplying it with a weight matrix (3 × 3) whose values depend on the kernel being used. The values to the weight matrices are given as positive values to the areas which are expected to give high scores for that part of the face, and negative values to the areas which are not expected to give any activation. This is done in order to decrease false results given by random activations. In this fashion different weights are assigned to each kernel depending on the expected position with respect to the center.

The cumulative score obtained from all 9 kernels is compared with another threshold whose value is obtained by experimentation.

After the first step at the original kernel size is done, a new kernel which is twice the original kernel size and centered at the original kernel is considered. The second check is done in the manner explained above in the new larger kernel that is obtained. These 9 kernels are of size 1.2 * Original Kernel Size. According to this size the values to the weight matrices are given. Then the score obtained is checked if it is greater than the threshold.

If this stage is passed a similar checking procedure is done, but this time with 1.5 * Original Kernel Size (for which the weight values will be slightly different) and the resultant score is compared with a different threshold whose optimum value is also obtained by experimentation.

We can even use additional checks after this with larger kernel sizes to improve the accuracy and avoid false detections.

If the kernel can satisfy all these criteria then the region enclosed gets detected as a face.

5 Experimental Results

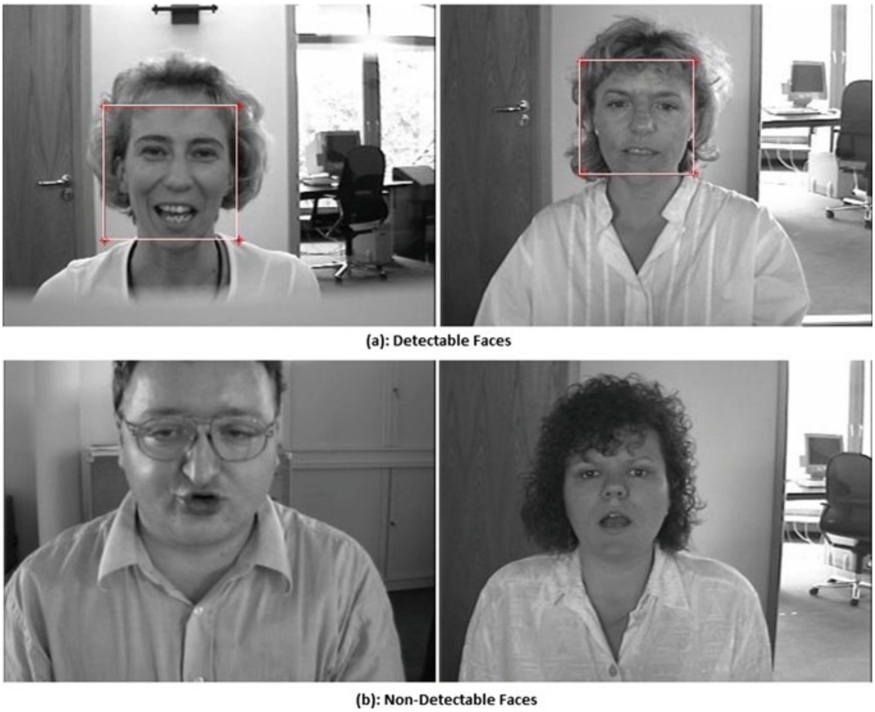
The threshold constants are found out by experimenting with a separate set of a few face images based on trial and error and intuition to try and maximize the number of faces that are detected with minimum number of false detections.

After this 53 samples are considered at random from the BioID database without changing the threshold constants and the results were as follows

- 42 faces were detected properly
- 11 faces were not detectable
- There were additionally 4 false detections

Each search takes about 7 s on a 1.8 Ghz processor.

Few of the results are shown in Fig. 3.



(a): Detectable Faces

(b): Non-Detectable Faces

Fig. 3 Results of face detection

6 Conclusions

The method used in this paper is a fresh idea in which lot of different approaches can be applied to improve the accuracy and speed of the overall system.

The neural network that is trained still has a considerable amount of generalization error. Despite having 300 faces for training and a hidden layer of size 100, its performance can be greatly improved by considering more network complexity and more faces for training. Additional layers can be added which may give a better representation for smaller amount of parameters. Also these methods will not only improve the accuracy of face detection, but decrease the rate of false detections and the speed of the overall system. The speed can be improved by increasing the value of the thresholds which results in a large number of reduced checks for faces. If the network is improved then it will be able to identify faces despite higher threshold values. In this implementation the above was not done due to machine limitations.

Apart from this modifications may be made to the proposed algorithm to improve the performance, such as using larger sectors to capture the variation of local face features in a better manner. This means that the 3 × 3 sector approach could be compared with 2 × 2 sectors.

Additional problems of the network include false activations for non-face features which are usually not much of a problem since the region of activation for non-face images seems somewhat random and this randomness is often detectable by the algorithm used. Additionally false positives can be reduced by training with some non-face feature images which may also improve the accuracy of the system.

Apart from this certain techniques can also be used to improve the performance of the network such as representing the features of the network in a different way to reduce dimensionality (like Principal Component Analysis or Linear Discriminant Analysis) as is done in [3] or using transforms such as wavelet transforms as is done in [4].

Also since the exact outline of face is not always selected additional techniques may be used such as contour analysis and shape detection [5, 6] to get the exact part of the face which then has to be sent to a facial recognition algorithm.

The proposed algorithm is capable of a significant amount of development for improved accuracy and speed. These techniques may be applicable to object detection applications in general which have a fair amount of complexity in the local areas.

Acknowledgments The author would like to thank professor G. Sahoo, CSE Department, BIT Mesra, for his support and guidance.

References

1. https://www.kaggle.com/c/facial-keypoints-detection/data which credits Dr. Yoshua Bengio of the University of Montreal
2. https://www.bioid.com/About/BioID-Face-Database
3. Nazeer, S.A., Omar, N., Khalid, M.: Face recognition system using artificial neural networks approach
4. Paliy, I., Sachenko, A., Koval, V., Kurylyak, Y.: Approach to face recognition using neural networks
5. Bai, X., Li, L., Liu, W., Latecki, L.J., Tu, Z.: Shape band: a deformable object detection approach
6. Yong, L., Changshui, Z., Xiaoguang, L., Zhang, D.: Face contour extraction with active shape models embedded knowledge

Part VI
Applications in Security

Scalable Framework for Developing Adaptive and Personalized User Interfaces

Aman Jhunjhunwala and Sudip Kumar Sahana

Abstract This paper presents the need for adaptive user interface, the challenges for client-side user interface applications and proposes a scalable framework for conquering those challenges to provide a pipeline-based fast, effective and extensible tool that adapts all the applications to the user's needs in a way that enhances his productivity the most. Different abstractions are explored in depth and their graphical results and possible implementations have been discussed.

Keywords Human–computer interaction · Adaptive websites · Intelligent websites

1 Introduction: The Need for Adaptive User Interface

1.1 Growing Complexity of Modern Web Applications

Before the advent of powerful front-end frameworks such as Bootstrap, AngularJS, ReactJSetc. websites were mostly static wireframes hosted on a server. They did very little or no computation and had minimum information to display to the end user in a simplistic manner. Today, a large developer base uses these frameworks to build creative but often complex and cryptic looking websites.

A. Jhunjhunwala (✉) · S.K. Sahana
Department of CSE, Birla Institute of Technology, Mesra 835215, Jharkhand, India
e-mail: amanjjw@gmail.com

S.K. Sahana
e-mail: sudipsahana@bitmesra.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_28

1.2 Diverse Characteristics of Users

Apart from understanding how to navigate the website, the biggest problem is the large amount of time it takes to find the content that one is looking for on the website. Different users have different goals and the web application should adapt itself to ease interaction and be relevant and productive to each user that accesses it by taking into account the differences in knowledge, style, language and preferences.

Most visitors desire productivity. They do not wish to scroll through hundreds of links to find the article that interests them. They wish to interpret data not as tables but as more intuitive bar-graphs, pie charts and other forms of data visualization that enhance productivity.

Some visitors desire ease of communication with the website. A large population of developing and underdeveloped countries are accessing the web for the first time. A large number of them are not trained or taught on how the web works. The website should still be understandable to them. It should inform them the consequences of clicking links—what each link does. These self-teaching website designs are going to be pivotal between 2015 and 2022 where the number of web users is expected to double.

1.3 Rapidly Expanding Web

The number of web applications is drastically increasing with each passing day. Each application uses a large and variable set of technologies to run. Some are actively maintained; others are deprecated but host useful information. It is not possible for each one of them to implement new frameworks or modify existing stacks to accommodate personal interests. Hence, a client-side implementation is required to realize the needs for personalized user interfaces.

2 Challenges for Client-Side Adaptive UI Implementation

2.1 Scrambling of Contents of Webpages

The biggest restriction in the field of adaptive user interfaces is that webpages can easily be scrambled [1]. As web developers get more innovative with the design, it becomes increasingly difficult to reorganize and adapt contents of the webpage from the client-side implementation. One slight mistake can scramble the entire content of the website. When adaptive interfaces are added from the server side, the developers of the website take excessive care that the re-organization of contents does not break the contents of their application. This, however, is a great challenge for general client-side UI implementations [2–4].

2.2 No Access to Server Logs

Without having direct access to server logs of all websites across the internet, it becomes difficult for client-side implementations to identify the user's activity across the application. It requires an intermediate server to track the activities of the user. These data are essential as it helps in implementation of better operations for adapting the webpage to suit to a particular user.

2.3 Significant Delays in Loading the Webpage

A client-side implementation requires the presence of an intermediary link between the remote server and user. This may lead to significant delay in the loading of the pages, especially when a lot of customization is required for a page.

3 Methodology

The proposed algorithm can be summarized as: (Figs. 1 and 2)

1. Request the page to browse through our proxy server
2. The proxy server forwards the request, fetchingthe contents from the remote host and stores the HTML in our database, if not cached earlier.
3. Caching works in 2 steps
    a. Long Term Caching : Store data of webpage in our cache if the number of hits are above a certain limit.
    b. Short Term Caching :Prefetch contents of links the user is most likely to visit and apply the pipeline processes to it.
4. If the user is new, determine a new pipeline based on k-nearest neighbours
5. If the user is returning, recommend him useful pipelines using Basket Analysis
6. Apply all the processes in a user's pipeline to the HTML and display the result, keeping complete track of all requests of each user, for increasing knowledge base.

Fig. 1 Proposed algorithm for adaptive user interface

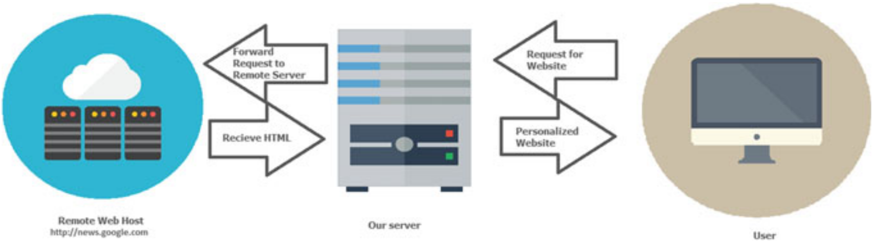


Fig. 2 Diagrammatic representation of top level abstraction

3.1 Top Level Abstraction: Flow of Data

At the top level, the user requests a website through our server (Example: www.example.com/www.news.google.com). The request is received and forwarded to get the contents of the requisite URL in our own server. The contents of the website (only the HTML enclosed within the < body > tags) are then processed through various Machine Learning algorithms described later and provide the processed contents of the website to the user—adding each request to our knowledge base.

3.2 Mid-Level Abstraction—The Concept of Pipeline

Inside our server, the received contents from the remote server pass through a series of pipeline as shown in Fig. (3).

Each stage of the pipeline manipulates the contents received from the previous stage to accomplish one task. There are four major issues associated with pipeline structure: -

Recommending Processes to an existing User Pipeline
For suggesting new processes to add to an existing pipeline, a modified version of **Basket Analysis** is proposed. This mode of learning is applicable to any setting where a group of objects are present and a new object needs to be recommended. The knowledge base consists of the set of pipelined processes often used together by different users. It is analyzed to recommend new processes to users having similar processes in pipeline. This algorithm gets fooled by very frequently used processes and would simply recommend those which are popular without any personalization. To overcome the drawback, building of **association rules** is required.

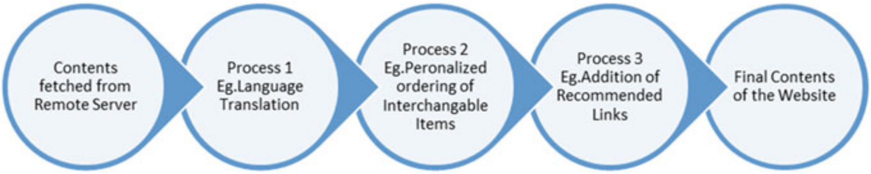

Contents fetched from Remote Server
Process 1 Eg.Language Translation
Process 2 Eg.Peronalized ordering of Interchangable Items
Process 3 Eg.Addition of Recommended Links
Final Contents of the Website

Fig. 3 Flow representation of a sample pipeline

An association rule is a statement of the "If X Then Y" form; if someone has used Process A then he will use process B. Note, however, that the rule is not deterministic but probabilistic in nature. The antecedents and conclusion may contain multiple objects. Multiple antecedents allow more specific predictions than are possible from single items.

It is easy to generate multiple rules by finding subsets of current baskets of processes, but only those rules that are valuable should be considered. The value of a rule here is measured by the ratio between the probability obtained by applying the rule and the baseline, commonly known as **lift**.

$$\mathrm{lift}(X \rightarrow Y) = \frac{P(Y|X)}{P(Y)}.$$

In the preceding formula, P(Y) is a fraction of all user pipelines that include Y while P(Y|X) is the fraction of pipelines that include X and Y both. Using lift helps avoid the problem of recommending popular but unrelated pipeline process.

Determination of the Initial Pipeline for New Users

The initial pipeline for a new user is obtained by applying a slight modification of K-Nearest Neighbors algorithm taking the age and location of users as features. The features of existing users are plotted in two-dimensional space and whenever a new user joins the closest k points (in terms of distance) are considered. All k neighbors vote to select the pipeline. The new user's pipeline would be composed of the processes that were common to a minimum c number of neighbors, where c is a pre-settable threshold. The value of k will increase with the increase in our knowledge base. The values of c and k can be decided by cross validation.

Caching—Increasing Performance

If the pipeline is big, the time taken to process webpages would be impermissible for users to use the framework. An intelligent caching system is necessary on our server. Two types of caching can be implemented for faster retrieval.

Long-term caching: A straightforward mechanism is implemented. Pages most often visited by users (number exceeding a certain threshold) of our application are stored and indexed in the database (in raw HTML form). Whenever a user requests a webpage, it is first checked if the webpage has been indexed already in the database. If present, it is sent to the user without fetching any content from the remote server that actually hosts the webpage. The raw contents of the database are refreshed every hour. Every time the database is over-capacity, server logs can be analyzed to find the Least Recently Used page and the contents are removed.

Short-term caching: This extends the long-term caching mechanism. Whenever the user fetches a webpage, the framework analyzes the links present on the webpage and starts prefetching the contents of the links that belong to the same domain, to prevent unnecessary fetching of advertised or sponsored content. If the user clicks a link already cached, the recently cached copy is handed over, without forwarding any request to the remote server. Whenever the user leaves the link, the data associated are removed. This caching can be improved further by intelligently pre-fetching only those URLs that the user might click by analyzing the text within the links themselves [5–7].

All the caching can take place in parallel through distributed queues or running the internal operations on separate workers or servers.

Compatibility of Pipeline Processes

Each pipeline can be composed of any number of pipes or processes. The compatibility of the pipes is defined by two parameters—Structure and Language. Each pipe may be viewed as a black box accepting a particular input following a particular structure and language and outputting data in the same or different structure and/or language.

Structure: This describes the structure of the data. Data might be available as Plain Text, XML, JSON, YAML, HTML, etc. Each of them follows a different parsing structure.

Language: This describes the semantics of the data. It could include the literal languages for Plain Text structure like German, English, etc.

The need for defining the parameters lies in the fact that some pipes may involve language processing tasks such as providing summaries of large articles which are heavily dependent on the language of the text [4, 8, 9]. An English summarizer does not work well with German text. Another important use is to allow processing using intermediate datatypes. There could be pipes outputting JSON data and pipes accepting JSON data for faster intermediate processing. The relative ordering of pipes accepting and outputting the same structure and language does not matter much, provided non-similar intermediate pipes are absent. Sequence of pipes having the same parameters allows the pipes to be evaluated in parallel, thereby further decreasing the processing time.

3.3 Lowest Level of Abstraction: The Pipes

To design new and innovative pipes or processes, the developer has full access to all the data stored in the running server. This includes all the sequences of links clicked by every user, the time of the click and the city-level location (giving out the exact location could raise privacy concerns. The location would be given out at a city-level rather than at a street level) and age of the user. Allowing the developers, this common set of data accessible through API GET calls will allow the development of native, powerful and useful processes. A few suggestive concepts are provided for the same. These include

- Skipping intermediate links in websites using PageGather algorithm [7]. This allows for faster browsing, reduction in complexity of the website and ease of finding relevant contents present in deeper sections of the websites.
- Expanding Navigational Links. The text of the link as well as the link itself can be examined. Using explanatory language processing algorithms, help texts for the links can be automatically generated.

- Translating webpages to the regional language—This allows naïve web users to browse the web comfortably. Services which currently offer their APIs for this functionality include Bing Translate, Google Translate, etc.
- Convert tables present in the HTML to a selection of graphs using JavaScript libraries like d3.js. Tables are analyzed if they possess data to be presented in graphical format. If so, they are displayed on hovering over the tables
- Increasing font size for older users
- Recommending content from external or internal websites based on browsing experience.
- Smart display of webpages using Cortana and other pluggable virtual assistants. Browsing a webpage, dynamic information based on the content of the webpages can be generated.

4 Implementation Details

A basic prototype of the proposed framework was implemented using Django—a Python Web Framework. Given the URL of a simple HTML webpage, the server would scrap the contents of the webpage through Python's BeautifulSoup4 Library, add the contents to our Database, if not already present. Then, the scraped contents pass through different pipes, implemented as Python Modules, each adding or manipulating the contents to add personalized features to the website. The 3 pipes implemented in the prototype were Language Translation, Conversion of tables to Graphic Diagrams and Adding to Social Sharing Links.

The Language translation was implemented using Bing Translator API within our Django Application to detect the country of the user and translate only the text of the HTML to the native language, using API calls.

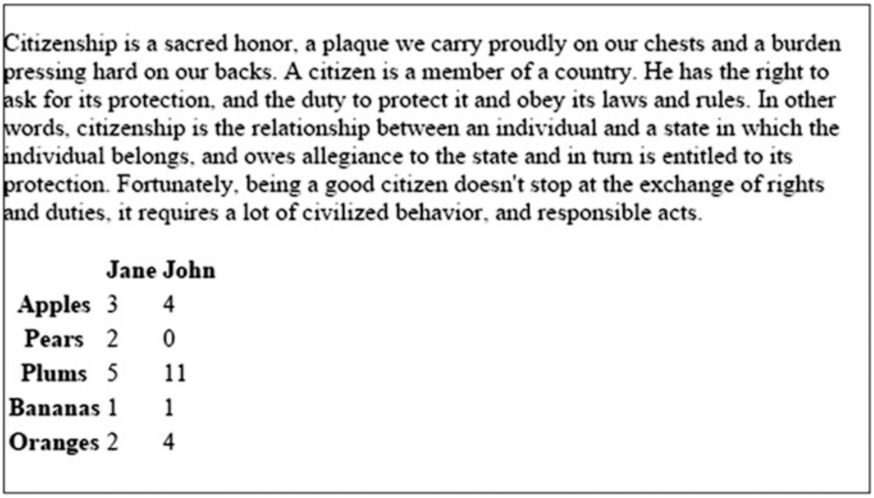
The conversion of tables with relevant data to charts was implemented using d3.js a vector Javascript library. The data from the tables were extracted and the relevant axes and scaling were automatically framed and graphs obtained.

The Addition of Sharing Links is obtained by adding an AddThis® Javascript widget to the web application to allow for sharing the page on all the social media platforms.

5 Results and Discussion

The results of the prototype have been presented in Figs. 4 and 5. The contents of a simple HTML file have been transformed to be interactive and more functional.

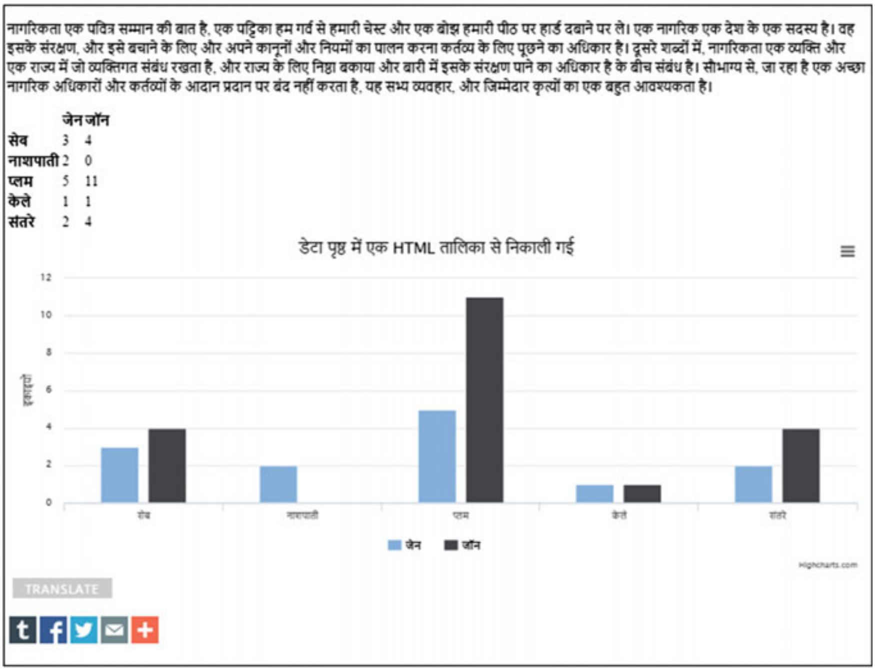
The language has been translated (both the graphs and the text) and the data from the table were automatically extracted and plotted. Social Sharing links are available as well.



Citizenship is a sacred honor, a plaque we carry proudly on our chests and a burden pressing hard on our backs. A citizen is a member of a country. He has the right to ask for its protection, and the duty to protect it and obey its laws and rules. In other words, citizenship is the relationship between an individual and a state in which the individual belongs, and owes allegiance to the state and in turn is entitled to its protection. Fortunately, being a good citizen doesn't stop at the exchange of rights and duties, it requires a lot of civilized behavior, and responsible acts.

	Jane	John
Apples	3	4
Pears	2	0
Plums	5	11
Bananas	1	1
Oranges	2	4

Fig. 4 Before implementing the framework, the contents of a sample webpage



नागरिकता एक पवित्र सम्मान की बात है, एक पट्टिका हम गर्व से हमारी चेस्ट और एक बोझ हमारी पीठ पर हार्ड दबाने पर ले। एक नागरिक एक देश के एक सदस्य है। वह इसके संरक्षण, और इसे बचाने के लिए और अपने कानूनों और नियमों का पालन करना कर्तव्य के लिए पूछने का अधिकार है। दूसरे शब्दों में, नागरिकता एक व्यक्ति और एक राज्य में जो व्यक्तिगत संबंध रखता है, और राज्य के लिए निष्ठा बकाया और बारी में इसके संरक्षण पाने का अधिकार है के बीच संबंध है। सौभाग्य से, जा रहा है एक अच्छा नागरिक अधिकारों और कर्तव्यों के आदान प्रदान पर बंद नहीं करता है, यह सभ्य व्यवहार, और जिम्मेदार कृत्यों का एक बहुत आवश्यकता है।

	जेन	जॉन
सेब	3	4
नाशपाती	2	0
प्लम	5	11
केले	1	1
संतरे	2	4

Fig. 5 After passing through the prototype framework, the contents of the webpage were automatically changed as shown

This accessibility comes at the cost of time. For testing, an untampered HTML page of size 163 kilobytes took 0.4s and 6 requests to load. The same page after passing through our framework took 1.0s to load with an increased size of 306 kilobytes and 10 client requests through the browser. Both the time required to load the page and the size of the page have scaled up, and there is a tradeoff between the accessibility the users desire and what time do they wish to sacrifice for the purpose.

6 Comparison with Standard Algorithms

Currently, most of the client-side adaptive web design algorithms focus on the responsive aspect of interface designing. For responsive web design, "browser sniffing" or "user-agent" detection is used as ways of deducing if certain HTML or CSS features are supported. They rely on graceful degradation to make a complex image heavy site work on mobile phones to tablets. Server-side "dynamic CSS" implementation of stylesheet languages like Sass or Incentivated's MML can be part of such an approach by accessing a server-based API which handles the device differences in conjunction with a device capabilities database to improve usability. Although these frameworks are general in nature, they do not take into account user's preferences or past history to display the user interface. The only metric for adaptivity is screen size and device type. Our proposed framework takes more resources (caches, proxy servers, etc.) than this framework to operate, but offers virtually unlimited customizing power and scalability for almost the same operating time.

However, most of the adaptive applications today rely on server-side adaptive interfaces, where user interfaces of their own site are customized based on server logs and past history of users on their particular site. Although this method overcomes the huge problem of content scrambling, as the layouts are handled by the site designers themselves, they are not very general. Less than 1 % of all the websites and applications use this technique. It adds additional complexity which designers find unnecessary in most situations. With our implementation, the designers do not need to worry about manually adding in these technologies; the framework will handle the structural changes on their part. Each and every web application on the web can be adapted to suit each person's individual needs.

7 Conclusion and Future Work

The novelty of proposed Adaptive Personalized User Interface lies in its simplicity and ease of use. Although a variety of research and development efforts have shown the potential of adaptive user interfaces, there remains much room for extending their capabilities and efficiency. Future work would include deriving a better understanding of HTML semantics so as to allow the framework to work on complicated websites seamlessly. Collection of user data will generate new metrics and will give us an in-depth characterization of each user. Better and more effective algorithms for caching would be required for minimum waiting time to users. In this context Computational Intelligence will always play a key role to improve the efficiency and effectiveness of the proposed framework.

References

1. Langley, P.: Machine learning for adaptive user interfaces. KI-97: Advances in Artificial Intelligence. LNCS, vol. 4128, pp. 53–62. Springer, Berlin Heidelberg (1997)
2. Rogers, S., Langley, P., Johnson, B., Liu, A.: Personalization of the automotive information environment. In: Proceedings of the Workshop on Machine Learning in the Real World: Methodological Aspects and Implications, pp. 28–33. Nashville, TN (1997)
3. Rudstrom, A.: Applications of machine learning. Licentiate thesis, Department of Computer and Systems Sciences, Stockholm University, Sweden. (1995)
4. Schlimmer, J.C., Hermens, L.A.: Software agents: completing patterns and constructing user interfaces. J. Artif. Intell. Res. **1**, 61–89 (1993)
5. Dent, L., Boticario, J., McDermott, J., Mitchell, T., Zaborowski, D.: A personal learning apprentice. In: Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 96–103. AAAI Press, San Jose, CA (1992)
6. Giordana, A., Saitta, L., Bergadano, F., Brancadori, F., De Marchi, D.: Enigma: a system that learns diagnostic knowledge. IEEE Trans. Knowl. Data Eng. KDE-5, 15–28 (1993)
7. Perkowitz, M., Etzioni, O.: Towards adaptive Web sites: conceptual framework and case study. In: Enslow, Jr. P.H. (ed.) Proceedings of the Eighth International Conference on World Wide Web (WWW'99), pp. 1245–1258. Elsevier North-Holland, Inc., New York, NY, USA (1999)
8. Michie, D.: Problems of computer-aided concept formation. In: Quinlan, J.R. (ed.) Applications of expert systems, vol. 2. Addison-Wesley, Wokingham, UK (1989)
9. Hinkle, D., Toomey, C.N.: Clavier: applying case-based reasoning to composite part fabrication. In: Proceedings of the Sixth Innovative Applications of Artificial Intelligence Conference, pp. 55–62. AAAI Press, Seattle, WA (1994)

Fuzzy-Based Privacy Preserving Approach in Centralized Database Environment

V.K. Saxena and Shashank Pushkar

Abstract In data mining applications, sharing of huge volume of detailed personal data is proved to be beneficial. Different types of data include criminal records, medical history, shopping habits, credit records, etc. These types of data are very much significant for any company and governments for decision making. When analyzing the data, privacy policy may avoid data owners from sharing information. In privacy preserving, data owner must provide a solution for achieving the dual goal of privacy preservation as well as accurate clustering result. Nowadays, privacy issues are challenging concern for the data miners. Privacy preservation is a multitalented task as it ensures the privacy of individuals without trailing the correctness of data mining results. This paper proposes data transformation methods for privacy preserving clustering based on fuzzy in centralized database environment. After experimenting, results proved that hybrid approach gives best results for all the member functions.

Keywords Clustering • Data transformation • Fuzzy membership function • Privacy preservation • Random rotation perturbation

1 Introduction

Nowadays, the data collected from any organizations has been an extensive growth. These data are collected from various sources such as medicinal, financial, documentation, cellular phone, and shopping records. Such data can be incorporated and analyzed digitally as it is possible due to the rapid growth in database, networking, and computing technologies. Data storage is growing at an exceptional speed. Data

V.K. Saxena (✉)
School of Engineering & Technology, Vikram University, Ujjain, M.P, India
e-mail: vksaxena2002@yahoo.co.in

S. Pushkar
Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India
e-mail: shashank_pushkar@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_29

users are concerned to mine useful information from these huge amounts of data. Data mining is gradually a more important tool that can transform data into constructive knowledge. Data mining tasks provide precise information for decision making. A number of data mining tasks are classification, association rules, prediction, and clustering. Clustering is a well-recognized problem in engineering and statistics and widely used in a variety of applications including biology, marketing, medicine, etc. Clustering is the process of combining the set of items by finding similarities between the data according to characteristics found in the data. In the field of data mining, basically two types of privacy are concerned. Primarily, discovery of individual or susceptible data existed in the databases. Secondarily, susceptible patterns should not be disclosed when mining is performed on the collective data. Novel technologies are required for shielding susceptible information in electronic commerce [9]. Privacy preserving data mining allows mining useful information while preserving privacy of individuals [1]. Privacy issues are taking into account in two situations in the databases, i.e., centralized environment and scattered environment. In centralized environment, database is available in single site and privacy preserving data mining techniques are used to conceal susceptible data of individuals. In distributed database environment, data are scattered to multiple sites and privacy preserving data mining techniques are applied for integrating the data from multiple sites, without enlightening the privacy of individuals [12]. Zadeh introduced fuzzy sets to represent uncertainty, vagueness and provide formalized outfits for dealing with the consciousness inherent to many problems [16]. This paper proposed two methods in fuzzy-based data transformation for privacy preserving clustering in database environment which is centralized. First method proposed a fuzzy data transformation method, based on fuzzy membership functions to convert the original dataset. Second method proposed a hybrid method which is composed of Random Rotation Perturbation (RRP) and fuzzy data transformation approach.

2 Related Work and Background

A.F. Westin conducted privacy surveys about consumer privacy, general privacy and formed privacy indexes to sum up the results and discussed about privacy trends [3]. In Hippocratic databases, privacy issues are discussed and recognize the technological challenges, efforts in designing the databases and recommended some approaches that may possibly lead to solutions [2]. Oliveira and Zaiyane discussed the problem of protecting the essential attribute values in clustering when sharing the data and proposed a spatial transformation method [13]. They also presented an approach for hybrid data transformation [14]. Li and Zhang proposed hybrid data transformation approach for preserving privacy in clustering [10]. Mohammad Ali Kadampur et al. presented privacy preserving clustering approach through cluster bulging [6]. For privacy preserving clustering, Jie Liu et al. proposed random response method of geometric transformation in centralized database environment

[11]. B. Karthikeyan, et al. proposed a fuzzy-based approach for preserving privacy in clustering [8]. K. Chen and L. Liu discussed an approach for random rotation perturbation and structure of random rotation perturbation for privacy preserving classification [4].

3 Proposed Methods

In privacy preserving data mining, clustering analysis is commonly used tool. In this paper, two fuzzy-based methods are proposed. First method proposes, a data transformation method and a variety of experiments are conducted by altering the fuzzy membership functions such as Triangular membership function, Z-shaped fuzzy membership function and Gaussian membership function to convert the original dataset. In the subsequent method, a combination of fuzzy data transformation with various membership functions précised in first method and Random Rotation Perturbation (RRP) is proposed as a hybrid method. In an additional experiment, novel additive perturbation method is applied on unique dataset to get the distorted dataset for evaluation purpose.

3.1 Data Transformation Based on Fuzzy

In the process of data distortion, susceptible data values can be hiding without thrashing of information. A fuzzy transformation method distorts the responsive numerical attributes using fuzzy membership functions, i.e., Z-shaped fuzzy membership function (Zmf), Triangular fuzzy membership function (Trimf), and Gaussian fuzzy membership function (Gaussmf). Algorithm for fuzzy-based data transformation method is shown in Table 1. The input to the fuzzy transformation method is a dataset D which consists of sensitive attribute data in m rows and n columns. The input data D are transformed by initially suppressing the identifier attributes and distorting the dataset using fuzzy membership function.

3.2 Hybrid Method

To attain the twin objective of privacy and efficacy, a clustering technique is introduced for privacy preservation. The hybrid method is composed of the strength of active techniques and gives better outcomes, when compared to the single data perturbation approach. This method is composed of a combination of the two techniques which is Random Rotation Perturbation (RRP) and fuzzy data perturbation. The significant feature of RRP is preserving the geometric properties of the dataset. Hence, the distorted dataset is clustered with analogous precision, when

Table 1 Algorithm for fuzzy-based data transformation method

```
Input:

(a) Dataset D consists of sensitive attribute data in m rows and n columns.
(b) Fuzzy membership functions such as Zmf, Trimf, and Gaussmf.

Output: Distorted Datasets and each D′ consist of m rows and n columns.
Begin
          (1) Suppress the identifier attributes
          (2) For each membership function (Zmf, Trimf, Gaussmf)
                    (3) For each sensitive element in D do
                              (4) Convert the element using selected
                                  fuzzy membership function.
                    (5) End For
          (6) End For
          (7) Release the all distorted datasets for clustering analysis
End
```

clustering is performed on original dataset. In this method, the novel dataset is altered using fuzzy data transformation method, described in the Table 1, which will be specified as input for RRP method to get the ultimate distorted dataset. The subsequent table displays the algorithm for proposed hybrid method (Table 2).

Table 2 Algorithm for hybrid method

```
Input:
   (a) Original Dataset D consists of sensitive attribute data of size m x n.
   (b) Fuzzy membership functions such as Zmf, Trimf, and Gaussmf.

        Output: Distorted datasets and each D″ consist of size m x n.

Begin
          (1) Suppress the identifier attributes.
          (2) For each membership function (Zmf, Trimf, Gaussmf)
                    (3) For each sensitive element in D do
                              (4) Convert the element using
                                  fuzzy membership function
                    (5) End For
                    (6) Create an n x n rotation matrix R randomly.
                    (7) Obtain the final distorted dataset D″ = D x R.
          (8) End For
          (9) Release the distorted dataset D″ for clustering analysis.
End
```

Table 3 Algorithm for novel additive data perturbation

(1) Data owner acquires the input data from multiple parties by giving queries.
(2) Identify the sensitive data items and perform additive data perturbation on the selected values by adding small amount of noise to protect the values of sensitive data items.
(3) To enhance the privacy protection of additive data perturbation, perform swapping on the perturbed dataset obtained in step 2.
(2) Release the final distorted dataset to perform data mining operations such as classification and clustering.

3.3 Novel Additive Perturbation Method

In this section, a review of novel additive perturbation method is specified for privacy preserving data mining [7]. This method is used to amend the given input dataset to cover the extremely susceptible information. The additive data perturbation method is considered for scattered atmosphere, where a data holder desires to convert the input data extracted from cluster of parties. The distorted data are used to perform the data mining operations, i.e., clustering and classification. Table 3 shows the additive data perturbation algorithm.

4 Implementation

The evaluation of the data distortion methods is being performed by the proposed methods. Experiments are performed for implementing the proposed hybrid method and fuzzy data transformation approach on three actual living datasets taken from UCI [15]. Inclusion of Wine data set by means of thirteen attributes and 178 records, Iris data set with four attributes and 150 records, Zoo dataset with five numerical attributes and hundred instances are considered. In the first experiment, a fuzzy data transformation method is conducted to convert the novel dataset by altering the fuzzy membership functions such as Z-shaped fuzzy membership function, Triangular membership function and Gaussian membership function which is given in Table 1. In second experiment, a hybrid method is conducted by combining fuzzy data transformation approach précised in first experiment and Random Rotation Perturbation (RRP) as given in Table 2. The measurement of clustering quality is used by the k-means clustering algorithm. When the data are transformed, the original dataset clusters must be equivalent to the clusters in the deformed dataset. For testing clustering accuracy of the novel and tailored data base, Waikato Environment for Knowledge Analysis software [5] is used. The effectiveness is considered by misclassification error [13]. The misclassification error, M_E, is measured as follows:

Table 4 Misclassification error rates of fuzzy data transformation

Data distortion methods	Iris	Wine	Zoo
Zmf	0.1949	0.1784	0.2484
Trimf	0.30	0.34	0.4446
Gaussmf	0.3462	0.3570	0.4328
Novel additive	0.392	0.398	0.4864

$$M_E = \frac{1}{N} = \sum_{i=1}^{k} \left(|\text{cluster}_i(D)| - |\text{cluster}_i(D')| \right)$$

In the above formula

N is the number of points in the original dataset.

K is the number of clusters.

Cluster$_i$ (D) the number of data points of the ith cluster in the original dataset.

Cluster$_i$ (D′) the number of data points of the ith cluster in the transformed dataset.

The following table shows the M_E values obtained for the proposed fuzzy data transformation method. Elevated M_E values indicate inferior clustering quality, whereas lesser M_E shows the better clustering quality. The experiments are conducted 10 times and M_E value is taken as an average of 10. The misclassification error values of fuzzy data transformation approach are given in Table 4.

In Table 4, when comparing the misclassification error values, it is established that the proposed fuzzy data transformation methods which use the three membership functions give the inferior misclassification error for all the three datasets. Along with three membership functions, Z-shaped fuzzy membership function gives the inferior misclassification error.

It obviously indicates in all the three datasets that the proposed fuzzy data transformation method gives the inferior misclassification error for these three member functions, which are shown in Fig. 1. These results proved that fuzzy data transformation methods give higher efficacy than the novel additive data perturbation method. The misclassification error values of the proposed hybrid method are given in Table 5.

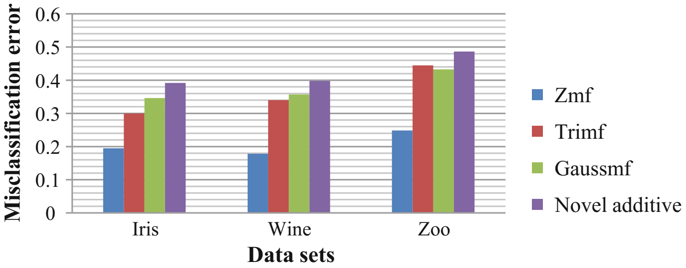


Fig. 1 Comparison of misclassification error values data transformation approaches

Table 5 Misclassification error rates of hybrid method

Data distortion methods	Iris	Wine	Zoo
Zmf	0.19492	0.1784	0.2484
Hybrid method (Zmf and RRP)	0.16238	0.13706	0.1902
Trimf	0.30	0.34	0.4446
Hybrid method (Trimf and RRP)	0.2666	0.2954	0.4
Gaussmf	0.3462	0.357	0.4328
Hybrid method (Gaussmf and RRP)	0.3198	0.3224	0.353
Novel additive	0.392	0.398	0.4864

Table 5 shows the misclassification error values intended for the proposed methods and novel additive data perturbation method. When comparing the misclassification error values of the fuzzy data transformation method with hybrid method, it indicates that the proposed hybrid method provides inferior misclassification error values for all the three datasets. Consequently, the hybrid method provides superior clustering quality and the altered dataset that is generated with the hybrid method looks extremely different from the novel dataset, which preserves the confidentiality of individuals. The following graphical representation shows the effectiveness of the proposed methods.

Using Z-shaped fuzzy membership function and novel additive perturbation methods, Fig. 2a shows the misclassification error values of fuzzy data transformation and hybrid method. Among these methods, hybrid method gives inferior misclassification error. So, the hybrid method effectively preserves the clustering accuracy and also provides privacy. Figure 2b shows the misclassification error values of the original additive perturbation method and triangular membership function used in fuzzy data transformation method and hybrid method. Among these methods, hybrid method gives lower misclassification error which provides higher data utility when compared to the fuzzy data transformation method and novel additive method. The misclassification error values of fuzzy data transformation method and hybrid method based on Gaussian membership function and novel additive method are shown in Fig. 2c. It obviously reveals that hybrid method yields inferior misclassification error. The misclassification error values on three datasets of fuzzy data transformation methods and hybrid method for membership functions are shown in Fig. 2d. It evidently shows that the proposed hybrid method gives inferior misclassification error for all the three member functions and for all these datasets.

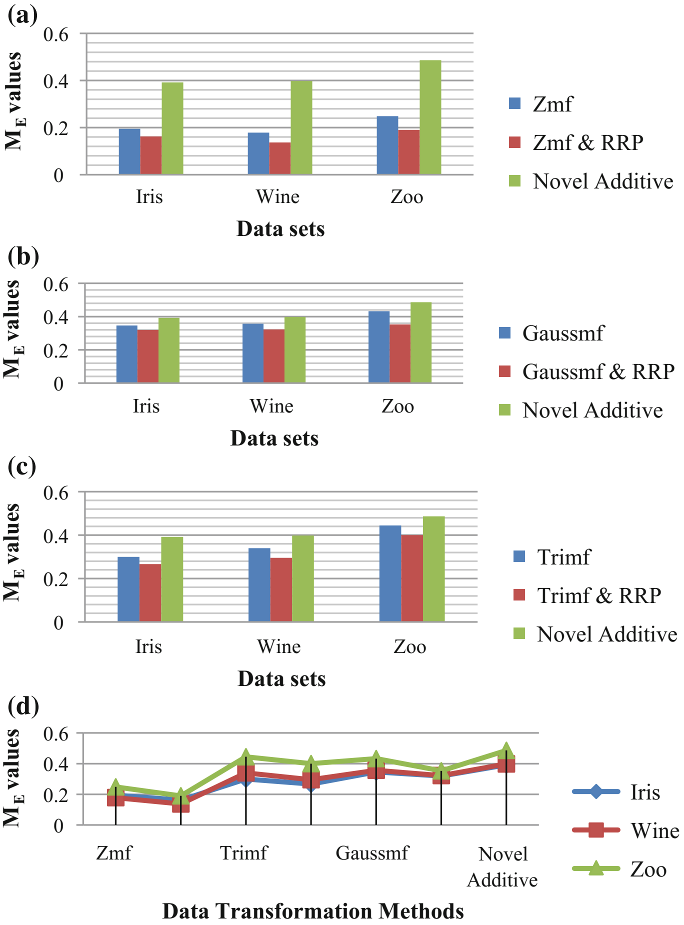


Fig. 2 Comparison of misclassification error values of fuzzy data transformation methods

5 Conclusion

In various organizations, privacy preservation plays a crucial role, when data consist of susceptible information and these data can be shared by different users. This paper considers the problem, defending individual privacy even as releasing the data for clustering study. In clustering analysis, random rotation is the accepted approach for data perturbation and it can protect privacy without affecting the accuracy. This paper proposed two methods. In first method, Z-shaped fuzzy membership function, Triangular membership function and Gaussian membership functions are being used in fuzzy-based transformation approach for data transformation. After experimentation on three types of actual life datasets from UCI, the results showed that the proposed method fulfilling the privacy limitations and continues to have the clustering superiority. Second method enhances the privacy preservation, which is a hybrid method and is proposed by adopting the fuzzy data transformation approach and random rotation perturbation. Experimentation on these three valid life datasets divulges that hybrid method is efficient for data consumption as well as privacy protection.

References

1. Ahmed, S.M.T., Haque, S., Tauhid, S.M.F.: A fuzzy based approach for privacy preserving clustering. Int. J. Sci. Eng. Res. **5**(2), 1067–1071 (2014)
2. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic Databases. In: Proceedings 28th International Conference on Very Large Databases (VLDB'02), pp. 143–154 (2002)
3. Westin, A.F.: Freebies and privacy: what net users think, Technical Report, Opinion Research Corporation (1999)
4. Chen, K., Liu, L.: A random rotation perturbation approach to privacy data classification. In: Proceedings IEEE International Conference on Data Mining (ICDM), pp. 589–592 (2005)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. **11**(1) (2009)
6. Kadampur, M.A., Somayajulu, D.V.L.N., Shivaji Dhiraj, S.S., Satyam Shailesh, G.P.: Privacy preserving clustering by cluster bulging for information sustenance. In: 4th International Conference on Information and Automation for Sustainability (ICIAfS'08), Colombo, Sri Lanka (2008)
7. Kamakshi, P., Babu, A.V.: A novel framework to improve the quality of additive perturbation technique. In: Proceedings International Journal of Computer Applications, vol. 30, No. 6 (2011)
8. Karthikeyan, B., Manikandan, G., Vaithiyanathan, V.: A fuzzy based approach for privacy preserving clustering. J. Theor. Appl. Inf. Technol. vol. 32, No.2 (2011)
9. Korba, L.: Privacy in distributed electronic commerce. In: Proceedings 35 Hawaii International Conference on System Science, Hawaii (2002)
10. Li,L., Zhang, Q.: A privacy preserving clustering technique using hybrid data transformation method. In: Proceedings IEEE international conference of grey systems and intelligent services (2009)
11. Liu J., XU, Y.: Harbin: privacy preserving clustering by random response method of geometric transformation. In: Proceedings Fourth International Conference on Internet Computing for Science and Engineering (2009)
12. Naga Lakshmi, M., Sandhya Rani, K.: Privacy preserving clustering based on fuzzy data transformation methods. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(8), 1027–1033 (2013)
13. Oliveira, S.R.M., Zaiyane, O.R.: Achieving privacy preservation when sharing data for clustering. In: Proceedings Workshop on Secure Data Management in a Connected World, in Conjunction with VLDB'2004, Toronto, Ontario, Canada, 67–82 (2004)
14. Oliveira, S.R.M., Zaiyane, O.R.: Privacy preserving clustering by data transformation. In: Proceedings 18th Brazilian Conference on Databases (2003)
15. UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html
16. Zadeh, L.A.: Fuzzy sets. Inf. Control, **8**, 338–353 (1965)

Cryptographic Key Extraction from Music

Chandan Kumar, Sandip Dutta and Soubhik Chakraborty

Abstract Random number generation is of utmost importance for cryptographic and other purposes like simulation. In this paper we present a novel algorithm that extracts cryptographic keys from music. The algorithm uses the sample values from a musical piece and a 128 bit password from the user. The algorithm then uses a 128 bit linear feedback register to produce a stream of random bits with input as binary stream of music and a feedback function at each clock pulse. The experimental results of our algorithm shows that 85 % of the streams pass all the tests while upon a proper filtering 98 % of the stream generated pass the NIST randomness test.

Keywords Cryptographic key · Random bit sequence · Pseudo random number generator

1 Introduction

Random number generation is an important task for various fields in statistical computing and its allied areas [1–3]. Random numbers are used in cryptography, simulation, especially using Monte Carlo method, computational modeling, biological computation, game theory, probability theory, quantum theory and finance [4–6]. As random numbers play a vital role in simulating randomness and unpredictability they find their importance in above-mentioned areas. To generate random numbers on large scale we need good quality of physical source of random

C. Kumar (✉) · S. Dutta
Department of CSE, Birla Institute of Technology, Mesra, Ranchi 835215, India
e-mail: chandankr@bitmesra.ac.in

S. Dutta
e-mail: sandipdutta@bitmesra.ac.in

S. Chakraborty
Department of Mathematics, Birla Institute of Technology, Mesra, Ranchi 835215, India
e-mail: soubhikc@yahoo.co.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_30

nature. Unfortunately, the physical sources which we used generally have low quality of randomness and are incompatible with the finite state generators and computer devices. The quality of randomness demanded by the above-mentioned domains is different depending on the use of the random sequences. Online games and gambling applications are severely dependent on the fact that how random the numbers are in nature.

Random numbers are used in cryptographic protocol, generating one time pads, generating nonce in cryptocurrencies. Hardware based random number generators use the random nature of physical phenomena. The time elapsed between particles emitted in radioactive decay, thermal noises produced by semiconductors, noise from the micro phone, jitters in the video streaming, transient fluctuation in electric and magnetic fields of electromagnetic devices [7, 8]. However, physical devices which produce random numbers are cumbersome in installation, are generally slow and costly, and are unable to produce the same sequence twice. Software based random number generator can exploit the current time stamp, system time, the time duration between key stroke patterns, operating system interrupts, networks statistics, number of peoples logged into a particular website, number of tweets per second in social media, number of interactions per second, contents of input–output buffers.

For generating random numbers a linear feedback registers can be used. Linear feedback shift registers (refer Fig. 1) are usually shift registers whose input bit is a function of the previous states of the shift registers. Generally linear feedback shift registers LSFRs have linear input function and use exclusive or for the output. The bit positions that affect the next state are called the taps. The feedback function block has a polynomial function whose coefficients determine the taps. The rightmost bit of the LFSR is the output bit. The taps are XOR'd sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream. Linear chaotic function can be used to generate random numbers. Generally 2-D and 3-D chaotic functions are used with cat maps and trent map functions to generate pseudo random sequences. The basic of the pseudo random number generator used linear congruential generator, quadratic residue generator. Blum Blum Shub generator is the example of quadratic residue generator. In [9] authors have used quantum properties to generate random numbers. In [10] the authors have used cellular automata to generate cryptographically secure random numbers. In [11] authors have used graphics processor for generating random

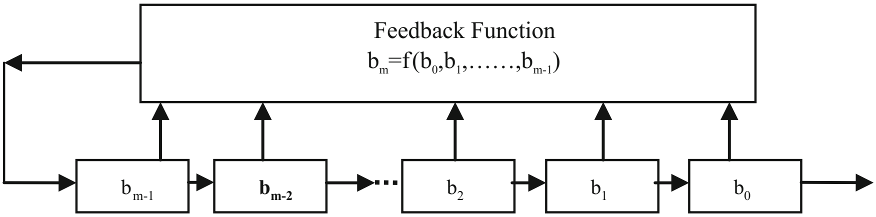


Fig. 1 Linear feedback shift register

numbers. Random number generators using Mersenne Twister on graphic processor have been tried in [12]. In the next section we describe the tests used by NIST for randomness check [13]. In [14, 15] authors have proposed audio random number generator which uses audio and video frames for generating random numbers. In [16] the author have generated random numbers from white noise of web camera. In [17] the author has used image for generating random numbers.

1.1 Statistical Test Suite for Random Number Generator

Generally, a number of test suites are available to check the randomness level of a random bit sequence [13, 18]. Diehard Battery of Tests [19] and Dieharder test are some of the test to check the randomness of a sequence. National Institute of Standards and Technology (NIST) has come up with a test suite comprising of 16 different tests. The details of these tests are mentioned below.

Frequency (Monobit) Test: This test determines whether a sequence of random bits have equal number of ones and zeros which should have been expected from a truly random sequence. This test assesses whether the ratio of ones and zeros are closer to half or not. All other subsequent tests are dependent on this test.

Frequency Test within a block: This test determine whether the number of ones and zeros in a non-overlapping block is same or not.

Runs Test: This test aims to find the total number of runs in the sequence, where a run will mean an uninterrupted sequence of similar bits. This test determines the oscillation between zeros and ones in the entire sequence.

Test for the longest run of ones in a block: This test aims to find the longest run of ones in a M-bit block. Any inconsistency in the fluctuation of ones will also lead to inconsistency in the longest run of zeros.

Binary Matrix Rank Test: This test aims to find the rank of disjoint sub-matrices of the entire subsequence. This test finds the linear dependence between fixed length substring of the entire sequence.

Discrete Fourier Transform (Spectral) Test: This test first finds the Discrete Fourier Transform of the sequence of bits and finds for the peak heights. The test aims to find the periodicity in the sequence which leads to the deviation from randomness.

Non-overlapping Template matching Test: This test aims to find the number of co-occurrences of a pretargeted string. The test finds the generators which produce too many occurrence of given non-periodic patterns. An m-bit pattern is used to search for co-occurrence of the same in entire sequence. The window slides while searching, if the pattern is not found it slides one bit otherwise it slides by m-bits

Overlapping Template matching Test: The test is same as of non-overlapping template matching but with a slight change. The m-bit window slides by a single bit whether the co-occurrence has been found or not.

Maurer's "Universal Statistical" Test: This test finds the number of bits between matching pattern. The idea is that, if the sequence can be compressed then there is a lack of randomness.

Linear Complexity Test: The test aims to find the number of linear feedback shift register. The idea is to determine whether the sequence is complex enough to be declared as random.

Serial Test: This test tries to find the deviation from the number of occurrences of overlapping patterns of *m* bits expected for random sequences.

Approximate Entropy Test: The aim of this test is to find the frequency of all possible overlapping m-bit patterns over the entire sequence. The test compares the frequency of overlapping blocks of two consecutive/adjacent lengths against the expected result for a random sequence.

Cumulative Sums (cusum) Test: This test replaces the bits in the sequence by (–1, +1) and computes the cumulative sums of subsequences of bits. The result for random sequences should be near zero.

Random Excursions Test: This test tries to find the deviation from the distribution of the number of visits of a random walk to a certain state.

Random Excursions Variant Test: This test tries to find the deviation from the distribution of the total number of visits across many random walks to a certain state.

2 Methodology

2.1 Proposed Key Extraction Scheme

The proposed scheme uses a music file provided by the user along with a 128 bit password to work upon. The password is used for the initialization of the seed value for the LSFR and the sample values of the music file is preprocessed and converted to binary sequence and then is provided as clocked input to the LFSR.

The propose scheme can be summarized as:

1. Arrange the music samples in a two dimensional array in a row major order.
2. Rearrange the sample element in a zig-zag scan order.
3. Properly quantize and scale the sample values.
4. Convert the entire data values in binary stream.
5. Feed the binary stream as one of the input to the 128 bit shift register with a predetermined feedback function. Initialize the registers of the feedback with the user provided password.
6. At each clock pulse feed the feedback value and the binary input from the music sample and shift values by one position.
7. Take each 128 bit output and test it for randomness. If the binary string passes the randomness test use it, otherwise use the filtering technique to make it a acceptable random sequence.

2.2 Pre-processing Step

The sample value of the music stream is first arranged in a square matrix discarding some of the samples which are above the nearest perfect square of the number of samples present in the music. Subsequently the sample values are recorded according to the zig-zag scan pattern refer Fig. 2d. This reordering is done to remove the correlation of nearby sample values. Then the sample values are again quantized and scaled. Say the representation of audio samples is between –1 and +1. The sample values are scaled to 0 through 255. A proper quantization value should be taken so that all the values in the range attain at least some values which can be further be analyzed using the histogram spread of the scaled samples. It has been observed that if the histogram of the sample values produce a Gaussian distribution the result are finer in a sense of getting a linear distribution of random number.

The sample values are then converted to its binary equivalents and concatenated one by one in the same order which was achieved in zig-zag scan. After the initial pre-processing this binary sequence is fed as an input to the 128 bit binary feedback shift register. For better results the silence zones of the music should be discarded, which is also to be included in preprocessing step.

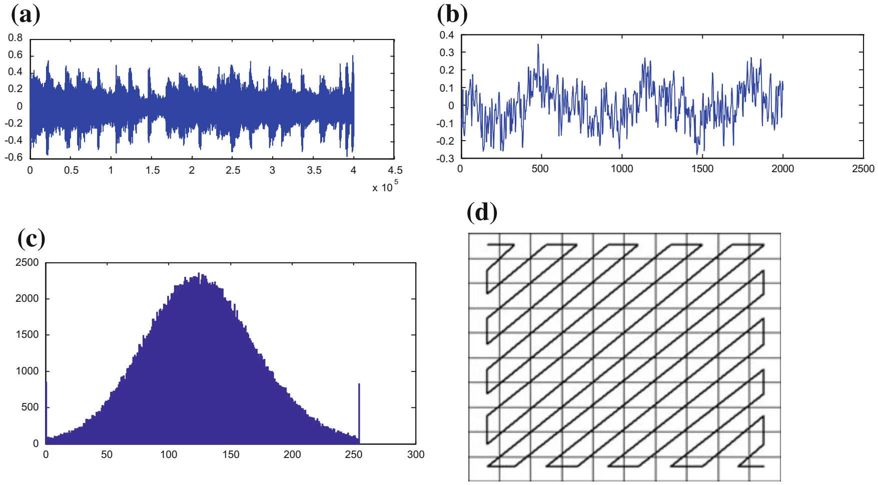


Fig. 2 **a** Shows the original musical sequence provided by the user as one of the input to the proposed system. **b** Shows the sample values of the music given as input. **c** Is the histogram of the sample values of the music provided. The sample values here were scaled to 0–255. **d** Shows the order in which the sample values are to be processed. The samples are first ordered in row major order and then a zig-zag scan is performed to take the sample values as input

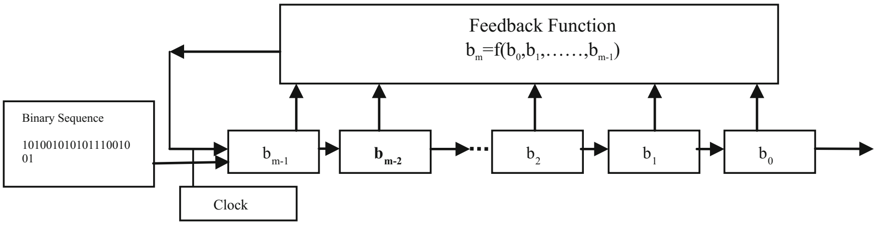


Fig. 3 Proposed LFSR scheme for extracting random sequence

2.3 Input to the Feedback Register and Updating the Vales of Feedback

The proposed LSER scheme for extracting random sequence is explained in Fig. 3.

The feedback function of the binary feedback shift register can be seen as a function (refer function 1)

$$\text{feed} = \mathrm{a}_0 R_0 + a_1 R_1 + a_2 R_2 + \cdots + a_{126} R_{126} + a_{127} R_{127}. \tag{1}$$

The values of the feedback registers are initialized with a user chosen password which is subsequently 16 character password. If the password is longer than 16 characters the character from 17 onwards are dropped while in case of password less than 16 characters the password is repeatedly concatenated to get 16 characters.

The values of the feedback registers are also updated using functions

$$R_{13} = b_7 + b_{13} + b_{17} + \cdots + b_{121} \tag{2}$$

$$R_{17} = b_5 + \ b_{19} + \cdots + b_{83} \tag{3}$$

$$.$$

$$R_{121} = b_0 + \cdots + b_{127} \tag{4}$$

These functions (2–4) are used to change the values of the feedback shift registers so that the cycle in the produced random bits is minimized. These functions can be inherent to the system where in a situation none of these are present in the system. At each clock pulse the binary input along with the feedback is given as an input to the shift register. The subsequent bits are shifted by one position giving up an output bit at each clock pulse.

2.4 Post-Processing and Getting Final Output

Take four subsequent 128 bit streams as output from the system say seq$_1$, seq$_2$, seq$_3$, seq$_4$. Xor them to get the final 128 bit random stream. Process all the subsequent group of 4 sequences and concatenate them together as a final random sequence. This step is done as a filtering to generate high quality randomness in the output stream. It was observed during experimentation that plain sequence output from the shift registers has a passing rate of 85 % while applying this filtering 98 % of the sequences passed all the tests, while 2 % were also showing statistical randomness but were lacking true randomness features.

3 Results and Discussion

When applying NIST SP 800-22rev1a 15 statistical tests on the proposed random number generator without filter, around 85 % of generated random number sequences passed the tests, while using the filtering 98 % of the sequences passed all the tests. The results of the test applied are listed in Table 1.

The proposed scheme does not directly use the sampled music values; rather it takes those values in a zig-zag scan order which removes the inherent correlation between the nearby sample values. The output of the feedback shift register has not been directly used to generate the random sequence while 4 chunks of 128 bit sequences have been used to get a 128 bit output using the whitening XOR

Table 1 Results obtained in the empirical evaluation with NIST test suite

Test	p-value	Conclusion
Frequency test	0.751621	Random
Frequency test within a block	0.484417	Random
Runs test	0.402143	Random
Test for longest run of ones in a block	0.212337	Random
Binary matrix rank test	0.687341	Random
Discrete fourier transform test	0.128756	Random
Maurer's universal statistical test	0.501256	Random
Linear complexity test	0.321379	Random
Non-overlapping template matching test	0.105432	Random
Overlapping template matching test	0.723678	Random
Approximate entropy test	0.235216	Random
Serial test	0.172205	Random
Cumulative sums test	0.098734	Random
Random excursion test	0.568542	Random
Random excursion variant test	0.105321	Random

(eXclusive OR) operation. These two modifications were done to remove the predictability and correlation between nearby sample values.

4 Conclusion and Future Scope

In this paper, we have proposed a linear feedback shift register based random number generator which uses music samples to generate cryptographically secure random sequence of bits. The results of the proposed scheme were tested with the standard test suite of NIST randomness test and the results were quite convincing. We are towards generating a cryptographically secure random number from sung musical notes. The biometric features of the singer could also be used for generating cryptographic keys from the musical pieces rendered by the singer.

References

1. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
2. Stallings, W.: Cryptography and Network Security. Prentice Hall, Englewood Cliffs, NJ (2003)
3. Blum, M., Silvio, M.: How to generate cryptographically strong sequences of pseudorandom bits. SIAM J. Comput. **13**(4), 850–864 (1984)
4. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM J. Comput. **15**(2), 364–383 (1986)
5. L'Ecuyer, P.: Random numbers for simulation. Commun. ACM **33**, 85–97 (1990)
6. Lee, I.: Parallel random number generations for Monte Carlo simulation. In: Proceedings of the 49th Annual Southeast Regional Conference ACM-SE'11, pp. 330–331 (2011)
7. L'Ecuyer, P.: Random number generation, in Handbook of Computational Statistics, pp. 35–71. Springer Handbooks of Computational Statistics, Springer (2012)
8. Alimomeni, M., Safavi-Naini, R.: Human assisted randomness generation using video games. IACR Cryptol. ePrint Archive **2014**, 45 (2014)
9. Shenoy, H.A., Srikanth, R., Srinivas, T.: Efficient quantum random number generation using quantum indistinguishability. Fluct. Noise Lett. **12**, Art. No. 1350020 (2013)
10. Chaudhuri, P.P., Chowdhury, D.R., Nandi, S., Chattopadhyay, S.: Additive Cellular Automata, Theory and Applications. IEEE Computer Society (1997)
11. Howes, L., Thomas, D.: GPU Gems 3, chapter Efficient Random Number Generation and Application Using CUDA. Addison-Wesley (2007)
12. Saito, M.: A variant of Mersenne Twister suitable for graphic processors (2011). http://arxiv.org/abs/1005.4973
13. National Institute of Standards and Technology: A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications (2010)
14. Chen, I.T., Tsai, J.M., Tzeng, J.: Audio random number generator and its application. In: 2011 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 4, pp. 1678–1683. IEEE (2011)
15. Chen, I.T.: Random numbers generated from audio and video sources. Mathematical problems in engineering (2013)

16. Tsai, J.M., Chen, I.T., Tzeng, J.: Random number generated from white noise of webcam. In: Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IIH-MSP'09, pp. 214–217. IEEE (2009)
17. Alsultanny, Y.A.: Random-bit sequence generation from image data. Image Vis. Comput. **26** (4), 592–601 (2008)
18. Soto, J.: Statistical testing of random number generators. In: Proceedings of the 22nd National Information Systems Security Conference (1999)
19. Marsaglia, G.: DIEHARD: a battery of tests of randomness (1996). http://stat.fsu.edu/~geo/diehard.html

A Programming Based Boosting in Super-Classifier for Fingerprint Recognition

Sumana Kundu and Goutam Sarker

Abstract A super-classifier with programming based boosting has been designed and established for fingerprint recognition. This multiple classifier set is comprised of three different classifiers. The first classifier is an OCA based modified RBFN with BP learning, second classifier is a combination of Malsburg learning and BP Network and third classifier is a SOM based modified RBFN with BP learning. These three individual classifiers perform fingerprint identification separately and these are fused together in a super-classifier which integrates the different conclusions using programming based boosting to perform the final decision regarding recognition. The learning of the system is efficient and effective. Also the performance measurement of the system in terms of accuracy, TPR, FPR and FNR of the classifier are substantially high and the recognition time of fingerprints are quite affordable.

Keywords Fingerprint recognition · OCA · Malsburg learning · SOM · BPN · RBFN · Programming based boosting · Holdout method · TPR · FPR · FNR

1 Introduction

Biometric identification is a technology, which identifies a person based on their physiology or behavioral characteristics. Fingerprint recognition is reliable and accurate biometric method that has been extensively used in a number of applications for a person's identity authentication. Fingerprint recognition is very effective in fields such as improving airport security, strengthening the national borders, in travel documents, in preventing ID theft, person authentication, access control system and retrieval of an identity from a database for criminal investigation

S. Kundu (✉) · G. Sarker
Computer Science and Engineering Department, NIT Durgapur, Durgapur, India
e-mail: sumana.kundu@yahoo.co.in

G. Sarker
e-mail: g_sarker@ieee.org

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_31

etc. Fingerprint recognition is a difficult task because the fingerprints diverge highly in terms of quality, size, shape, rotation and occlusion.

A RBFN with Optimal Clustering Algorithm (OCA) was established in [1, 2], for clear and occluded fingerprint identification and localization. A combination of Malsburg learning and Back propagation Network (BPN) was established in [3], for clear, occluded and rotated fingerprint recognition and rotation and location invariance localization of the fingerprints in different fingerprint image frames.

Many fingerprint identification methods which also already established are based on feature (minutiae) extraction and minutiae matching. This method mentioned in [4]. This methodology mainly involves extraction of minutiae points from the sample fingerprint images and then performing fingerprint matching based on the number of minutiae pairings among two fingerprints.

A technique for fingerprint identification by minutiae feature extraction using back-propagation algorithm has been approached in [5]. The digital values of extracted minutiae are applied as input to the neural network for training purpose. For fingerprint recognition, the verification part of the system identifies the fingerprint based on training performance of the network.

A fingerprint recognition algorithm using ellipsoidal basis function neural network (EBFNN) has been proposed in [6]. Here, fingerprint features are extracted by six-layer wavelet transform (WT) decomposition on binary images. Then, the extracted features are input into the designed EBFNN. Finally, EBFNN is trained and fingerprint recognition is accomplished by the trained EBFNN.

A principal component analysis (PCA) of symmetric sub-space model of neural network algorithm (SSA) for fingerprint recognition was presented in [7]. The convergence of Symmetric subspace algorithm (SSA) is also analyzed.

The above mentioned fingerprint recognition systems are unable to deal with noisy fingerprint images properly or give poor accuracy for the overall system due to poor fingerprint patterns. Therefore, in an attempt to alleviate such limitations, in this present paper, a multi-classification system has been designed and developed which contains three different classifiers to perform fingerprint recognition separately. These three different classifiers are an OCA based modified RBFN, a combination of Malsburg learning and BPN and a Self-organizing mapping (SOM) based modified RBFN respectively. Finally super-classifier draws the conclusion for proper identification of fingerprint based on programming based boosting method.

2 System Overview and Approach

2.1 Preprocessing of Fingerprint Patterns

The fingerprint patterns of training and test databases for learning as well as recognition have to be preprocessed to ultimately obtain perfect or enhanced image. The different steps in preprocessing for image enhancement which has been implemented are as follows.

1. Conversion of RGB fingerprints to gray scale patterns.
2. Removal of noise from fingerprint patterns.
3. De blurring of the patterns.
4. Background elimination.
5. Conversion of gray scale patterns into binary patterns.
6. Pattern Normalization.
7. Conversion of binary patterns into 1D matrix.

This 1D matrix file is the input to three different clustering algorithms of three different classifiers.

2.2 Theoretical Approach of the System

This present multiple classification system comprises of three different classifiers for fingerprint recognition. The first classifier is an OCA based modified RBFN with BP learning [1, 2]. Second classifier is a combination of Malsburg learning and BP Network [3]. Third classifier is a SOM based modified RBFN with BP learning. These three individual classifiers perform fingerprint recognition separately and the super-classifier concludes the final identification of the fingerprint based on programming based boosting method.

In the first classifier, OCA [8] is utilized to form clusters of the input fingerprint set which are taken as input of the Radial Basis Function Network (RBFN). It creates clusters of different qualities of fingerprints of every person and angle (person-angle). The mean "μ" and standard deviation "σ" of every cluster created by OCA with approximated normal distribution output function are used for each basis unit of RBFN. Then BP Learning Algorithm classifies the "person fingerprint-angle" into "person fingerprint". Here, the number of inputs equals to the number of fingerprint images of the training database, while the number of outputs sets to the number of classes and the number of hidden units is equal to the number of cluster created by OCA based on qualities of fingerprints of each "person-angle". Hence, the OCA based modified RBFN is used for fingerprint identification.

In the second classifier, a pattern clustering network or a competitive network (Malsburg Learning) [3, 9] is utilized to create groups of the input fingerprint set. It creates clusters of different qualities of fingerprints of every person and angle (person-angle). Then the Back Propagation network classifier is used to classify the "person fingerprint-angle" into "person fingerprint". Here, the number of inputs equals to the number of fingerprint images of the training database, the number of nodes in the hidden layer sets to the number of clusters produced by Malsburg learning network and the number of outputs sets to the number of classes. Hence, a combination of Malsburg Learning and Back Propagation network classifier is used for fingerprint recognition.

In the third classifier, a feature mapping network that is Self-organization network [10, 11] with Kohonen's learning is utilized to create clusters of the preprocessed input fingerprint set which are taken as input of the RBFN [1, 2].

The SOM can organize the two dimensional feature map from which the number of clusters can be evaluated directly. It creates clusters of different qualities of fingerprints of every person and angle (person-angle). The mean "μ" and standard deviation "σ" of every cluster created by SOM with approximated normal distribution output function are utilized for each basis unit of RBFN. Then BP Learning Algorithm classifies the "person fingerprint-angle" into "person fingerprint". Hence, the SOM based modified RBFN is used for fingerprint identification.

Radial Basis Function Network (RBFN)

The RBFN [1, 2, 12] comprised of three layers like an input layer for fingerprint pattern presentation, a hidden (clustering) layer consisting 'basis units' and an output (classification) layer. The clustering outputs (mean μ, standard deviation σ and corresponding approximated normal distribution output functions) are used in 'basis units' of RBFN. Thus, for the first and third classifier OCA and SOM are the first phase learning respectively and Back Propagation (BP) learning is the second phase of learning. The hidden layer use neurons with RBF activation functions describing local receptors. Then output node is used to combine linearly the outputs of the hidden neurons.

Programming Based Boosting

The system uses *programming based boosting* in super-classifier, i.e. super-classifier concludes the final identification of the fingerprint based on *programming based boosting* method considering the decisions of three individual classifiers. In case of programming, the weight of the vote of each classifier is pre assigned or 'programmed' beforehand. The weights of the different links from the individual classifiers into the integrator are programmed. These weights are the performances in terms of normalized *accuracy* of the individual classifiers.

Identification Learning with Training Fingerprints

The training database contains fingerprints of 50 different persons. For each person's fingerprint, three different qualities of fingerprints like hard press, medium press and soft press and for each person, also 3 different angular (0°, 90° and 180°) fingerprints are there. (Refer to Fig. 1)

After preprocessing all the fingerprint patterns are fed as input to three different classifiers. When the networks have learned all the different qualities of hard, medium and soft pressed fingerprints of all the angles (0°, 90° and 180°) for all the different persons, the networks are ready for identification of learned fingerprint images, which are termed as 'known' fingerprints. The fingerprints which were not learned during this process of learning are termed as 'unknown' fingerprints.

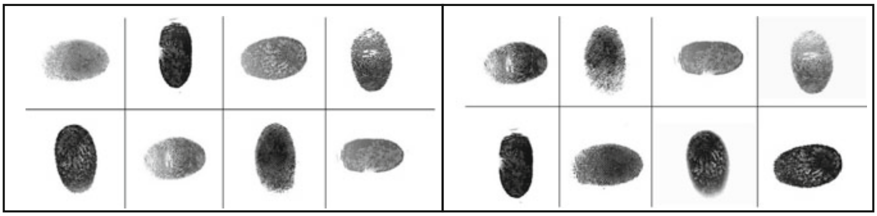


Fig. 1 Sets of few training and test fingerprint images

Identification Testing with Test fingerprints

The test database to evaluate the performance of the classifier with Holdout method consists of 50 different people's (same as training data set) fingerprints, and the qualities are different with reference to training database. It also consists of unknown fingerprints (which are not used during the training of networks) of various qualities (Refer to Fig. 1).

The test fingerprint patterns from the test database are fed as input to the preprocessor. The preprocessed fingerprints are fed as input to the previously trained networks of three individual classifiers. The trained BP network classifiers identify same person's fingerprint with different angular view as same person's fingerprint. After training, the networks of all the classifiers give high output values for known fingerprints and low output value for unknown fingerprints. To differentiate between known and unknown fingerprints a threshold value has been set. The threshold value is set as the mean of the minimum output value from known fingerprints and maximum output value from unknown fingerprints. The corresponding output value above threshold is considered as corresponding known fingerprint. The BP network produces different output activation indifferent overall output units. The normalized activation of each and every output unit represents the probability of belongingness of the input test fingerprint into the different classes. We considered the test fingerprint to belong to a class for which the normalized activation itself represents the probability of belongingness of that input test pattern into the particular category or classes. Finally super-classifier concludes the final identification of the fingerprint based on *programming based boosting* method considering the decisions of three individual classifiers. Here, the weight assigned for each link is the normalized *accuracy* of that corresponding classifier. Similarly, finally we calculate the probability of belongingness of the input test fingerprint for that corresponding class concluded by super-classifier by taking the minimum value of probability among three different classifiers. (Refer to Fig. 2 and the algorithm mentioned below)

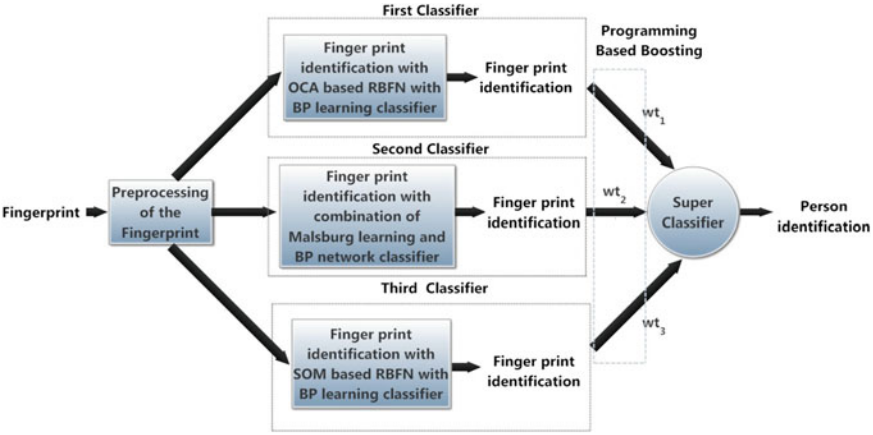

First Classifier
Finger print identification with OCA based RBFN with BP learning classifier
Finger print identification
Programming Based Boosting
wt$_1$
Second Classifier
Fingerprint
Preprocessing of the Fingerprint
Finger print identification with combination of Malsburg learning and BP network classifier
Finger print identification
wt$_2$
Super Classifier
Person identification
Third Classifier
Finger print identification with SOM based RBFN with BP learning classifier
Finger print identification
wt$_3$

Fig. 2 Block diagram of the proposed system for identification testing

Algorithm for fingerprint identification using super classifier:
Input: *Test folder containing known and unknown fingerprints.*
Output: *Identification of the given test fingerprint.*

1. *Get the fingerprint to test.*
2. *Preprocess the fingerprint.*
3. *Feed it as inputs to the trained BP networks of three different classifiers individually for fingerprint recognition.*
4. If the nth output for each of the BP networks is approximately 1 or above the threshold, then conclude the respective fingerprint to be of the nth person, otherwise conclude ‘unknown finger print’.
5. *Calculate the probabilities of belongingness of the fingerprint for particular classes from the normalized activation of the BP networks of three individual classifiers.*
6. *Get the results of three classifiers.*
7. *Input this result to super classifier.*
8. *Now the weights of the links of three individual classifiers become,* $wt_1 = \frac{w_1}{w_1 + w_2 + w_3}$, $wt_2 = \frac{w_2}{w_1 + w_2 + w_3}$ *and* $wt_3 = \frac{w_3}{w_1 + w_2 + w_3}$ *respectively, where* w_1, w_2 *and* w_3 *are the evaluated respective accuracies of 3 different single classifiers.*
9. *If all classifiers outputs as nth fingerprint, then super-classifier calculate the maximum weight among 3 links weight and concludes that fingerprint as nth fingerprint, otherwise conclude ‘unknown finger print’.*
10. *The probability of belongingness of the fingerprint into this nth class is,* $p = min(p_1, p_2, p_3)$, *where* p_1, p_2, p_3 *are the probabilities of belongingness into the classes for three individual classifiers.*
11. *If any two classifiers outputs as nth fingerprint (say, for fingerprint of person 2, 1st and 3rd classifier give outputs as person 2) and one classifier output as different (say, for fingerprint of person 2, 2nd classifier give output as person 1), then super classifier first sum up the weights (say,* $wt = wt_1 + wt_3$*) of the links of the classifiers which give same outputs, then compare this result with the third one (say,* wt_2*). Get the maximum weight, (say,* $mw = max(wt, wt_2)$*) and concludes the identification of the fingerprint as per this result.*
12. *The probability of belongingness of the fingerprint into the particular class depends on the previous step. If super classifier conclude the decision over the majority of the classifiers, then probability of belongingness of the fingerprint into the particular class is,* $p = min(p_i, p_j)$, *where* $i = j = 1$ *or 2 or 3. Otherwise,* $p = p_k$, *where* $i = j \neq k$.
13. *If all three classifier outputs as different, then the super-classifier calculate the maximum weight among 3 links and concludes the identification of the fingerprint as of that corresponding link otherwise as, “unknown fingerprint”.*
14. *The probability of belongingness of the fingerprint into the particular class is,* $p = min(p_1, p_2, p_3)$, *where* p_1, p_2, p_3 *are the probabilities of belongingness into the classes for three individual classifiers.*
15. *If any more testing is required, go to step 1.*
16. *Stop.*

If the output of two or more classifiers are contradictory in nature, then the conclusion obtain by the classifier with higher weighted link has to be accepted. So, the ensemble algorithm for super-classifier works well in such contradictory situations.

3 Result and Performance Analysis

The training and test dataset of fingerprint samples were taken from FVC 2000, FVC 2002 and FVC 2004 databases. (http://www.advancedsourcecode.com/fingerprintdatabase.asp).

3.1 Performance Evaluation Metrics of the Classifiers

Holdout method [3] is utilized to evaluate the performance of the classifier.

From the above mentioned confusion matrix (Refer to Fig. 3), if there are only two classes (say *X* and *Y*), then the accuracy is defined as follows:

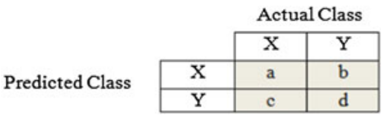
$$\text{Accuracy }(\%) = \frac{a+d}{a+b+c+d} \times 100 \tag{1}$$

If we take a two-class prediction problem (binary classification) [13], where the results are categorized either as positive (p) or negative (n), there are four possible results from a binary classifier. If the result from a prediction is p and the actual value is also p, then it is termed a *True Positive* (*TP*); however if the actual value is n then it is said to be a *False Positive* (*FP*). On the contrary, a *True Negative* (*TN*) has occurred when both the prediction result and the actual value are *n*, and *False Negative (FN)* is when the prediction result is n while the actual value is p. Now, *True Positive Rate* (*TPR*), *False Positive Rate* (*FPR*) *and False Negative Rate* (*FNR*) are defined as follows:

$$TPR\ (\%) = \frac{TP}{TP+FN} \tag{2}$$

$$FPR\ (\%) = \frac{FP}{FP+TN} \tag{3}$$

Fig. 3 Confusion matrix (2 class)



		Actual Class	
		X	Y
Predicted Class	X	a	b
	Y	c	d

$$FNR\ (\%) = \frac{FN}{FN + TP} \tag{4}$$

3.2 Experimental Results

The proposed system was developed to learn on a computer with Intel Core 2 Duo E8400, 3.00 GHz processor with 4 GB RAM and Windows 7 32-bit O.S.

Some salient portion of experimental result which handles contradictory situation (each classifier is identifying separate person) is given below:

```
According to first classifier given fingerprint is of person :3 with probability :0.39974
According to Second classifier given fingerprint is of person :2 with probability :0.92153
According to Third classifier given fingerprint is of person :1 with probability :0.83337
Super-classifier conclude, given fingerprint is of person :2 with probability :0.39974
```

Now if we use simple majority voting logic then super-classifier conclude the given fingerprint as of 'unidentifiable person' because 3 different classifiers give 3 different result of recognition. But when programing based boosting technique is used, the super-classifier concludes the given fingerprint as of person 2 with probability 0.39974. Thus the maximum weighted result with minimum probability (the probability which is safest to accept) is concluded.

From Table 1 we find that the accuracies of three different classifiers are 83.79, 88.97, 85.79 % and the accuracy of the super-classifier is 95.25 %. Thus, it is evident that the super-classifier is efficient for fingerprint identification than considering single classifiers individually. Also in Table 1, the present system displays low recognition time (<1 s) for individual fingerprint from the standard test data set. The limitation or drawback of the present system is that, the training time of this

Table 1 Accuracy of the classifiers (holdout method) and Fingerprint Learning time (in sec.)

Type of the classifiers	Accuracy (%)	Training time for 54 training samples	Recognition time (single test sample)	Total (learning) time
First classifier (OCA based RBFN)	83.79	66.403	0.0176	66.4206
Second classifier (Combination of Malsburg learning and BP network)	88.97	94.343	0.0063	94.3493
Third classifier (SOM based RBFN)	85.79	59.081	0.0248	59.1058
Super-classifier	95.25	219.827	0.0491	219.8761

Table 2 True positive rate of the classifiers for few people

Fingerprint	Classifiers			
	First classifier (%)	Second classifier (%)	Third classifier (%)	Super-classifier (%)
Person 1	100	100	100	100
Person 2	88.89	55.56	88.89	88.89
Person 10	100	100	100	100
Person 29	77.78	100	77.78	77.78
Person 30	100	77.78	100	100
Person 49	100	88.89	100	100
Person 50	66.67	66.67	100	100

Table 3 False positive rate of the classifiers for few people

Fingerprint	Classifiers			
	First classifier (%)	Second classifier (%)	Third classifier (%)	Super-classifier (%)
Person 1	0	0	0	0
Person 2	0	0	0	0
Person 10	0	0	0	0
Person 29	0	0	0	0
Person 30	1.33	0.22	1.33	1.33
Person 49	0	0	0.67	0
Person 50	0.67	0	0.67	0

Table 4 False negative rate of the classifiers for few people

Fingerprint	Classifiers			
	First classifier (%)	Second classifier (%)	Third classifier (%)	Super-classifier (%)
Person 1	0	0	0	0
Person 2	11.11	44.44	11.11	11.11
Person 10	0	0	0	0
Person 29	22.22	0	22.22	22.22
Person 30	0	22.22	0	0
Person 49	0	11.11	0	0
Person 50	33.33	33.33	0	0

system is quite high. But training is only for one time while recognition is for multiple times. Once the training completes, recognition is possible for different inputs many times as per user choice. So with the help of multiple classifiers we get accurate recognition with minimum recognition time at the cost of training time.

From Tables 2, 3 and 4, True Positive Rate, False Positive Rate and False Negative Rate clarify the performance of few classes for few people's fingerprints.

Table 5 Comparative study with accuracy of the systems

Systems	Accuracy (%)
Minutiae matching	65–70
Minutiae extraction-BPN	95
EBFNN	91.8
PCA-SSA	91.67
Present system	95.25

Table 5 display a comparative study of the proposed system in terms of accuracy with other systems [4–7] mentioned in Sect. 1. Hence, the proposed method displays improvement in terms of accuracy with low recognition time as compared to techniques mentioned in the Sect. 1.

4 Conclusion

In the present system instead of using a single classifier, an attempt has been made to employ multiple classifiers for fingerprint recognition. The advantage is that we need not rely on a single classifier and instead the decision coming out of the different types of classifiers are suitably integrated based on programming based boosting method. Thus, the different conclusions from individual classifiers are fused together to find out the most reliable conclusion. This multi-classification system gets the general advantages of three different hybrid classifiers. The performance evaluation in terms of accuracy, TPR, FPR, FNR with Holdout method is quite high for fingerprint patterns. Also the recognition time is moderately low for different types of fingerprints. The present multi-classification system is efficient and effective compared to most other conventional fingerprint identification technique.

The work can be extended by increasing the number of recognizable person's fingerprints and number of qualities of fingerprints. Also occluded and rotated fingerprints with different angles may be identified by this system.

References

1. Kundu, S., Sarker, G.: A modified radial basis function network for occluded fingerprint identification and localization. IJCITAE **7**(2), 103–109 (2013)
2. Sarker, G., Kundu, S.: A modified radial basis function network for fingerprint identification and localization. In: International Conference on Advanced Engineering and Technology, pp. 26–31 (2013)
3. Kundu, S., Sarker, G.: A modified BP network using Malsburg learning for rotation and location invariant fingerprint recognition and localization with and without occlusion. In: Seventh International Conference on Contemporary Computing, pp. 617–623 (2014)

4. Bana, S., Kaur, D.: Fingerprint recognition using image segmentation. Int. J. Adv. Eng. Sci. Technol. **5**(1), 012–23 (2011)
5. Chatterjee, A., Mandal, S., Rahaman, G.A., Arif, A.S.M.: Fingerprint identification and verification system by minutiae extraction using artificial neural network. JCIT **1**(1), 12–16 (2010)
6. Luo, J., Lin, S., Ni, J., Lei, M.: An improved fingerprint recognition algorithm using EBFNN. In: Second International Conference on Genetic and Evolutionary Computing, pp. 504–507 (2008)
7. Yu, C., Jian, Z., Bo, Y., Deyun, C.: A novel principal component analysis neural network algorithm for fingerprint recognition in online examination system. In: 2009 Asia-Pacific Conference on Information Processing, pp. 182–186 (2009)
8. Sarker, G.: An unsupervised natural clustering with optimal conceptual affinity. IJCITAE **2** (1), 45–52 (2008)
9. Sarker, G.: An unsupervised learning network for face identification and subsequent localization. In: International Conference on Communications, Devices and Intelligent Systems (CODIS), pp. 632–635 (2012)
10. Turky, A.M., Ahmad, M.S.: The use of SOM for fingerprint classification. In: International Conference on Information Retrieval and Knowledge Management, pp. 287–190 (2010)
11. Yegnanarayana, B.: Artificial Neural Networks. PHI Learning Private Ltd, pp. 223–228 (1999)
12. Aziz, K.A.A., Ramlee, R.A., Abdullah, S.S., Jahari, A.N.: Face detection using radial basis function neural networks with variance spread value. In: International Conference of Soft Computing and Pattern Recognition, pp. 399–403 (2009)
13. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)

A Super Classifier with Programming-Based Boosting Using Biometrics for Person Authentication

Sumana Kundu and Goutam Sarker

Abstract A boosting-based multi-classification system has been designed and established using different biometric features, that is, fingerprints of both right and left hands and handwriting for authenticated person identification. This multi-classifier comprises of three different classifiers. The first classifier is an Optimal Clustering Algorithm (OCA)-based modified Radial Basis Function Network (RBFN) with Back Propagation (BP) learning, second classifier is a Heuristic Based Clustering (HBC) algorithm-based modified RBFN with BP learning and third classifier is a combination of Malsburg learning and BP Network. These three individual classifiers identify fingerprints of both right and left hands and handwriting, respectively, and the super classifier perform the fusion of three conclusions to establish the final decision based on programming-based boosting method for person authentication. The technique of using multiple classifiers in a single system is efficient and effective. Also the accuracy, precision, recall, and F-score of the classifiers are substantially moderate and the training and testing time of all the biometrics are quite low and affordable.

Keywords Fingerprint identification · Handwriting identification · OCA · HBC · RBFN · Malsburg learning · BPN · Programming based boosting · Holdout method · Precision · Recall · F-score

1 Introduction

Person authentication using different biometric traits is very fruitful and necessary in many security systems. Single biometric systems have limitations, like uniqueness, high spoofing rate, high error rate, non-universality, and noise. Multimodal

S. Kundu (✉) · G. Sarker
Computer Science and Engineering Department, NIT Durgapur, Durgapur, India
e-mail: sumana.kundu@yahoo.co.in

G. Sarker
e-mail: g_sarker@ieee.org

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_32

biometric systems overcome some of these problems by strengthening the proof picked up from a number of sources. Biometric features are collected from various sources to identify a person. Different features can be inspected by a single system or different systems and their conclusions can be merged together. Most of the multimodal systems are based on fusion technique and use two biometric features for identification.

A feature (minutiae) extraction and minutiae matching technique mentioned in [1] comprises extraction of minutiae points from the sample fingerprint images and makes fingerprint matching based on the number of minutiae pairings among two fingerprints.

A technique for fingerprint identification by minutiae feature extraction use back propagation [2], where the digital values of extracted minutiae are taken as input of the neural network for training. The authentication part of the system identifies the fingerprint depending on training performance of the network.

In a fingerprint recognition algorithm using ellipsoidal basis function neural network (EBFNN) [3], fingerprint features are extracted by six-layer wavelet transform (WT) decomposition on binary images. Then, the extracted features are fed to the designed EBFNN to train and perform finger print recognition.

A principal component analysis (PCA) of symmetric subspace model of neural network algorithm (SSA) for fingerprint recognition was presented in [4]. The convergence of Symmetric subspace algorithm (SSA) is also analyzed.

An automatic handwriting identification using scanned images of handwriting with feed forward neural network was presented in [5]. The feed forward neural network architecture was: 400 neurons in input layer, 200 neurons in one hidden layer, and 3 neurons in output layer (in this case for three hand writers).

A wavelet-based GGD method was used instead of traditional 2-D Gabor filters for handwriting-based writer identification in [6]. In GGD method, first, decomposed the handwriting image via db4 wavelet transform at three levels, and then applied GGD model on the wavelet decomposition sub-bands except the HH sub-band at the finest scale.

A writer identification system was presented in [7], that uses Radial Basis Function (RBF) in the Off-line mode. This method is text-independent and utilizes text lines as basic entities, from which features are extracted. For every writer a recognizer was trained and unknown input text line was presented to each recognition system.

A multi-classification system based on simple bagging method was developed in [8]. In this system fingerprint, iris and face were used individually in OCA-based modified RBFN classifier. Finally, an integrator concludes the decision of person identification based on simple voting logic.

In this paper, we propose a multi-classification system for person identification where each classifier operates on different aspects of the input. There are three individual classifiers for Right hand fingerprints, Left hand fingerprints, and Handwriting identification and also a Super classifier which give the proper identification of person based on programming-based boosting logic considering the result of three individual classifiers.

In the present work, the adopted ANN models are the Radial Basis Function Network (RBFN) with Optimal Clustering Algorithm (OCA) for training units and Back Propagation (BP) learning for classification, the RBFN with Heuristic Based Clustering (HBC) for training units and Back Propagation (BP) learning for classification and a combination of Malsburg Learning and Back Propagation network for training the classification. Finally, super classifier draws the conclusion for proper identification of the person based on programming-based boosting method.

2 Overview of the System and Approach

2.1 Preprocessing

All the fingerprint images of right and left hands, and handwriting images of training and test datasets have to be preprocessed before learning as well as recognition. The preprocessing steps are described below:

1. Conversion of RGB images to gray scale images.
2. Removal of noise from the images.
3. De blurring the images.
4. Background elimination.
5. Conversion of grayscale images into binary images.
6. Image Normalization/Image Compression.
7. Conversion of binary images into 1D matrix.

In the sixth step, all handwriting image patterns are normalized into 20 × 70 pixels and the fingerprint images of the left and right hands are compressed instead of normalization. The fingerprint images are compressed by replacing block of pixels with the mode value of the pixel intensities where mode is the intensity value which has occurred more frequently in the block. At the final step, the 1D matrix file sets of the right hand fingerprints, left hand fingerprints, and the handwriting images are the inputs to the OCA, HBC, and the Malsburg Learning Network, respectively.

2.2 Theory of the Operation

This present multiple classification system comprises of three different classifiers for right hand fingerprints, left hand fingerprints, and handwriting identification, respectively. The first classifier is an OCA-based modified RBFN with BP learning [9, 10]. Second classifier is a HBC-based modified RBFN with BP learning [11] and third classifier is a combination of Malsburg learning and BP Network [12]. These three individual classifiers perform right and left hand fingerprints and handwriting identification separately and the super classifier concludes the final identification of the person based on programming-based boosting method.

In the first classifier, OCA [13] is utilized to create clusters of the preprocessed input right hand fingerprint (thumb, second, third, and fourth finger—as per standard dataset CASIA version 5) pattern set which are taken as input of the RBFN. Here OCA creates clusters of different qualities of fingerprints of every person and individual fingers (person-finger). The mean "μ" and standard deviation "σ" of each cluster created by OCA with approximated normal distribution output function are utilized in each basis unit of RBFN. Then BP Learning Algorithm classifies the fingerprints of different fingers of a person into "person fingerprint". Hence, the OCA-based modified RBFN is used for right hand fingerprints identification.

In the second classifier, HBC [14] algorithm is utilized to create clusters of the preprocessed input left hand fingerprints (thumb, second, third, and fourth finger—as per standard dataset CASIA version 5) pattern set which are taken as input of the RBFN. Here, HBC also creates clusters of different qualities of fingerprints of every person and individual fingers (person-finger). Then, the RBFN with Back propagation network classifier is utilized to classify the fingerprints of different fingers of a person into "person fingerprint" like first classifier. Hence, the HBC-based modified RBFN is used for left hand fingerprints identification.

In the third classifier, a pattern clustering network, i.e., a competitive network (Malsburg Learning) [12, 15] is utilized to create clusters of the preprocessed input handwriting data set. It creates clusters of different qualities of handwritings (name and surname) of each person. Then the BP network classifier is used which classifies the "person name-person surname" into "person name". Here, the number of inputs equals to the number of handwriting images of the training database, the number of nodes in the hidden layer sets to the number of clusters produced by Malsburg learning network and the number of output sets to the number of classes. Hence, a combination of Malsburg Learning and BP network classifier is used for handwriting identification.

Radial Basis Function Network (RBFN)

The RBFN [7, 9, 10] comprised of three layers like an input layer for pattern presentation, a hidden (clustering) layer consisting 'basis units' and an output (classification) layer. The clustering outputs (mean μ, standard deviation σ, and corresponding approximated normal distribution output functions) are utilized in 'basis units' of RBFN.

Thus, for the first and second classifier OCA and HBC are the first phase of learning, respectively, and Back Propagation (BP) learning is the second phase of learning (Refer to Fig. 1.). The output node is used to combine linearly the outputs of the hidden neurons. So, the output of the hidden units,

$$y = e^{\frac{-(x-\mu)^2}{\sigma^2}}. \tag{1}$$

Programming-Based Boosting

In this system super classifier concludes the final identification of the person based on *programming-based boosting* method considering the decisions of three individual classifiers. In case of programming, the weight of the vote of each

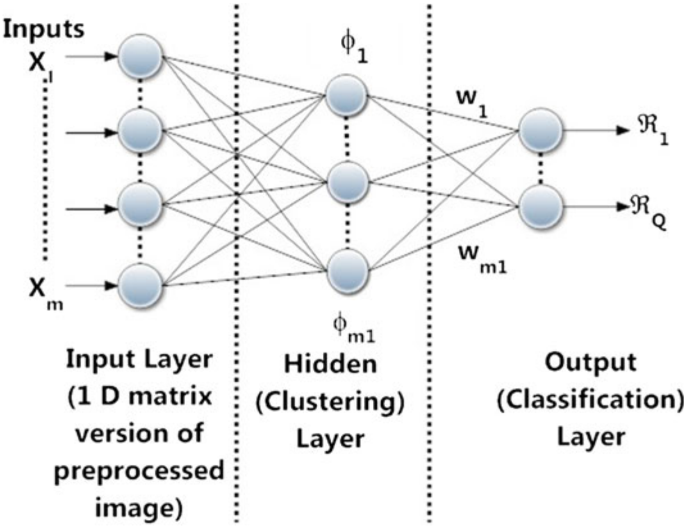

Inputs
X_1
X_m
ϕ_1
ϕ_{m1}
w_1
w_{m1}
$\Re_1$
$\Re_Q$
Input Layer (1 D matrix version of preprocessed image)
Hidden (Clustering) Layer
Output (Classification) Layer

Fig. 1 Modified RBFN architecture

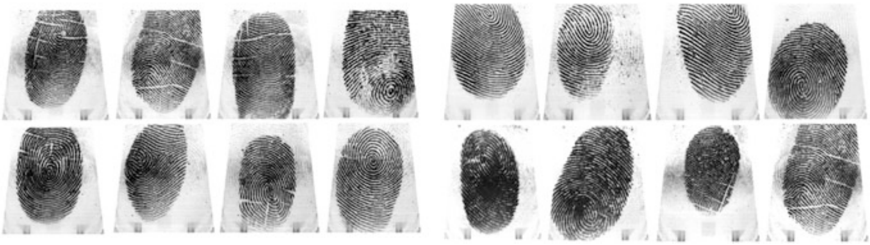


Fig. 2 Sets of few training and test right hand finger print images

classifier is preassigned or 'programmed' beforehand. The weights of the different links from the individual classifiers into the integrator are programmed. These weights are the performances in terms of normalized *accuracy* of the individual classifiers.

Identification Learning with Training Patterns

There are three training databases for three individual classifiers. Each database consists of different image patterns, i.e., right and left hand fingerprints and handwriting patterns of four different persons. In the right and left hand fingerprints, individual training databases, for every person's fingerprint, three different qualities of finger prints like hard press, medium press, and soft press and for each person, fingerprints of four different fingers (thumb, second, third, and fourth finger—as per standard database CASIA version 5) are also included. The handwriting training database contains six different qualities of handwritings (name and surname separately) for each person. (Refer to Figs. 2, 3 and 4).

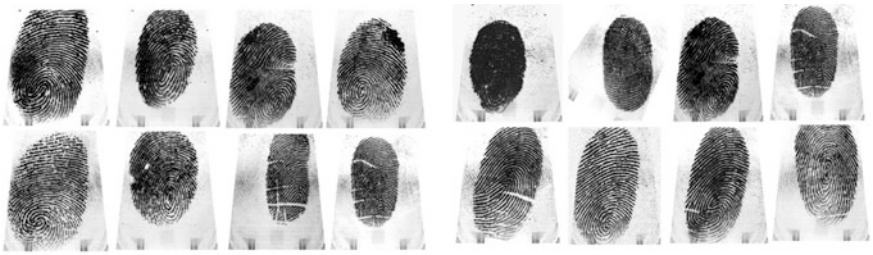


Fig. 3 Sets of few training and test left hand finger print images

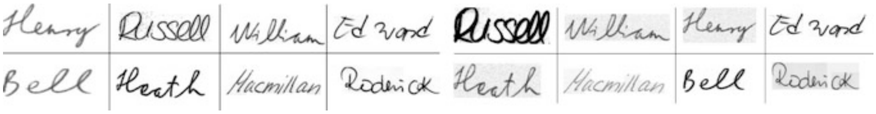


Fig. 4 Sets of few training and test handwriting images

After preprocessing, all the patterns are fed individually as input to the different clustering algorithm or clustering networks of individual classifiers. When the networks have learned all the different patterns of training databases for all different people, the networks are ready for identification of learned image patterns, which are termed as 'known' patterns. The patterns which were not learned during this process of learning are termed as 'unknown' patterns.

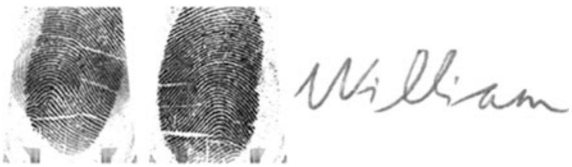
Identification Testing with Test patterns

The test sets for testing to estimate the performance of individual classifiers with *Holdout method* contain four different people's (same as training data set) patterns (right and left hand fingerprints and handwriting) of various qualities. These patterns are totally different from training databases. (Refer to Figs. 2, 3 and 4).

The test set for testing to estimate the performance of super classifier contains pattern sets for four different people (same as training data set), i.e., each pattern set comprises of one right hand fingerprint, one left hand fingerprint and one handwriting images. The patterns of each pattern set are also of various qualities which are also completely different from training set. (Refer to Fig. 5).

The test sets for individual three classifiers and for super classifier also contain some unknown fingerprints of both right and left hands and handwriting patterns of various qualities which are not included to train the classifiers.

Fig. 5 A sample of test set (person 2) for super classifier for person identification



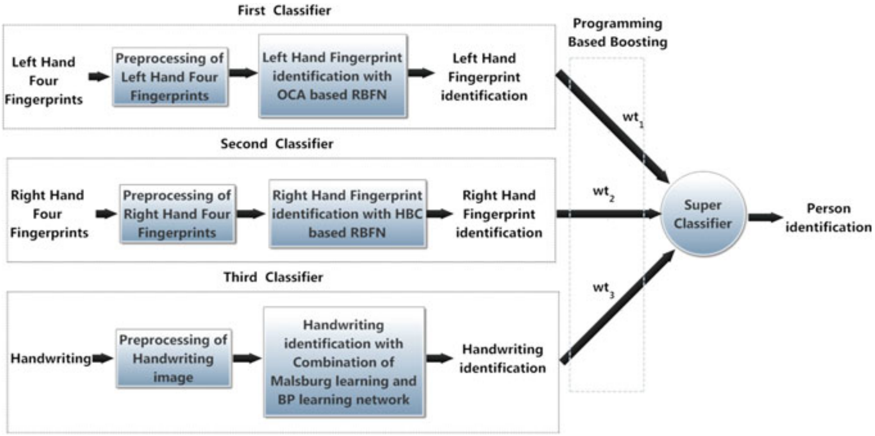

First Classifier
Left Hand Four Fingerprints
Preprocessing of Left Hand Four Fingerprints
Left Hand Fingerprint identification with OCA based RBFN
Left Hand Fingerprint identification
Second Classifier
Right Hand Four Fingerprints
Preprocessing of Right Hand Four Fingerprints
Right Hand Fingerprint identification with HBC based RBFN
Right Hand Fingerprint identification
Third Classifier
Handwriting
Preprocessing of Handwriting image
Handwriting identification with Combination of Malsburg learning and BP learning network
Handwriting identification
Programming Based Boosting
wt1
wt2
wt3
Super Classifier
Person identification

Fig. 6 Block diagram of the proposed system for identification testing

The test patterns from the test image sets are fed as input to three individual preprocessors. The preprocessed patterns are fed as input to the previously trained networks of three individual classifiers. After training, the networks of all the classifiers give high output values for known patterns and low output value for unknown patterns. To differentiate between known and unknown patterns a threshold value has been set. The corresponding output value above threshold is considered as corresponding known patterns. Finally super classifier concludes the final identification of the person based on *programming-based boosting* method considering the decisions of three individual classifiers. Here, the weight assigned for each link is the normalized *accuracy* of that corresponding classifier. (Refer to Fig. 6. and the algorithm mentioned below).

Algorithm for person identification using super classifier:

Input: *Test folder containing known and unknown pattern sets. Each pattern set consists of one right hand fingerprint, one left hand fingerprint, and one handwriting pattern.*

Output: *Identification of the given test pattern set.*

1. *Get the pattern set to test.*
2. *Preprocess each pattern (right hand fingerprint, left hand fingerprint, handwriting) individually with the preprocessor of individual three classifiers.*
3. *Feed these preprocessed patterns as inputs to the previously trained BP networks of individual classifiers for pattern recognition.*
4. *If the nth outputs (1 $\leq$ n $\leq$ 4) of the BP networks are approximately 1, then conclude the respective pattern as n.*
5. *Get the results of three classifiers.*
6. *Input this result to super classifier.*
7. *Now, the weights of the links of three individual classifiers become, wt_1, wt_2, wt_3. Where $wt_1 = \frac{w_1}{w_1 + w_2 + w_3}$, $wt_2 = \frac{w_2}{w_1 + w_2 + w_3}$, and $wt_3 = \frac{w_3}{w_1 + w_2 + w_3}$, respectively.*

Here, w_1, w_2, and w_3 are the respective accuracies of three individual classifiers.

8. *If all classifiers output as nth (1 ≤ n ≤ 4) pattern, then super classifier calculate the maximum weight among the weights of three links and concludes that pattern as nth pattern.*
9. *If any two classifiers output as nth (1 ≤ n ≤ 4) pattern (say, for pattern of person 2, 1st and 3rd classifier give outputs as person 2) and one classifier outputs as different (say, for pattern of person 2, 2nd classifier give output as person 1), then super classifier first sums up the weights (say, $wt = wt_1 + wt_3$) of the links of the classifiers which give same outputs, then compare this result with the third one (say, wt_2).Get the maximum weight,(say, $mwt = max(wt, wt_2)$) and concludes the identification of the pattern as per the output corresponding to maximum weight.*
10. *If all three classifier outputs as different, then the super classifier calculates the maximum weight among three links and concludes the identification of the pattern as of that corresponding link otherwise as, "unidentifiable person".*
11. *If any more testing is required, go to step 1.*
12. *Stop.*

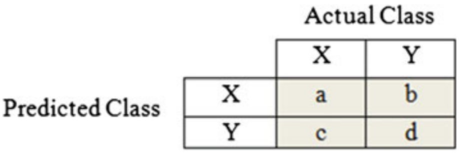
3 Result and Performance Analysis

We were unable to gather all the different biometric patterns from one standard database. That is why it was assumed that, different biometric patterns of different standard databases were of same particular people. The training and test datasets for Right and Left hand Fingerprint samples were taken from CASIA Fingerprint Image Database Version 5.0 (http://biometrics.idealtest.org/dbDetailForUser.do?id=7) and Handwriting samples from IAM handwriting database (http://www.iam.unibe.ch/fki/databases/iam-handwriting-database/download-the-iam-handwriting-database).

3.1 Performance Evaluation Metrics of the Classifiers

Holdout method [12] is utilized to evaluate the performances of the classifiers (Fig. 7).

Fig. 7 Confusion Matrix (2 class)



		Actual Class	
		X	Y
Predicted Class	X	a	b
	Y	c	d

From the above mentioned confusion matrix, if there are only two classes (say X and Y), then the accuracy, precision, recall, and F-score [12] are defined as follows,

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} \times 100, \tag{2}$$

$$\text{Precision} = \frac{a}{a+b}, \tag{3}$$

$$\text{Recall} = \frac{a}{a+c}, \tag{4}$$

$$F - \text{score} = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}. \tag{5}$$

For the performance evaluation of the classifier, by utilizing holdout method we were capable to test patterns which were excluded in training pattern set. In evaluation of a classifier with accuracy metric, the overall performance of the classifier is reflected irrespective of the individual performance evaluation for every class. This is more suitable for evaluating the system performance through a specific numeric value. Precision, recall, and F-score metrics were utilized to clarify the performance of each class.

3.2 Experimental Results

The proposed system was made to learn on a computer with Intel Core 2 Duo E8400, 3.00 GHz processor with 4 GB RAM and Windows 7 32-bit Operating System. (Refer to Tables 1, 2 and Figs. 2, 3, 4, 5).

From Table 1, we find that the accuracies of three different classifiers for three different biometric features (left hand fingerprints, right hand fingerprints, and

Table 1 Accuracy of the classifiers and Learning time (in seconds) of the biometric features

Type of the biometrics	Accuracy (%)	Training time (No. of Training samples for each biometric feature = 48)	Testing time (Single test sample)	Total (Learning) time
First classifier (Right Hand Fingerprints)	81.67	13.198	0.0143	13.2123
Second classifier (Left Hand Fingerprints)	95.00	19.051	0.0143	19.0653
Third classifier (Handwriting)	95.00	33.704	0.0043	33.7083
Super classifier	96.67	65.953	0.0333	65.9863

Table 2 Performance measurement of the classifiers with holdout method

Performance evaluation metrics person wise		1st classifier (Right Hand Fingerprints)	2nd classifier (Left Hand Fingerprints)	3rd classifier (Handwriting)	Super classifier
Precision	Person 1	1.00000	1.00000	1.00000	1.00000
	Person 2	1.00000	1.00000	1.00000	1.00000
	Person 3	1.00000	0.85714	0.92308	1.00000
	Person 4	1.00000	1.00000	1.00000	1.00000
	Unknown	0.52174	0.90909	0.84615	0.85714
Recall	Person 1	0.66667	0.91667	0.83333	0.83333
	Person 2	0.75000	1.00000	1.00000	1.00000
	Person 3	0.75000	1.00000	1.00000	1.00000
	Person 4	0.91667	1.00000	1.00000	1.00000
	Unknown	1.00000	0.83333	0.91667	1.00000
F-score	Person 1	0.80000	0.95652	0.90909	0.90909
	Person 2	0.85714	1.00000	1.00000	1.00000
	Person 3	0.85714	0.92308	0.96000	1.00000
	Person 4	0.95652	1.00000	1.00000	1.00000
	Unknown	0.68571	0.86957	0.88000	0.92308

handwriting) are 81.67, 95.00, and 95.00 %, and the accuracy of the super classifier is 96.67 %. Thus, it is evident that the super classifier is efficient for person identification than considering single classifiers using single biometric features individually. In Table 2, precision, recall, and *F*-score metrics explain the performance of every class with holdout method. Also, in Table 1, the proposed system displays overall low testing time (<1 s) for the standard test data sets. Hence, the proposed approach displays improvement both in terms of accuracy and testing time as compared to uni-modal systems mentioned in the Sect. 1.

4 Conclusion

In the present system, instead of using a single classifier with a single biometric feature for person identification/authentication, an attempt has been made to employ multiple classifiers acting on the different biometric features. The advantage is that we need not rely on a single classifier acting on a particular biometric and instead, the decision coming out of the different types of classifiers using different biometric features are suitably integrated based on weighted voting logic. Thus, the different conclusions from individual classifiers are fused together to find out the most reliable conclusion. The performance evaluation in terms of accuracy, precision, recall, F-score with Holdout method for individual three classifiers and also for super classifier is moderately high for different biometric traits. Also, the training and testing time is moderately low for different biometrics. The present multi-classifier based on different biometrics is capable and faster than other conventional uni-modal identification systems.

References

1. Bana, S., Kaur, D.: Fingerprint recognition using image segmentation. IJAEST **5**(1), 012–023 (2011)
2. Chatterjee, A., Mandal, S., Rahaman, G.M.A., Arif, A.S.M.: Fingerprint identification and verification system by minutiae extraction using artificial neural network. JCIT **1**(1), 12–16 (2010)
3. Luo, J., Lin, S., Ni, J., Lei, M.: An improved fingerprint recognition algorithm using EBFNN. In: Second International Conference on Genetic and Evolutionary Computing, pp. 504–507 (2008)
4. Yu, C., Jian, Z., Bo, Y., Deyun, C.: A novel principal component analysis neural network algorithm for fingerprint recognition in online examination system. In: 2009 Asia-Pacific Conference on Information Processing, pp. 182–186 (2009)
5. Anton, C., Ştirbu, C., VasileBadea, R.: Identify handwriting individually using feed forward neural networks. Int. J. Intell. Comput. Res. **1**(4), 183–188 (2010)
6. He, Z., Fang, B., Du, J., Tang, Y.Y., You, X.: A novel method for off-line handwriting-based writer identification. In: Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05) (2005)
7. Ashok, J., Rajan, E.G.: Off-line hand written character recognition using radial basis function. Int. J. Adv. Networking Appl. **2**(2), 792–795 (2011)
8. Kundu, S., Sarker, G.: An efficient multiple classifier based on fast RBFN for biometric identification. In: 2nd International Conference on Advanced Computing, Networking, and Informatics—Volume 1, Smart Innovation, Systems and Technologies, vol. 27, pp. 473–482 (2014)
9. Kundu, S., Sarker, G.: A modified radial basis function network for occluded fingerprint identification and localization. IJCITAE **7**(2), 103–109 (2013)
10. Sarker, G., Kundu, S.: A modified radial basis function network for fingerprint identification and localization. In: International Conference on Advanced Engineering and Technology, pp. 26–31 (2013)
11. Kundu, S., Sarker, G.: A modified RBFN based on heuristic based clustering for location invariant fingerprint recognition and localization with and without occlusion. In: IEEE's International Conferences for Convergence of Technology, pp. 1–6 (2014)
12. Kundu, S., Sarker, G.: A modified BP network using Malsburg learning for rotation and location invariant fingerprint recognition and localization with and without occlusion. In: Seventh International Conference on Contemporary Computing, pp. 617–623 (2014)
13. Sarker, G.: An unsupervised natural clustering with optimal conceptual affinity. J. Intell. Syst. **19**(3), 289–300 (2010)
14. Sarker, G.: A heuristic based hybrid clustering. JIE Comput. Eng. Div. **89**, 7–10 (2008)
15. Sarker, G.: An unsupervised learning network for face identification and subsequent localization. In: International Conference on Communications, Devices and Intelligent Systems, pp. 652–655 (2012)

Improving Personalized Recommendations Through Overlapping Community Detection Using Multi-view Ant Clustering and Association Rule Mining

Thenmozhi and Ezhilarasi

Abstract Recommender system is a technique to generate meaningful personalized recommendations, suggestions for particular customers. Due to the huge amount of data on the users and their item preferences, the existing recommendation approaches are time-consuming, and they face many performance issues during data processing. Hence, clustering users into overlapping communities will help with the data sparsity problem and enhance recommendation diversity. Another important factor in recommendation system is dynamic, user interest in which the user interest changes over time. Hence, this paper focuses on to develop a multi-view clustering approach using ant clustering method for community detection. To improve the quality of the recommendation, the overlapping communities are further classified based on temporal factors. Finally, for predicting user interest from the communities' adaptive association rule, mining has been applied.

Keywords Recommender system • Ant clustering method • Temporal overlapping communities • Personalized recommendation • Adoptive association rules

1 Introduction

Recommendation system (RS) which involves in predicting the user interest has been widely adapted in many applications. The biggest challenge in recommendation system is to identify the actual interest of the user. In the recommendation process, the collected information's are filtered and sorted for a particular user to

Thenmozhi (✉) · Ezhilarasi
Department of Computer Science and Engineering, Pondicherry Engineering College,
Puducherry 605014, India
e-mail: thenmozhi@pec.edu

Ezhilarasi
e-mail: ezhil.9944@pec.edu

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_33

provide personalized recommendations. To ease the process of recommendation, communities of users are formed and opinions are collected from the community, which helps to identify the interest of the user [1]. A number of recommender systems are used in various domains on the internet, and each one of them tries to predict the user preference accurately. In addition, various researchers have proposed techniques to improve the interest prediction in recommender systems.

However, the accurate interest predictions of an individual user is a difficult process and insufficient to develop the best recommendation systems. Among different algorithms that are used to build a recommender system, collaborative filtering methods have been used in many commercial applications [2]. Although collaborative filtering gained popularity, it also has difficulties in searching similar users when facing large data set. Hence, it is necessary to develop model-based methods for the recommendation. In the case of social media sites, the great challenge in providing recommendation services is predicting the dynamic user interest [3]. Static recommender will not capture the user's interest that is constantly changing over time. Users may belong to multiple interest communities, and their interest is highly dynamic. Among various model-based methods, the most popular method is matrix factorization which is capable of adopting a large data set. However, a critical drawback in this method is that it cannot quickly retain the model with newly issued ratings, and moreover, it is time-consuming and costly. A few authors have adapted clustering based recommender system [4], but still it suffers from low accuracy and coverage.

In this paper, we propose a recommendation method which generates communities of users by multi-view ant colony clustering approach in which the cluster is based on the similar ratings and social trust relationship. Different set of communities is formed based on the overlapping community detection technique, and it is further partitioned based on the time frame. To predict user interest, the association rule is then generated for certain community. In Sect. 2, we discuss about the literature survey of existing work and the problem which is present in the existing system. In Sect. 3, the proposed system architecture and different steps involved have been provided. In Sect. 4, evaluation of the proposed work has been provided. The conclusion and the future work have been presented in Sect. 5.

2 Literature Review

2.1 Cluster-Based Community Detection

Clustering techniques are used for identifying groups of users who have similar preferences. Here, they take the average of the user opinions in the cluster to make personalized recommendations. Sarwar et al. proposed a scalable neighborhood using clustering with bisecting k-means algorithm in real-time applications and improved the quality of recommendations [5]. Pujol et al. emphasized enhancing algorithm to form community using spectral analysis and modularity optimization in complex networks. [6]. Brian Lee and Yung Rowe use the social network graph theory analysis to predict user interest for community detection. It creates more noisy data and faces complexity in predicting user biggest interest [7]. Ding et al. developed a method called community detection algorithm based on topology potential, but it only focuses on the user similarity [8]. Fatemi et al. proposed community-based social recommender system (CBSRS) method, but it failed to focus on the overall quality of the communities formed [9]. An efficient form of multi-view ant clustering algorithm with inspired from the behavior of real ants, which works efficiently with more fuzzy, and complex data has been proposed in this paper to handle the issues exists in community formation in RS.

2.2 Overlapping Community Detection

However, many real networks have more complex and fuzzy overlapping community structures, in which two communities have more similar nodes that lead to data and time complexity. For overlapping community detection models, Sun et al. have proposed a fuzzy-based relation operation clustering method to detect the overlapping communities in complex networks [10]. Lancichinetti and Fortunato proved that the modularity function did not fit the overlapping community context very well, and it had a problem with resolution limit and degradation under overlapping communities [11]. Feng et al. proposed a temporal overlapping community detection method that can handle the pervasively overlapping community problem [12]. In this paper, we adopted the overlapping community detection method proposed by Feng et al. to improve the efficiency and accuracy in overlapping community formation.

2.3 Association Rule

In static RS, association rules have been successfully used to represent user interest. For dynamic interest models, Sharma et al. proposed an optimized temporal weighted association rule miner for generating rule for seasonal items, in which the data items are partitioned according to the seasons. This method faces complexity with the fuzzy data set [13]. Lee et al. adapted apriori algorithm in association rules mining to detect similar user object data access [14].

3 Multi-view Ant Cluster Method with Adoptive Association Rule Mining

In this section, we proposed an approach called multi-view clustering approach using ant clustering method for community detection. To improve the quality of personalized recommendation to the users, rules are generated from the communities through adaptive association rule mining. In Fig. 1, the movies lens data set is given as the input and the users are clustered based on two views. These two kinds of clusters are integrated based on the more overlapping nodes between them. After community formation, the members of the community are divided based on the time they rated the movies. For each community, frequent item set mining is calculated, and rules for each community are generated based on frequent item set. Finally, the rules for each user are refined from the community to obtain the top—N recommended items/movies for each user. The following sections provide the detail description of each step.

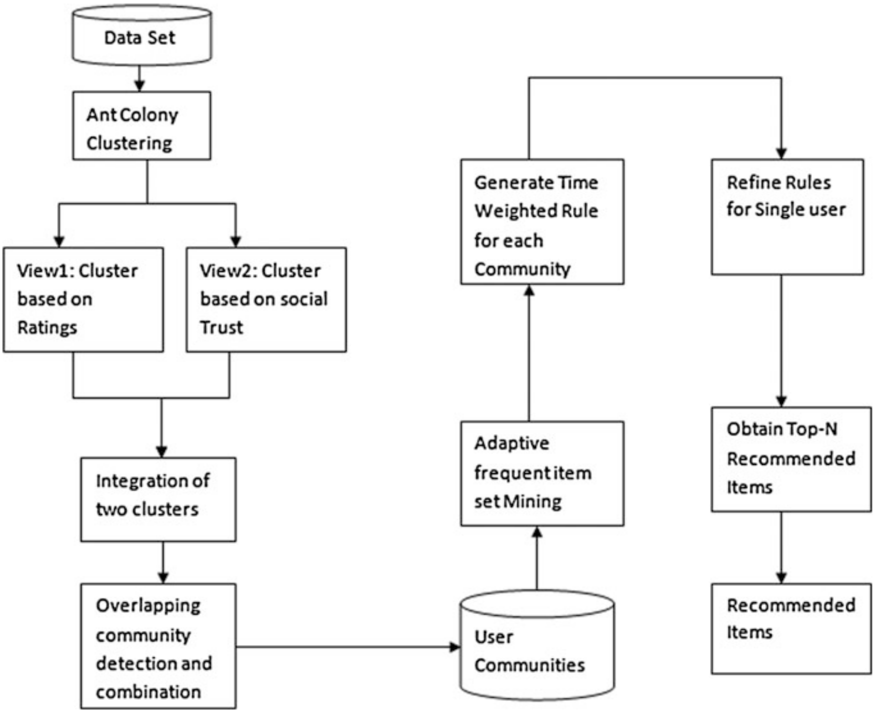

Data Set
Ant Colony Clustering
View1: Cluster based on Ratings
View2: Cluster based on social Trust
Integration of two clusters
Overlapping community detection and combination
User Communities
Adaptive frequent item set Mining
Generate Time Weighted Rule for each Community
Refine Rules for Single user
Obtain Top-N Recommended Items
Recommended Items

Fig. 1 Multi-view ant cluster method with adaptive association rule mining

3.1 Overlapping Community Generation

Community generation is the form of structuring the users into groups in social networks which improves the system functionality. We propose a multi-view ant colony-based clustering approach for community detection to improve the quality of communities formed. It incorporates two views of data cluster: one is based on user similar ratings and other is based on a trust score between users.

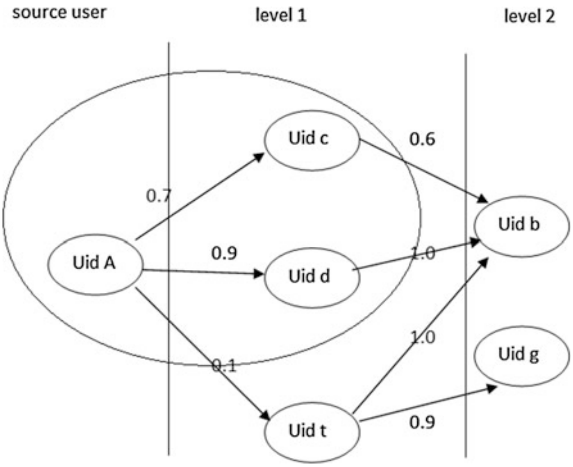
3.1.1 Multi-view Clustering

The multi-view clustering method is applied to form the user community from both the user similarity of ratings and also with the trust between the other users [15]. The clustering algorithm based on the ant colony optimization method reduces the execution time and improves the cluster effect and stability. First, the users are clustered based on similar ratings. Second, the users are clustered based on the trusted neighbors among the set of users. In Fig. 2, an example to predict the trust score of a target user [2] is given. User A trusts the user *c* and *d*, it does not accept the user t, and hence, the trust score for user *t* is 0.1, which is less than the given threshold 0.5. The trust value is defined as:

$$t_{u,v} = 1/d_{u,v}, \tag{1}$$

where $t_{u,v} \in (0,1)$ is the trustworthiness measure between user *v* and user *u*, and $d_{u,v}$ is the minimum distance between users *v* and *u*, where *u* and *v* are in social trust relationship. There are two steps in calculating the trust value. The first step involves in removing cycles in the trust network and transform into a directed graph. Second step consists of a graph walk starting from the source node to the neighbor nodes to computing the trust score of visited nodes.

Fig. 2 Trust example



The ant cluster algorithm steps are described in Table 1. The following threshold formulae are used to pick and drop the data items by the agent:

$$\mathrm{P}^{*}_{\mathrm{pick}(i)} = \left\{ 1.0 \quad \mathrm{if}\, f^{*}(i) \leq 1.0 \quad \mathrm{else} \quad 1/f^{*}(i)^{2} \right\} \tag{2}$$

$$\mathrm{P}^{*}_{\mathrm{drop}(i)} = \left\{ 1.0 \quad \mathrm{if}\, f^{*}(i) \geq 1.0 \quad \mathrm{else} \quad f^{*}(i)^{4} \right\} \tag{3}$$

where $f^{*}(i)^{n}$ is a Lumer and Faieta's neighborhood function which is used for picking and dropping the data items.

Finally, ant cluster algorithm gives the set of clustered communities.

3.1.2 Overlapping Community Detection and Combination

If two communities have too many overlapping nodes, then it should be merged into a single community. Figure 3 describes the overlapping community detection, where C1 is similarity based cluster and C2 is trust-based cluster. Dashed lines indicate the similarity where solid lines represent user (u) trust.

Table 2 explains the overlapping community combination steps. The overlapping communities are of different size; hence, it is necessary to find the overlapping proposition measure. If the overlapping communities are of same size, then the following formula is used to find the overlapping proposition measure:

$$\delta_{pq} = \beta * \left|c_{p} \cap c_{q}\right| / \left|c_{p} \cup c_{q}\right| + (1-\beta) * \left|Nc_{p} \ \cap \ Nc_{q}\right| / \left|Nc_{p} \ \cup \ Nc_{q}\right|, \tag{4}$$

where C_p and C_q are the pth and qth overlapping communities, NC_p and NC_q are the set of neighbor nodes that directly connect with the nodes in C_p and C_q. For combining the overlapping communities, a combination threshold $\rho 1 \in [0, 1]$ is predefined. If overlapping proposition measure is greater than the threshold, then the two communities are combined. When the size of one community is much

Table 1 Ant clustering algorithm

1. Scatter all data items randomly in a grid
2. Randomly select the *k* number of agents
3. Select one agent in a random manner
4. The agent will pick (2) a data item and moves it in a random direction
5. The agent drop (3) the data item it holds in the current position or in the immediate neighborhood position
6. After dropping the data item it searches the next data item to pick up (2)
7. Continue step 5 until all data items are met once

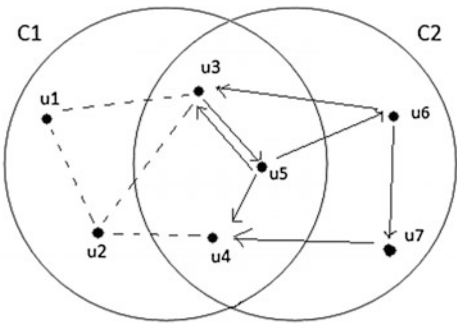


Fig. 3 Overlapping community detection

smaller than the other community, then the overlapping proportion measure is calculated using the following formula:

$$\delta_{pq} = \left|c_p \cap c_q\right| / \min\left(\left|c_p\right| \cup \left|c_q\right|\right). \tag{5}$$

For combining the overlapping communities, a combination threshold $\rho2 \in [0, 1]$ is predefined. If $\delta_{pq} > \rho2$, then the two communities are combined. We first determine a threshold ζ value for the community scale ratio to the smaller community size from the larger community size:

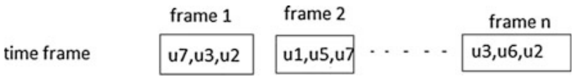
$$\zeta = (\rho2 + \rho2 * \rho1 - \rho1)/\rho1. \tag{6}$$

Then, the community combination process is processed, as shown in Table 2.

Table 2 Overlapping community combination

Step 1: The ratio of the two community sizes is calculated as $r_{pq} = \max(\lvert c_p\rvert, \lvert c_q\rvert)/\max(\lvert c_p\rvert, \lvert c_q\rvert)$
Step 2: If r_{pq} <, which means that there is no significant difference in size between the two communities, calculate the overlapping proportion measure using Eq. (4). Otherwise, go to step 4
Step 3: If $\delta_{pq} > \rho1$, the two communities should be combined; otherwise, there is no combination operation executed. Go to step 6
Step 4: If $r_{pq} > \zeta$ which means that the size of one community is much smaller than the other, Eq. (5) is used to calculate the overlapping proportion measure
Step 5: If, $\delta_{pq} > \rho2$, the two communities should be combined; otherwise, there is no combination operation executed. Go to step 6
Step 6: Output the community combination result

Fig. 4 Time frame for communities



3.1.3 Partition User Based on Time

Based on the characteristics of the community the user ratings are divided into time frames with respect to the different time periods. Based on the time frame, the recent interest of the user is predicted easily and accurate recommendation is provided. The purpose of time frame partition within the community is to improve the effectiveness and quality of the recommendation system. In Fig. 4 users u7, u3, and u2 belong to the time frame 1, in which these users rated the items or movies on the same time.

3.2 Adaptive Association Rule Mining

Conducting association rule mining on the each overlapping communities gives more desired rules for recommending items to the users. The minimum support value for each community should be adjusted based on the characteristics of a community which gives more accurate rules for predicting the desired interest of the user in a community.

3.2.1 Adaptive Frequent Item Set Mining

The frequent pattern growth (FP Growth) algorithm has been used to accelerate the search process, and also it provides a parallelization scheme. The important factor in mining the frequent item set is the minimum support value which decides the overall performance of the algorithms in the search process. This algorithm works as follows:

1. Find the support for each item in the data set.
2. Eliminate the items which are infrequent.
3. Based on the support value, the frequent items are sorted in decreasing order.

3.2.2 Association Rule Mining

We apply association rule mining methods based on the FP Growth model. Association rules are generated based on the frequent patterns to identify the most important relationships using the measures such as support and confidence. Association rules mining procedure in a certain community steps:

Input: frequent item sets Fset_k and data set D_k for the cat community.

1. For each item set in the frequent item set that forms a rule A → B.
2. Count the number of items A appearances in the dataset and count the number of {A, B} appearance on frequent item set.
3. The rule A → B is accepted only when it satisfies the confidence value else the rule rejected.

3.3 *Personalized Recommendation*

The proposed recommendation model, the membership value of a user and the adaptive associated rules in every community is considered to provide personalized recommendations. The recommendation value for a user is detected based on a ranking of all recommendations from the number of communities the user present. The ratio of all the user's link weights in this community to the total link weights of the user in all his communities gives the membership value of a user in a certain community. The membership of *i*th user in the *k*th community is calculated as follows:

$$\mathrm{memb}_k^i = \sum\nolimits_{\mathrm{i}=1}^{|c_k|} U_{il} / \sum\nolimits_{C_{j \in \{1\ldots\mathrm{cp}\}}} \sum\nolimits_{\mathrm{i}=1}^{|c_j|} l, \tag{7}$$

where C_1 to C_p are all communities that contain the *i*th user. $U_{i\mathrm{l}}$ calculate the degree of interest similarity between *i*th user and the *l*th user. Finally, the top-N recommendation list of items is extracted from the communities the user exists and recommended to the user.

4 Evaluation

In the proposed recommendation system, we use the movie lens which consists of the percentage of movies in different categories rated by a user over 8 months. This data set clearly shows that user interest is dynamic and change over time. The proposed recommendation system is under development; hence, results providing a comparative analysis are our immediate future work.

To evaluate the proposed recommendation strategy, we use three evaluation metrics, such as precision, recall, and F-measure. Here, the data can be divided into four parts, namely, the true positive (TP) which denotes what the user actually likes and other items are considered as false positive (FP). The items that are actually liked by the user but are not recommended to the user are considered as false negative (FN), and other items are denoted as true negative (TN).

The top N recommendation list contains three parts: TP, FP, and ignored recommendations (IR) and total recommendation Number (N).

$$N = TP + FP + IR \tag{9}$$

Generally, precision and recall are diverging properties, in which increase in recall usually results in a decrease in the value of precision. The mean value of precision and recall is called as F1 Measure. As the proposed work is under development, the evaluation of the system recommendation is our immediate future work.

5 Conclusion and Future Work

In this paper, we proposed a new recommendation system with the multi view ant cluster algorithm with temporal community detection and adaptive association rule mining for the generation rules for recommendation. This approach is used to capture the dynamic user interest for better recommendation to the user. The movie lens data set is used for evaluating this approach. This recommendation system is being developed for the evolution of the system with the existing methods. We can extend this study in number of directions. To predict user interest, the other parameters, such as their age, reviews, dislikes, etc., which also contain some important information that can be considered for accurate predictions.

References

1. Hendler, J.: FilmTrust: Movie Recommendations Using Trust in Web-based Social Networks, Conference Paper (2006). doi:10.1109/CCNC.2006.1593032. Source: IEEE Xplore
2. McDonald, D.W., Ackerman, M.S.: Expertise recommender: a flexible recommendation system and architecture. In: CSCW'00 Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work
3. Statho, V., Poulos, I.K., Jose, J.M.: On social networks and collaborative recommendation. In: SIGIR'09 Proceedings of the 32nd International ACM SIGIR Conference, USA ©2009
4. Xue, G.-R., Yang, Q.: Scalable collaborative filtering using cluster-based smoothing. In: SIGIR '05 Proceedings of the 28th Annual International ACM SIGIR Conference, USA ©2005
5. Sarwar, BM, Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large scale e-commerce: scalable neighborhood formation using clustering. GroupLens Research Group/Army HPC Research Center Department of Computer Science and Engineering University of Minnesota, Minneapolis, MN 55455, USA, 2004
6. Pujol, J.M., Béjar, J., Delgado, J.: Clustering algorithm for determining community structure in large networks. Phys. Rev. E **74**, 016107 (2006)
7. Rowe, B.L.Y., Novo, Z.: Using Social Network Graph Analysis for Interest Detection. Data Analytics, School of Professional Studies, CUNY (2014)

8. Ding, X., Wang*, Z., Chen, S., Huang, Y.: Community-based Collaborative Filtering Recommendation Algorithm, College of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China
9. Fatemi, M., Tokarchuk, L.: A Community Based Social Recommender System for Individuals and Groups, School of Electronic Engineering and Computer Science. IEEE International Conference on Privacy, Security, Risk and Trust (2010)
10. Sun, P.G., Gao, L., Han, S.S.: Identification of Overlapping and Non-overlapping Community Structure by Fuzzy Clustering in Complex Networks, vol. 181, Issue 6. Elsevier Science Inc. New York (2011)
11. Lancichinetti, A., Fortunato, S.: Limits of modularity maximization in community detection. Phys. Rev. E **84**(6), 066122 (2011)
12. Feng, H.a, Tian, J.a*, Wang, H.J.b, Li, M.a: Personalized recommendations based on time-weighted overlapping community detection, Tianjin University, Tianjin 300072, PR China b University of Delaware, Newark, DE 19716, USA
13. Sharma, P., Saxena, K.: Temporal weighted association rule mining for classification. Int. J. Comput. Theory Eng. **4**(5) (2012)
14. Leea, C.-H., Kim, Y.-H., Rhee, P.-K.: Web personalization expert with combining collaborative filtering and association rule mining technique. Expert Syst. Appl. **21**(3), 131–137 (2001)
15. Guo, G., Zhang, J., Yorke-Smith, N.: Leveraging Multiviews of Trust and Similarity to Enhance Clustering-based Recommender Systems. Elsevier, Knowledge-Based Systems (2014)
16. Mase, H., Kanamori, K., Ohwada, H.: Trust-aware recommender system incorporating review contents. Int. J. Mach. Learn. Comput. **4**(2) (2014)
17. Boryczka, U.: Ant Clustering Algorithm, Intelligent Information Systems 2008, ISBN 978-83-60434-44-4, pp. 377–386

Part VII
Other Applications of Computational Intelligence

Vehicle Vibration and Passengers Comfort

Syeda Darakhshan Jabeen

Abstract To accelerate the level of passengers comfort, the author presents in this paper a heuristic approach to model the vehicle suspension parameters. The goal in this work is to design suspension parameters, in such a way that the passengers comfort is not decorated by forces induced by the road bump(s) or irregular rough surface. Here, the half car linear passive suspension model with passengers has been considered. Initially, model simulation over different roads is done. Later, by varying the mass of the passengers within a given range the variation and range of the vibration levels of the dynamical system have been analyzed. Then, the vibration level of the system is minimized using genetic algorithm and the suspension parameters are found out. This is done by keeping the masses of the passengers constant. With those parameters, again the vibration analysis is done with changing passenger's masses in a suitable interval. The tabulated results and graphical representations show that the suspension parameters found can serve as an efficient value to enhance comfort.

Keywords Road bump • Suspension parameters • Vibration • Genetic algorithm and optimization

1 Introduction

Ride comfort is influenced by disturbances, especially vibrations induced from the tire-ground contact and are mainly effected by the spring stiffness and the damping properties of the suspension [1]. Thus, designing a good suspension has now become a prevailing philosophy in the automobile industries. To meet these demands, many types of suspension systems ranging from passive, semi-active to active suspensions have been proposed [2–7]. Suspension parameters are often

S.D. Jabeen (✉)
Department of Mathematics, Birla Institute of Technology, Mesra,
Ranchi 835215, India
e-mail: jabeen@bitmesra.ac.in; syeda@iitk.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_34

found out by examining the behavior of the mathematical model under various conditions. The solutions to these mathematical models can be found analytically or numerically. To obtain the optimal values of suspension parameters, different types of control/optimization algorithms are cited in the literature [4–14]. Those include gradient-based Dynamic-Q algorithm, optimal control theory, fuzzy logic and neural network control, sliding mode and adaptive control, H∞ control and stochastic methods. Recently, genetic algorithm (GA) has attracted researchers due to its robustness and its ability in solving various vibration control problems [2, 3, 15–18]. The advantage of the algorithm over other methods is that it does not require gradient information but rather performs stochastic search to optimize the function. Many researchers have expressed their interest in suspension design problems giving priority to different comfort factors. Therefore, variance of the dynamic load [15], bouncing transmissibility of sprung mass [18, 19], sprung mass vertical and pitch accelerations [20], sprung mass vertical acceleration, angular motion and vertical displacement of unsprung masses [17] resulting from the vibrating vehicle were minimized to enhance comfort.

In this paper, we propose another alternative approach for designing passive suspension parameters of the half car model with passengers. At first model simulation has been done over the roads using industrial data to analyze the levels of vibration of the vehicle and passengers. The mass of the passenger is later incremented each time to identify the variation and the range of vibration with the change in passengers mass. The vibration of the system is then minimized by GA keeping passengers mass fixed and simultaneously the values of the suspension parameters are determined. The formulated optimization problem minimizes the weighted sum of the bouncing transmissibility of the sprung mass and passengers jerk. Furthermore, with the optimal suspension parameters, the variation and range of vibration of the model are found by changing passenger's mass. The envelope of the vibration levels and the range within which the vibration measuring parameters resides are then compared and illustrated for both NS and GA. The experimental results and graphical representation conclude that the optimal suspension parameters can serve as an efficient suspension design.

2 Dynamical Model of a Vehicle

Figure 1 shows a half car passenger model [18] with slight modification. Here, the suspension, tire and passengers seat are assumed to have a linear springs in parallel to viscous dampers.

The model consists of a sprung mass, two unsprung masses and two passengers. The sprung mass motions have bounce and pitch with every unsprung mass having its own bounce. The passengers are considered to have only vertical oscillations. In the figure, the variables q_2 and q_3 resent the vertical displacement and angular motion of the sprung mass; q_1 and q_4 the vertical displacements of the unsprung masses; q_5 and q_6 the vertical displacements of the passengers. The other

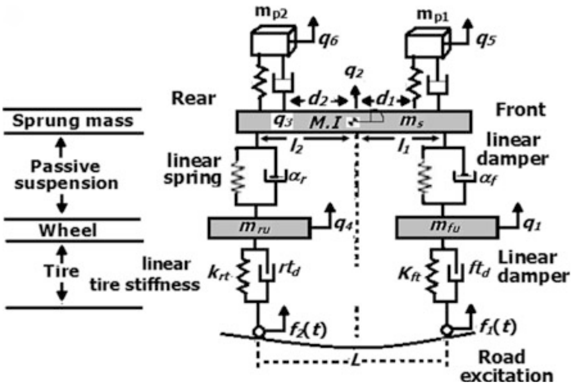


Fig. 1 A half car model

parameters in the figure remain the same as in Jabeen [18]. The equations of motion governing the dynamical system are obtained by the Newton's second law of motions and are given below:

Equations of motion of the front and the rear unsprung masses are given by

$$
\begin{aligned}
m_{fu}\ddot{q}_1(t) &= k_{fs}(q_2(t)+l_1q_3(t)-q_1(t))+\alpha_f(\dot{q}_2(t)+l_1\dot{q}_3(t)-\dot{q}_1(t))-k_{ft}(q_1(t)-f_1(t))\\
m_{ru}\ddot{q}_4(t) &= k_{rs}(q_2(t)-l_2q_3(t)-q_4(t))+\alpha_r(\dot{q}_2(t)-l_2\dot{q}_3(t)-\dot{q}_4(t))-k_{rt}(q_4(t)-f_4(t))
\end{aligned}
$$

The vertical and angular motions of the sprung masses are

$$
\begin{aligned}
m_s\ddot{q}_2(t) = &-c_{p1}(\dot{q}_5(t)-\dot{q}_2(t)-d_1\dot{q}_3(t))+k_{p1}(q_5(t)-q_2(t)-d_1q_3(t))+c_{p2}(\dot{q}_6(t)-\dot{q}_2(t)+d_2\dot{q}_3(t))+k_{p2}(q_6(t)-q_2(t)+d_2q_3(t))\\
&-\alpha_f(\dot{q}_2(t)+l_1\dot{q}_3(t)-\dot{q}_1(t))-k_{fs}(q_2(t)+l_1q_3(t)-q_1(t))-\alpha_r(\dot{q}_2(t)-l_2\dot{q}_3(t)-\dot{q}_4(t))-k_{rs}(q_2(t)-l_2q_3(t)-q_4(t))\\
J\ddot{q}_3(t) = &\{c_{p1}(\dot{q}_5(t)-\dot{q}_2(t)-d_1\dot{q}_3(t))+k_{p1}(q_5(t)-q_2(t)-d_1q_3(t))\}d_1-\{c_{p2}(\dot{q}_6(t)-\dot{q}_2(t)+d_2\dot{q}_3(t))+k_{p2}\\
&d_2(q_6(t)-q_2(t)+d_2q_3(t))\}-\{\alpha_f(\dot{q}_2(t)+l_1\dot{q}_3(t)-\dot{q}_1(t))+k_{fs}(q_2(t)+l_1q_3(t)-q_1(t))\}l_1+\\
&\{\alpha_r(\dot{q}_2(t)-l_2\dot{q}_3(t)-\dot{q}_4(t))+k_{rs}(q_2(t)-l_2q_3(t)-q_4(t))\}l_2
\end{aligned}
$$

Equations of motion of the front and the rear passengers are given by

$$
\begin{aligned}
m_{p1}\ddot{q}_5(t) &= -c_{p1}(\dot{q}_5(t)-\dot{q}_2(t)-d_1\dot{q}_3(t))-k_{p1}(q_5(t)-q_2(t)-d_1q_3(t))\\
m_{p2}\ddot{q}_6(t) &= -c_{p2}(\dot{q}_6(t)-\dot{q}_2(t)+d_2\dot{q}_3(t))-k_{p2}(q_6(t)-q_2(t)+d_2q_3(t))
\end{aligned}
$$

This is a system of linear second-order coupled simultaneous equations, where single and double dots represent first- and second-order derivatives with respect to time. Letting $z_i = \dot{q}_i$, i $= 1$ to 6, the above system of equations can be reduced to a system of linear first-order simultaneous Eq. (1). These equations can be solved in the time domain by any numerical methods. The above system of equations can now be expressed as:

$$\dot{X}(t) = AX(t) + B(t) \tag{1}$$

where,

$$\dot{X}(t)=[\dot{z}_1(t),\dot{z}_2(t),\dot{z}_3(t),\dot{z}_4(t),\dot{z}_5(t),\dot{z}_6(t),\dot{q}_1(t),\dot{q}_2(t),\dot{q}_3(t),\dot{q}_4(t),\dot{q}_5(t),\dot{q}_6(t)]^T,$$
$$X(t)=[z_1(t),z_2(t),z_3(t),z_4(t),z_5(t),z_5(t),q_1(t),q_2(t),q_3(t),q_4(t),q_5(t),q_6(t)]^T,$$

$$A=\begin{bmatrix}
\frac{-\alpha_f}{m_{fu}} & \frac{\alpha_f}{m_{fu}} & \frac{\alpha_f l_1}{m_{fu}} & 0 & 0 & 0 & \frac{-(k_{fs}+k_{ft})}{m_{fu}} & \frac{k_{fs}}{m_{fu}} & \frac{k_{fs}l_1}{m_{fu}} & 0 & 0 & 0\\
\frac{\alpha_f}{m_s} & a22 & a23 & \frac{\alpha_r}{m_s} & \frac{c_{p1}}{m_s} & \frac{c_{p2}}{m_s} & \frac{k_{fs}}{m_s} & a28 & a29 & \frac{k_{rs}}{m_s} & \frac{k_{p1}}{m_s} & \frac{k_{p2}}{m_s}\\
\frac{\alpha_f l_1}{J} & a32 & a33 & \frac{-\alpha_r l_2}{J} & \frac{c_{p1}d_1}{J} & \frac{-c_{p2}d_2}{J} & \frac{-k_{fs}l_1}{J} & a38 & a39 & \frac{-k_{rs}l_2}{J} & \frac{k_{p1}d_1}{J} & \frac{k_{p2}d_2}{J}\\
0 & \frac{\alpha_r}{m_{ru}} & \frac{-\alpha_r l_2}{m_{ru}} & \frac{-\alpha_r}{m_{ru}} & 0 & 0 & 0 & \frac{k_{rs}}{m_{ru}} & \frac{-k_{rs}l_2}{m_{ru}} & \frac{-(k_{rs}+k_{rt})}{m_{ru}} & 0 & 0\\
0 & \frac{c_{p1}}{m_{p1}} & \frac{c_{p1}d_1}{m_{p1}} & 0 & \frac{-c_{p1}}{m_{p1}} & 0 & 0 & \frac{k_{p1}}{m_{p1}} & \frac{k_{p1}d_1}{m_{p1}} & 0 & \frac{-k_{p1}}{m_{p1}} & 0\\
0 & \frac{c_{p2}}{m_{p2}} & \frac{-c_{p2}d_2}{m_{p2}} & 0 & 0 & \frac{-c_{p2}}{m_{p2}} & 0 & \frac{k_{p2}}{m_{p2}} & \frac{-k_{p2}d_2}{m_{p2}} & 0 & 0 & \frac{-k_{p2}}{m_{p2}}\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

where

$$a22=\frac{-(c_{p1}+c_{p2}+\alpha_f+\alpha_r)}{m_s},\ a23=\frac{-(c_{p1}d_1-c_{p2}d_2+\alpha_f l_1-\alpha_r l_2)}{m_s},\ a28=\frac{-(k_{p1}+k_{p2}+k_{fs}+k_{rs})}{m_s}$$
$$a29=\frac{-(k_{p1}d_1-k_{p2}d_2+k_{fs}l_1-k_{rs}l_2)}{m_s},\ a32=\frac{-(c_{p1}d_1-c_{p2}d_2+\alpha_f l_1-\alpha_r l_2)}{J},\ a33=\frac{-(c_{p1}d_1^2+c_{p2}d_2^2+\alpha_f l_1^2+\alpha_r l_2^2)}{J},$$
$$a38=\frac{-(k_{p1}d_1+k_{p2}d_2-k_{fs}l_1-k_{rs}l_2)}{J},\ a39=\frac{-(k_{p1}d_1^2-k_{p2}d_2^2-k_{fs}l_1^2+k_{rs}l_2^2)}{J}$$

and $B(t)=[k_{ft}f_1(t)\quad 0\quad 0\quad k_{rt}f_2(t)\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0]^T$,

Functions $f_1(t)$ and $f_2(t)$ are road excitations at time t at the front and rear tires, respectively. For road excitation different types of road, one with bumps and other with small ups and downs (random surface) have been considered. During car motion, the front wheel will first come into contact with road with bump and the rear wheel will face the same little later. The time will depend on the axel distance L and the velocity (v) of the car. Thus, the function $f_2(t)$ can be written as $f_2(t)=f_1\left(t-\frac{L}{v}\right)$.

3 Description of Road

Vehicle vibration not only depends on the suspension design, but also on the nature of vibration excitations. Human response to vibration is a function of several factors, such as vibration magnitude, frequency, character (rotational, linear) and duration of vibration originating from road conditions. Therefore, to access vibration transmitted to the passengers we have considered different types of excitations, i.e., deterministic and probabilistic/random excitations, where deterministic

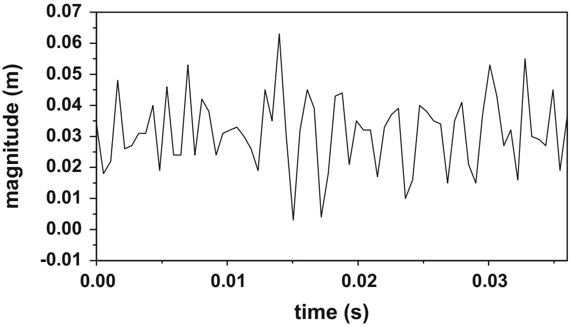


Fig. 2 Random excitation

excitation is periodic in nature. For periodic excitations, triangular and square wave functions have been used. Probabilistic excitations, which are in fact real road excitations, are generated from Gaussian distribution function. The descriptions of these external excitations are given below.

Case-1. Deterministic excitations

(a) **Triangular wave bump**

This bump is defined by the function $f(t)$ as follows:

$$f(t) = \begin{cases} \frac{2c}{T_1} t, \ (q-1)T_1 \le t \le \frac{(2q-1)}{2} T_1 \\ \frac{2c}{T_1}(T_1 - t), \ \frac{(2q-1)}{2} T_1 \le t \le qT_1 \quad q = 1 \end{cases}$$

(b) **Square wave**

This bump is defined by the function $f(t)$ as follows:

$$f(t) = \begin{cases} c, \ (q-1)T_1 \le t \le \frac{(2q-1)}{2} T_1 \\ 0, \ \frac{(2q-1)}{2} T_1 \le t \le qT_1 \quad q = 1 \end{cases}$$

Case-2. Probabilistic road

This type of road is artificially generated random numbers from Gaussian distribution function (Fig. 2).

4 Background of the Genetic Algorithm (GA)

A GA is an intelligent stochastic search algorithm that simulates the process of evolution by taking a population of solutions. It always seeks for a good solution by randomly generating a collection of potential solutions to the problem and then manipulating those solutions. The stochastic search is, however, improved by the application of genetic operators in each reproduction. Each solution in the population is evaluated according to some fitness measure. Fitter solutions in the population are allowed to reproduce, crossover and mute creating stronger population by replacing the unfit solutions in the populations. Fitter solutions are selected so that the offspring inherit the good attributes from the parents so that the average quality of the solution becomes better than that of the previous generation. Elitist is then applied to preserve the best-found solution in each generation. The cycle of evaluation—selection—reproduction is terminated when maximum generation is reached. Factors such as selection scheme and the choice of genetic operators play a tremendous role on the performance of the GA. The details of these operators have been discussed in Jabeen et al. 2006.

5 Optimization Problem

To enhance passengers comfort, we formulate a nonlinear constrained optimization problem. In this problem, the objective function minimizes the weighted sum of the maximum bounce of the sprung mass and the jerk of the passengers subject to some technological constraints. This objective function is, therefore, expressed as:

$$\begin{aligned}&\text{minimize} f(x),\\ &\text{where } f(x) = \left[w_1 \int_0^T \frac{|\max(q_2(t))|}{Amp} dt + w_2 \left(\int_0^T |\max(\dddot{q}_5(t) + \dddot{q}_6(t))| \right) \right] dt\end{aligned} \tag{4}$$

where *Amp* is the excited displacement amplitude imposed during car motion and $x = \{a \le k_{fs} \le b,\ \mathrm{c} \le k_{rs} \le d,\ \mathrm{e} \le \alpha_f \le f,\ \mathrm{g} \le \alpha_r \le h\}$.

The constraints have been constructed as follows:

1. To prevent the damage of the vehicle, caused by the hitting of the unsprung mass with the sprung mass, the limit on the suspension travel has been imposed. This has been expressed as:

$$|q_2(t) - q_1(t)| \le fdef_{\max} \text{ and } |q_2(t) - q_4(t)| \le rdef_{\max}$$

2. To maintain the circular shape of the tire under car load and to keep the natural frequency of the sprung mass less than the exciting frequency, it has been assumed that the spring stiffness does not exceed the tire stiffness, i.e., $k_{fs} \le k_{ft}$ and $k_{rs} \le k_{rt}$

4. $\sqrt{\frac{k_{fs}}{m_s}} \le \sqrt{\frac{k_{ft}}{m_s}}$ and $\sqrt{\frac{k_{rs}}{m_s}} \le \sqrt{\frac{k_{rt}}{m_s}}$

As the natural frequency of the sprung mass controls the deflection of the car body due to various input excitations such as road conditions, so when the natural frequency of the sprung mass is comparatively lower than the external excitation input, the bouncing transmissibility will be kept at a lower value.

6 Simulation Results and Analysis

Here, a comparative study on vibrations experienced by the dynamical system due to road excitations described in Sect. 3 has been performed in two parts. Sensitivity analysis has been carried out for passenger's mass variations to investigate the vibration behavior of both vehicle and the passenger. The output parameters which include maximum bounce and jerk of sprung mass, and maximum bounce of front and rear passengers are determined. Moreover, the graphs are plotted against the change in mass ratio (m_{p1}/m_{p2}) of the passengers.

6.1 Numerical Simulation (NS) Application

First model simulation has been done using industrial parameters (see Table 1). For this, the equations of motion (1) have been solved numerically in time domain with velocity of the car at 50 km/h. Under deterministic excitation, we consider a road with single bump of 0.5 m wide and of height 0.07 m whereas under probabilistic excitation, we consider rough road surface of length 0.5 m and height not exceeding 0.07 m. To investigate the impact of different forms of excitations on the vehicle and the passengers, model simulation has been carried out for 6.0 s by changing the mass of the passengers between 20 and 90 kg. Further, the variations of the output parameters due to the increase in mass ratio (m_{p1}/m_{p2}) of the two passengers, keeping the front passenger mass (m_{p1}) unchanged, are analyzed for both Case-1 and Case-2 road conditions separately, and are shown in Figs. 3, 4 and 5.

From Fig. 3, it is interesting to observe that if the mass of the passengers is in closed interval [20, 90 kg], the maximum bounce of sprung mass, front passenger and rear passenger and jerk of sprung mass will be 0.2578, 0.2831, 0.6226 and 92.2932, respectively, for the Case-1 (a) road condition. Similarly, for Case-1 (b) road condition, from Fig. 4, it is found that the expected maximum bounce of sprung mass, front passenger, rear passenger and jerk of the sprung mass will be 0.2572, 0.2834, 0.6211, and 105.2227, respectively.

Table 1 Comparison of output parameters under different road conditions

Not.	Description	Units	Value			
			NS	GA Case-1 (a)	GA Case-1 (b)	GA Case-2
k_{fs}	Front coil stiffness	N/m	66824.20	8029.59	4132.66	4631.27
k_{rs}	Rear coil stiffness	N/m	18615.0	2899.57	4425.94	2091.99
α_f	Front damper coefficient	Ns/m	1190.0	228.93	218.20.63	340.69
α_r	Rear damper coefficient	Ns/m	1000.0	280.72	237.89	233.63

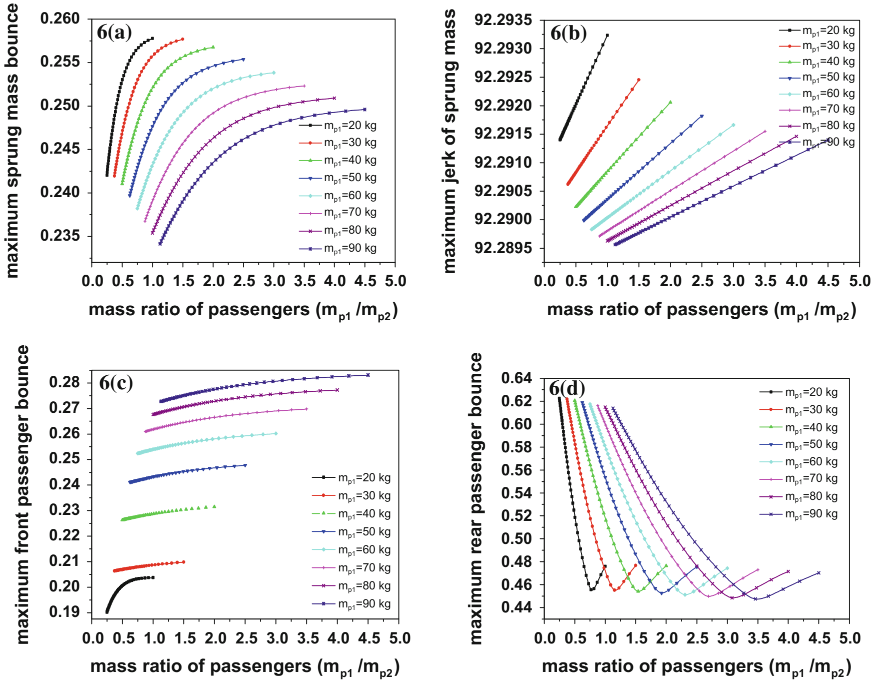


Fig. 3 Effect of the change in mass ratio of the passengers for NS under Case-1 (a) road

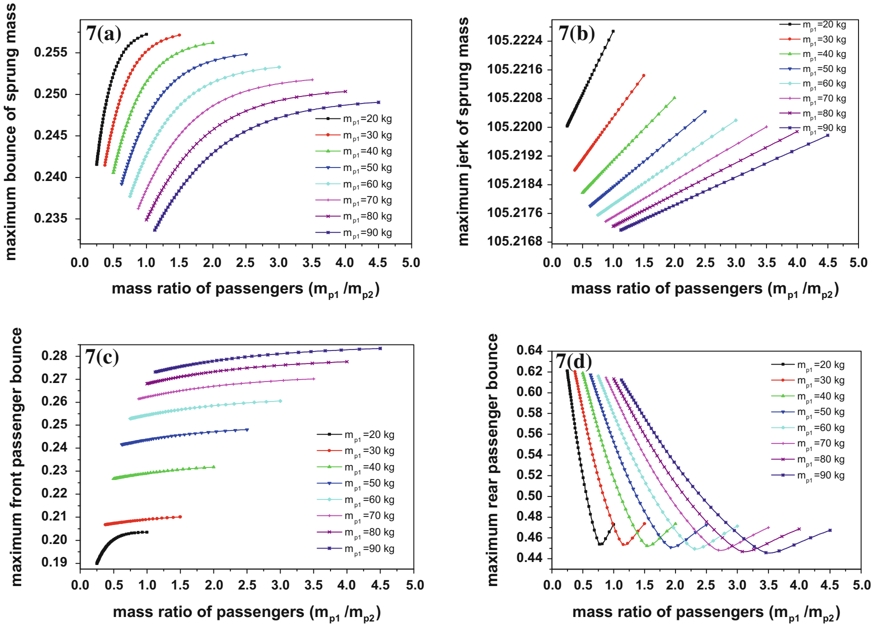


Fig. 4 Effect of the change in mass ratio of the passengers for NS under Case-1 (b) road

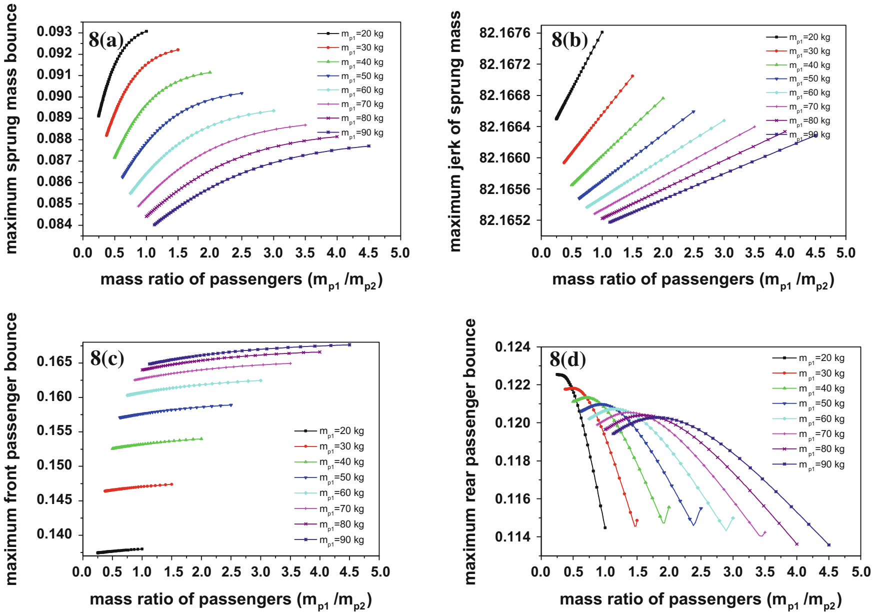


Fig. 5 Effect of the change in mass ratio of the passengers for NS under Case-2 road

It is interesting to observe from Fig. 5 that, if the mass of the passengers lies in the range [20, 90 kg] then, for Case-2 road condition, the maximum sprung mass bounce, front passenger bounce, rear passenger bounce and jerk of the sprung mass will be 0.0931, 0.1676, 0.1225 and 82.1676, respectively.

6.2 Genetic Algorithm (GA) Application

Here, GA has been applied to find better values of suspension parameters that will reduce the vibration levels to provide more comfortable ride to the passengers. The optimal values of those parameters are found by solving the optimization problem (4) with GA under technological constraints. The GA parameters that have been considered are: population size: 120, maximum number of generation: 150, crossover and mutation rates as decreasing function within the interval [0.8, 0.9] and [0.15, 0.2], tournament selection of size four and elitism of size three. The fitness of the individuals is computed after solving (1) by fourth order Runga Kutta method. The search domain comprising the parameters k_{fs}, k_{rs}, α_f and α_r is defined in [0, 68000], [0, 19000], [0, 1200] and [0, 1200], respectively. Other conditions are kept same as in NS. The front and rear deflections, $fdef_{\max}$ and $rdef_{\max}$, are limited to 0.064 m. The constrained optimization problem has been solved by converting it into an unconstraint one by penalty method. With the optimal suspension

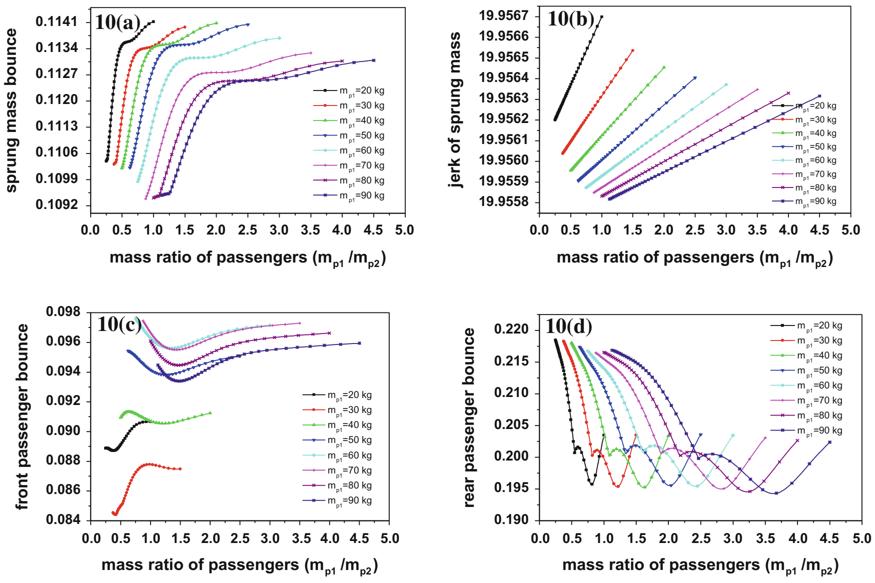


Fig. 6 Effect of the change in mass ratio of the passengers for GA parameters over Case-1 (a) road

parameters, the variations in vibration of the vehicle and the passengers are again analyzed by changing the mass of the passengers within the range [20, 90 kg]. The envelope of bounce of the sprung mass, passengers and jerk of the vehicle is then plotted against increasing mass ratios of the passengers and demonstrated in Figs. 6, 7, 8 and 9.

From Fig. 6, it is interesting to observe that in the Case-1 (a) road condition, if the mass of the passengers is within the range [20, 90 kg], the maximum sprung mass bounce, front passenger bounce, rear passenger bounce and jerk will be 0.1141, 0.0976, 0.2185 and 19.9567, respectively.

For Case-1 (b) road condition, from Fig. 7 it is found that the expected maximum bounce of sprung mass, front passenger, rear passenger and jerk of the vehicle will be 0.1305, 0.1003, 0.2650 and 19.6371 if the passenger masses are in the range [20, 90 kg].

It is interesting to observe from Fig. 8 that, if the mass of the passengers lies in the range [20, 90 kg] then, for Case-2 road condition, the maximum sprung mass bounce, front passenger bounce, rear passenger bounce and jerk of the sprung mass will be 0.0322, 0.0410, 0.0488 and 19.8520, respectively.

The results obtained after analysis carried out in NS and GA application are presented in Tables 2 and 3. Finally, with the optimal suspension parameters, the range of vibrations of the vehicle and passengers is found by varying mass of the passengers. This result is again compared with those of NS and is presented in Table 4. Output parameters obtained for vibration analysis are further compared graphically and presented in Figs. 9, 10 and 11 for Case-1 (a), Case-1 (b) and

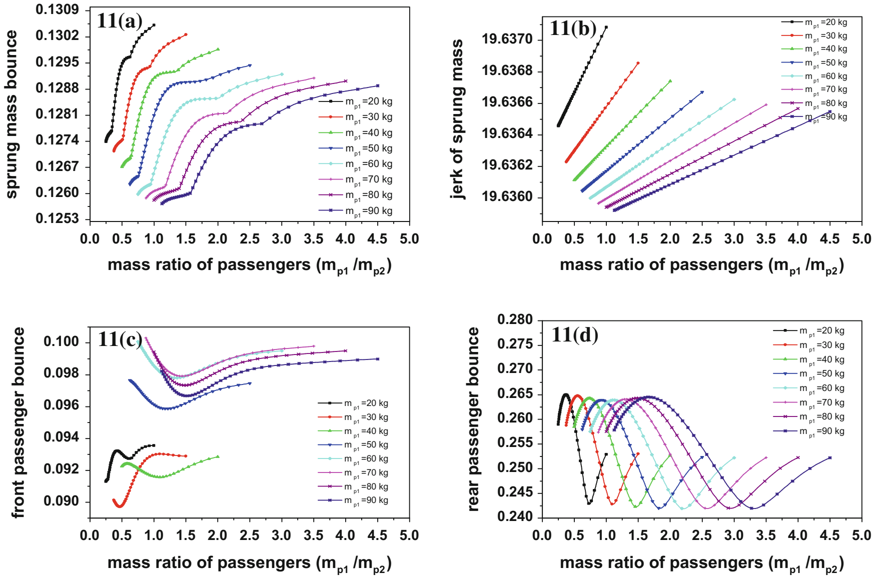


Fig. 7 Effect of the change in mass ratio of the passengers for GA parameters over Case-1 (b) road

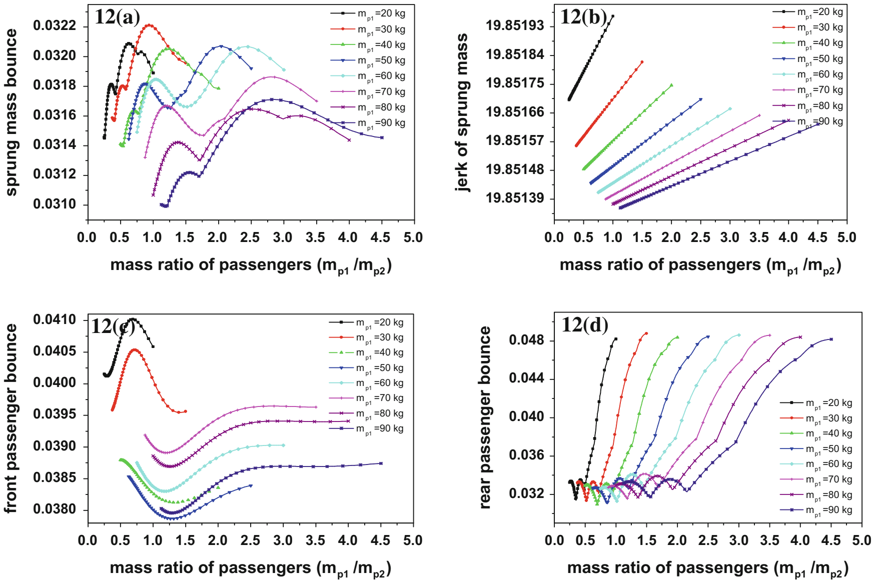


Fig. 8 Effect of the change in mass ratio of the passengers for GA under Case-2 road

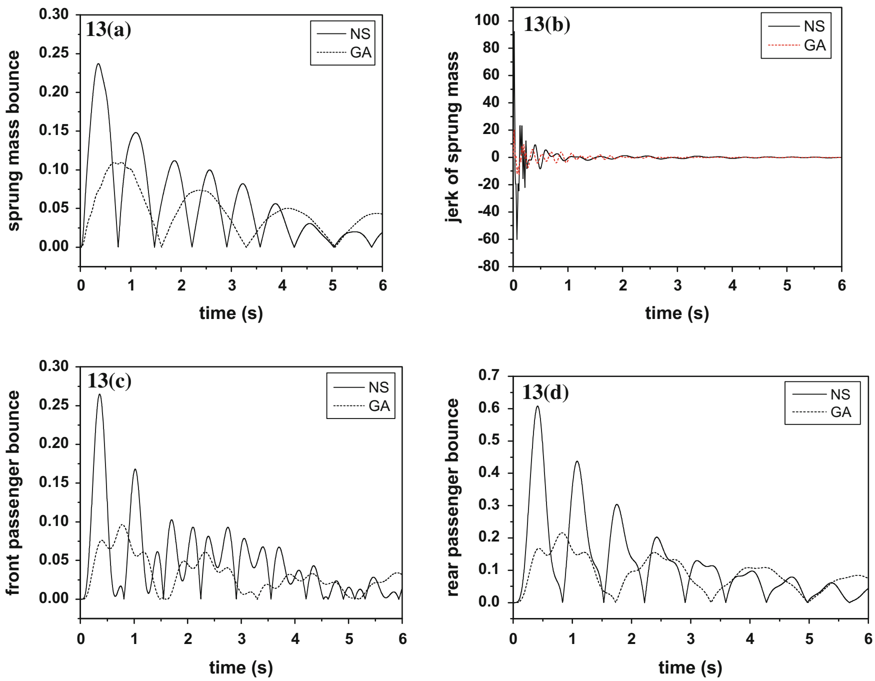


Fig. 9 Comparison between the output parameters for NS and GA results under Case-1 (a) road

Case-2 road situations respectably. The comparison made shows that with the GA parameters, the maximum sprung mass bounce, sprung mass jerk and passengers bounce, respectively, are found to be far less than those obtained by applying NS. The GA optimization technique can serve as efficient design method for vehicle suspension system.

7 Conclusion

This paper portraits an application of real life problem. The goal in this work is to design suspension parameters, in such a way that the passengers comfort is not decorated by forces induced by the road bump(s) or irregular rough surface. The study performed by considering varying mass of the passengers further focuses on the limits and variation of comfort, basically bounce and jerk they experience due to their change in mass under different road conditions. The tabulated results and graphical representations show that the suspension parameters found by GA can serve as an efficient value to enhance comfort.

Table 2 Comparison between the output parameters under different road

	Case-1(a)			Case-1(b)			Case-2		
	NS	GA	% Diff	NS	GA	% Diff	NS	GA	% Diff
Sprung mass bounce	0.2372	0.1095	53.84 %	0.2367	0.1259	46.81 %	0.0848	0.0313	63.09 %
Front passenger bounce	0.2649	0.0964	63.61 %	0.2654	0.0994	62.55 %	0.1634	0.039	76.13 %
Rear passenger bounce	0.608	0.2161	64.46 %	0.6066	0.2593	57.25 %	0.1199	0.033	72.48 %
Jerk of sprung mass	92.2897	19.9558	78.38 %	105.2174	19.6359	81.34 %	82.1653	19.8514	75.84 %

Table 3 Range of vibration of the output parameters with varying passengers' mass

		Case-1 (a)	Case-1 (b)	Case-2
Sprung mass	NS	[0.2341, 0.2578]	[0.2336, 0.2572]	[0.0840, 0.0931]
Bounce	GA	[0.1094, 0.1141]	[0.1257, 0.1305]	[0.0310, 0.0322]
Front passenger	NS	[0.1901, 0.2831]	[0.1899, 0.2834]	[0.1374, 0.1676]
Bounce	GA	[0.0844, 0.0976]	[0.0897, 0.1003]	[0.0379, 0.0410]
Rear passenger	NS	[0.4473, 0.6226]	[0.4456, 0.6211]	[0.1136, 0.1225]
Bounce	GA	[0.1943, 0.2185]	[0.2419, 0.2650]	[0.0309, 0.0488]
Sprung mass	NS	[92.2896, 92.2932]	[105.2171, 105.2227]	[82.1652, 82.1676]
Jerk	GA	[19.9558, 19.9567]	[19.6359, 19.6371]	[19.8514, 19.8520]

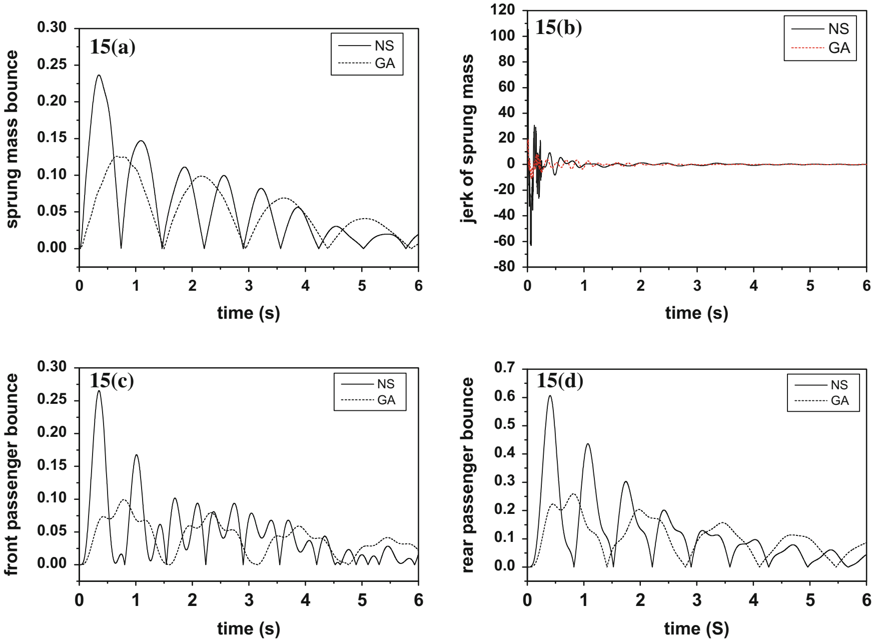


Fig. 10 Comparison between the output parameters for NS and GA results under Case-1 (b) road

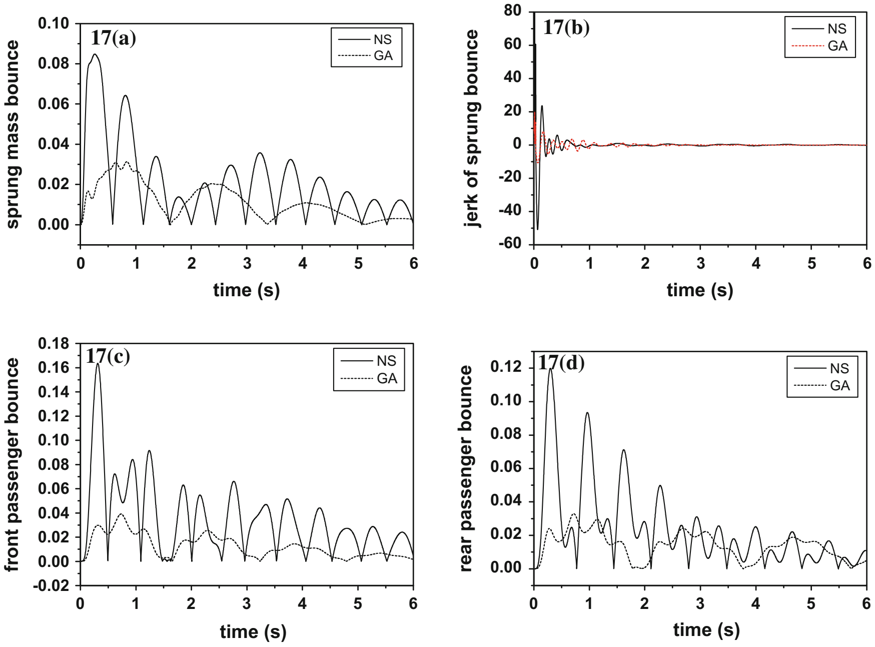


Fig. 11 Comparison between the output parameters for NS and GA results under Case-2 road

References

1. Jazar, R.N.: Vehicle Dynamics: Theory and Applications. Springer (2008)
2. Alkhatib, R., Jazar, G.N., Golnaraghi, M.F.: Optimal design of passive linear suspension using genetic algorithm. J. Sound Vib. **275**, 665–691 (2004)
3. Bourmistrova, A., Storey, I., Subic, A.: Multiobjective optimization of active and semi-active suspension systems with application of evolutionary algorithm. In: International Conference on Modeling and Simulation, Melbourne, pp. 12–15 (2005)
4. Yagiz, N., Hacioglu, Y., Taskin, Y.: Fuzzy sliding-mode control of active suspensions. IEEE Trans. Ind. Electron. **55**(11), 3883–3890 (2008)
5. Cao, J., Li, P., Liu, H.: An interval fuzzy controller for vehicle active suspension systems. IEEE Trans. Intell. Transp. Syst. **11**, 885–895 (2010)
6. Dong, X.M., Yu, M., Liao, C.R., Chen, W.M.: Comparative research on semi-active control strategies for magneto-rheological suspension. Nonlinear Dyn. **59**(3), 433–453 (2010)
7. Yoshimura, T., Kume, A., Kurimoto, M., Hino, J.: Construction of an active suspension system of a quarter car model using the concept of sliding mode control. J. Sound Vib. **239**, 187–199 (2011)
8. Sun, L.: Optimum design of 'road-friendly' vehicle suspension systems subjected to rough payment surfaces. Appl. Math. Model. **26**, 635–652 (2002)
9. Bucak, I.O., Zohdy, M.A., Shillor, M.: Motion Control of a Nonlinear Spring by Reinforcement Learning. Control Intell. Syst. ACTA Press **1**, 27–36 (2008)
10. Els, P.S., Uys, P.E., Snyman, J.A., Thoresson, M.J.: Gradient-based approximation methods applied to the optimal design of vehicle suspension systems using computational models with severe inherent noise. Math. Comput. Model. **43**, 787–801 (2006)

11. Ozgur, D., Iiknur, K., Saban, C.: Modeling and control of a nonlinear half-vehicle suspension system: a hybrid fuzzy logic approach. Nonlinear Dyn. **67**, 2139–2151 (2012)
12. Çetin, S., Akkaya, A.V.: Simulation and hybrid fuzzy-PID control for positioning of a hydraulic system. Nonlinear Dyn. **61**, 465–476 (2010)
13. Uys, P.E., Els, P.S., Thoresson, M.: Suspension settings for optimal ride comfort of off-road vehicles traveling on roads with different roughness and speeds. J. Terrramech. **44**, 163–175 (2007)
14. Hongyi, L., Honghai, L., Steve, H., Chris, H.: Design of robust H∞ controller for a half-vehicle active suspension system with input delay. Int. J. Syst. Sci. **44**(4), 625–640 (2013)
15. Sun, L., Cai, X., Yang, J.: Genetic algorithm-based optimum vehicle suspension design using minimum dynamic pavement load as a design criterion. J. Sound Vib. **301**, 18–27 (2007)
16. Zadeh, N.N., Salehpour, M., Jamali, A., Haghgoo, E.: Pareto optimization of a five-degree of freedom vehicle vibration model using a multi-objective uniform-diversity genetic algorithm (MUGA). Eng. Appl. Artif. Intell. **23**(4), 543–551 (2010)
17. Vladimír, G., Marian, K.: Optimization of vehicle suspension parameters with use of evolutionary computation. Procedia Eng. **48**, 174–179 (2012)
18. Jabeen, S.D.: Vibration optimization of a passive suspension system via genetic algorithm. Int. J. Model. Simul. Sci. Comput. **4**(1), 1250022-1–1250022-21 (2013)
19. Jabeen, S.D., Saha, J., Mukherjee, R.N.: On optimal design of passive suspension using GA based heuristic method. Int. J. Mater. Struct. Integrity **4**(1), 59–86 (2010)
20. Feng, J.Z., Li, J., Yu, F.: GA-based PID and fuzzy logic control for active vehicle suspension system. Int. J. Auto. Tech. **4**(4), 181–191 (2003)

Performance Evaluation of Various Classifiers in Emotion Recognition Using Discrete Wavelet Transform, Linear Predictor Coefficients and Formant Features

Allen Joseph and Rajeswari Sridhar

Abstract In this paper, a comparison is made on the classifiers K*, Neural network and Random forest for identifying emotion, based on a combination of Discrete Wavelet Transform (DWT), Linear Predictor Coefficients (LPC) and formant features. The feature set has been arrived after carrying out a survey on the existing works of emotion identification. The paper finally concludes with the apt choice of the classifier for the chosen feature set to identify emotion.

Keywords Discrete wavelet transform · Emotion identification · Formant · Linear predictor coefficients

1 Introduction

Emotions in humans are different for different ages. From a child who does not know what will happen if he/she touches the lighted candle to an adult who touches a lighted candle, emotions are different [1]. The child may cry whereas an adult may not. So, it is typical that speech or sound information may be used to distinguish between different emotions present among different ages. In our case, a child opens his/her mouth to cry but an adult may or may not open his mouth. This simple notion can be exploited to distinguish different emotions which are based on acoustic information. Depending on the opening or closing of the mouth, the acoustic information associated with it can be classified as voiced and unvoiced sound [2]. From the voiced and unvoiced signals of the sound, we can extract

A. Joseph (✉) · R. Sridhar
Department of Computer Science and Engineering, Anna University,
College of Engineering, Guindy, Chennai, India
e-mail: allenjoseph1985@gmail.com

R. Sridhar
e-mail: rajisridhar@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_35

certain features which are used to identify the emotion associated with the input sound.

This paper presents an overview on the features that are available from a typical speech signal and the features that could be used for emotion recognition. In this work, we propose a combination of features which has been used for an alternate speech processing purpose and use them for identifying emotion. The choice of classifier is very important and we compare the performances of three classifiers, namely K*, Neural network and Random forest.

Organization of this paper is as follows: Sect. 2 discusses on the state of the art work in choice of features for emotion identification, Sect. 3 on the proposed features combination and their implications, Sect. 4 on results and discussion and finally Sect. 5 concludes the work with possible future work.

2 Related Works

Generally, speech features could be categorized as Continuous and Cepstral [3]. Pitch, energy and formant are classified as continuous and Mel Frequency Cepstral Coefficients (MFCC) and LPC fall under Cepstral features.

Eszter et al. [4] manipulated the differences between speakers to identify emotions and they applied it to natural speech for the languages Austrian and Hungarian. The main disadvantage is that acted speech differs in pausing and accentuation. So, for our purpose also, we used natural speech. For feature extraction, MFCC is a dominant method in automatic speech recognition and is also used for speech emotion identification; the features obtained are reduced using Principal Component Analysis (PCA) [5]. But MFCC method of feature extraction has become common and they have adverse effects like lack of interpretation when used as a standalone methodology, so fusion-based approach is recommended for emotion identification. Siqing et al. [6] worked on features extracted from an auditory-inspired long-term spectro-temporal representation and classify the emotions using Support Vector Machines (SVM) and Radial Basis Function (RBF). Chengxi et al. [7] work on Perceptual Linear Prediction (PLP) coefficients for feature extraction and perform classification in the tangent space.

Survey shows that for feature extraction one can observe that mostly Linear Predictor Cepstral Coefficients (LPCC), MFCC, formant, a combination of LPCC + formants and MFCC + formants are used for feature extraction [8]. Normally, extensive research is carried out in fusion-based algorithms. Some of the fusion-based algorithms are fusion of DWT and MFCC [9] where they do a six-level wavelet decomposition in which much of the information is lost, so we go in for single level wavelet decomposition where most of the information is preserved. Others include fusion of pitch and MFCC [10].

Mohammad et al. [11] also describe about fusion-based analysis in their work, wherein they discuss about segmenting the input speech signal and classify the extracted features and fuse to obtain the result. From the above-mentioned studies, it is recommended that we use segment-based analysis for feature extraction and work on natural language. This work proposes a primitive approach for segment-based analysis on natural speech. The feature, formants (formants 1, 2 and 3) are extracted using LPC and before that DWT is applied to the segmented signal. Using these features and using K*, Random Forest and Neural Network as classifiers, emotion is identified from the input speech signal.

3 The Proposed System

At a given instance of time, the resonances of the vocal tract system are known as formants and they have bandwidth and amplitude which are unique to each of the sound units [8]. These features can be observed in the frequency domain.

First, the input speech signal is preprocessed and the DWT is applied to the preprocessed signal. After the transformation, the formants are found by taking the approximation coefficients of a single level DWT and applying LPC. The overall block diagram is shown in Fig. 1 and the individual modules are discussed in the following sub-sections.

3.1 Preprocessing

Generally, preprocessing involves removal of noise from the signal. Pre-emphasis is used to remove the DC component from the signal. In the acquired speech signal, to compensate for high-frequency component, a digital filter as expressed in Eq. 1 is used.

$$s'(n) = s(n) - \tilde{a}s(n-1) \tag{1}$$

$$H(z) = 1 - \tilde{a} * z^{-1} \tag{2}$$

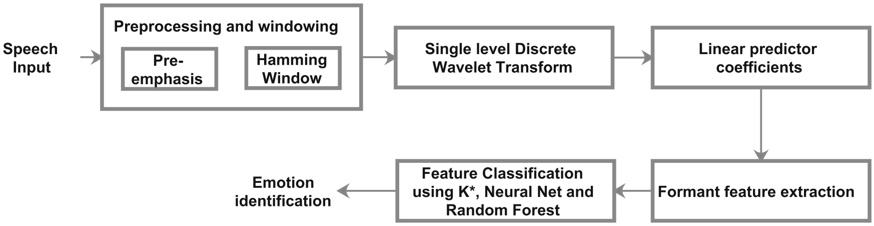

Speech Input
Preprocessing and windowing
Pre-emphasis
Hamming Window
Single level Discrete Wavelet Transform
Linear predictor coefficients
Formant feature extraction
Feature Classification using K*, Neural Net and Random Forest
Emotion identification

Fig. 1 Block diagram of the overall system

where n is the number of samples, and ã is the pre-emphasis filter coefficient. Equation 2 represents the z-transform of the filter. The degree of the pre-emphasis filter is controlled by a positive parameter, which is normally between 0.9 and 1.0 and for windowing, Hamming window is used.

3.2 Wavelet Transform

A window which is fixed is used uniformly for a spread of frequencies in Fourier Transform [12]. However, a wavelet transform uses a window function which is long at low frequencies and short at high frequencies [13]. This enables the characteristics of non-stationary nature of speech signals to be easily examined [9]. Wavelet Transform coefficients are localized in both time and frequency domains. But this localization has a constraint according to Heisenberg's uncertainty principle, which states that no transform can have high resolution in both time and frequency domains at the same time [14]. This useful locality property is exploited here. Since, the wavelet basis functions are generated by scaling from a mother wavelet; they are well localized in time and frequency domains.

This property of wavelet is suitable for processing of speech signal that requires high temporal resolution to analyze high-frequency components (mostly unvoiced sounds) and high-frequency resolution to analyze low-frequency components (voiced sounds and formant frequencies) [15].

Wavelets are a family of functions constructed from translations and dilations of a single function called the "mother wavelet" denoted by $\psi(\mathrm{t})$. They are expressed as:

$$\psi_{\mathrm{a,b}}(\mathrm{t}) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{\mathrm{t}-\mathrm{b}}{\mathrm{a}}\right) \tag{3}$$

where a is the scaling parameter, and it measures the degree of compression. b is the translation parameter which determines the time location of the wavelet. If $|a| < 1$ then the wavelet is compressed version of the mother wavelet and corresponds mainly to higher frequencies and if $|a| > 1$, then $\psi_{\mathrm{a,\ b}}$ (t) has a larger time-width than ψ (t) and corresponds to lower frequencies. Wavelet transform is applied to the input preprocessed signal and then the formant features are extracted using LPC.

3.3 Linear Predictor Coefficients (LPC)

We use the linear predictor coefficients to extract the formants. LPC are the coefficients of an auto-regressive model of a speech frame. The all-pole representation of the vocal tract transfer function is represented as:

$$H(z) = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{4}$$

where a_p are the prediction coefficients and G is the gain

The LPC can be derived by minimizing the mean square error between the actual samples of speech frame and the estimated samples by autocorrelation method. The formants (formant 1, 2 and 3) are obtained passing the signal to get LPC coefficients first and then the roots are calculated to determine the bandwidth of the formants.

4 Results

4.1 Dataset

The data that were used contained 609 samples of different emotions of male and female speakers. The data were obtained as.wav files and were fed to Matlab for processing. The database was English. A 3-D plot for different emotions with formants being the axes is shown in Fig. 2.

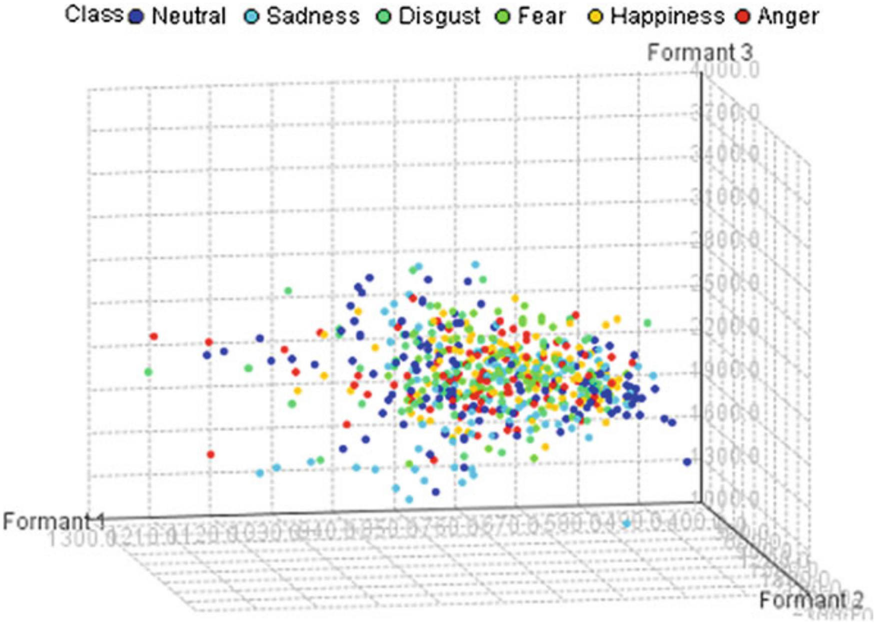


Fig. 2 3-D plot for different emotions

Table 1 Results of K* classifier with an average classification accuracy of 95.0739 %

	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC area	PRC area	Class
	0.962	0.021	0.927	0.962	0.944	0.929	0.998	0.993	Neutral
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Sadness
	0.885	0.011	0.928	0.885	0.906	0.891	0.996	0.975	Disgust
	0.847	0.027	0.856	0.847	0.851	0.823	0.991	0.954	Fear
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Happiness
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	0.999	Anger
Weighted avg.	0.951	0.011	0.951	0.951	0.951	0.940	0.998	0.988	

4.2 Experiment 1

The above-mentioned procedure in Sect. 3 is used for feature extraction and then we classify the signal using K* which is an instance-based learner that uses entropy as a distance measure for classifying the given dataset. The perception is that the distance between instances can be defined as the complexity of transforming one instance into another. Table 1 shows the results obtained using K* classifier for the various parameters of True positive, False positive, etc.

In this table

TP is the True-Positive rate which is the proportion of examples which were classified as class x, among all examples which truly have class x
FP is the False-Positive rate which is the proportion of examples which were classified as class x, but belong to a different class,
Precision is the proportion of the examples which truly have class x among all those which were classified as class x
F-measure is 2*Precision*Recall/(Precision + Recall)
MCC is Mathews Correlation Coefficient is given by,

$$(\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}) / \sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

ROC—Receiver Operating Characteristics

PRC—Precision recall Curve [16].

4.3 Experiment 2

In experiment no. 2, Random Forest classifier is used for classification which works on the principle of ensemble approach. Wherein, a group of “weak learners” can come together to form a “strong learner”. Table 2 shows the results obtained using Random Forest classifier

Table 2 Results of Random Forest classifier with an average classification accuracy of 92.9392 %

	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC area	PRC area	Class
	0.970	0.036	0.883	0.970	0.924	0.904	0.992	0.969	Neutral
	0.972	0.006	0.972	0.972	0.972	0.966	0.999	0.991	Sadness
	0.885	0.021	0.875	0.885	0.880	0.860	0.993	0.951	Disgust
	0.776	0.020	0.884	0.776	0.826	0.798	0.986	0.930	Fear
	0.979	0.002	0.990	0.979	0.984	0.982	1.000	0.998	Happiness
	0.978	0.002	0.989	0.978	0.983	0.980	0.999	0.995	Anger
Weighted avg.	0.929	0.016	0.930	0.929	0.929	0.915	0.995	0.972	

Table 3 Results of neural net classifier with an average classification accuracy of 31.5271 %

	TP rate	FP rate	Precision	Recall	F-measure	MCC	ROC area	PRC area	Class
	0.833	0.560	0.292	0.833	0.432	0.232	0.731	0.478	Neutral
	0.406	0.153	0.358	0.406	0.381	0.241	0.705	0.401	Sadness
	0.000	0.000	0.000	0.000	0.000	0.000	0.607	0.196	Disgust
	0.255	0.080	0.379	0.255	0.305	0.207	0.606	0.268	Fear
	0.082	0.045	0.258	0.082	0.125	0.063	0.658	0.274	Happiness
	0.067	0.017	0.400	0.067	0.115	0.114	0.631	0.202	Anger
Weighted avg.	0.315	0.171	0.286	0.315	0.246	0.152	0.663	0.318	

4.4 Experiment 3

Neural networks which use multilayer perceptron are used as a classifier in experiment 3. The neural net is grouped into a series of layers, where the input vector enters at the left side of the network, which is then projected to a "hidden layer". Each unit in the hidden layer is a weighted sum of the values in the first layer. This layer then projects to an output layer, where the output is obtained. Table 3 shows the results obtained using neural net classifier.

4.5 Analysis

Neural network normally performs better but, here, since we have "weak learners", K* and Random Forest perform better. The average classification accuracy for neural net is 31.52 % because we have the weighted average of true-positive rate to be only 0.315, whereas for Random forest and K* the classification accuracy is 92.93 % and 95.07 % with a weighted average of true-positive rate to be 0.929 and 0.951, respectively. The various performance measures for different classifiers are

Fig. 3 Performance graph for neural network classifier with their TP and FP rate

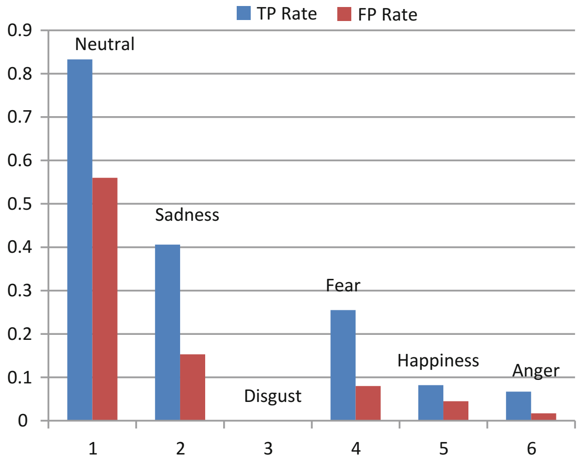
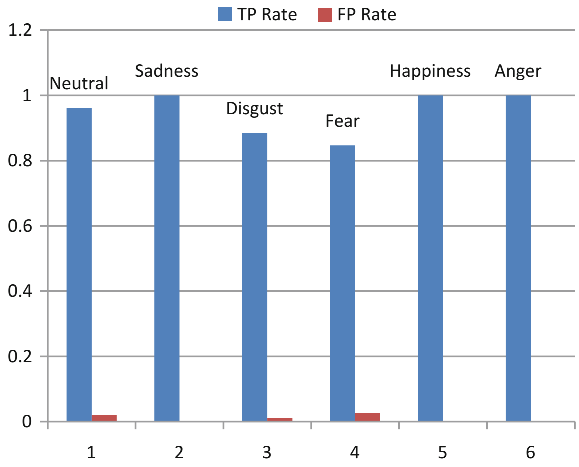


Fig. 4 Performance graph for K* classifier with their TP and FP rate



shown in Figs. 3, 4 and 5. Wherein, we have the graphs to be varying with respect to the true-positive and false-positive rate.

When we make a comparative study on the three different classifiers based on the observations, we find that the framework gives optimal results for neutral, sadness, happiness and anger in the case of K* classifier with a TP rate ranging 0.962 to 1, the reason being better feature extraction. And, we have optimal results in the case of Random Forest classifier with neutral, sadness, happiness and anger having a TP rate ranging from 0.970 to 0.979. The result of Neural Network classifier shows that the performance is not better because the maximum TP rate is only 0.833 for neutral and much lower for all the other emotions.

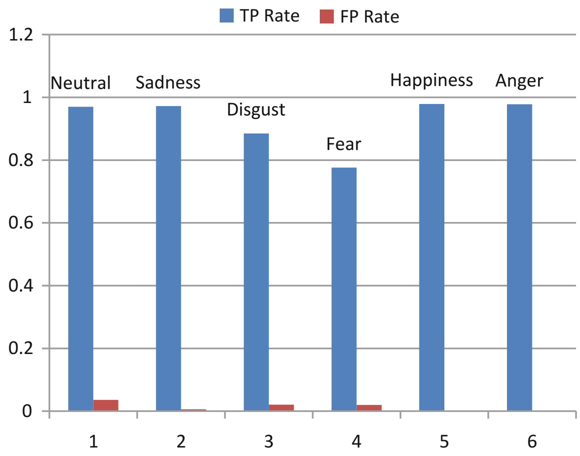


Fig. 5 Performance graph for Random Forest classifier with their TP and FP rate

5 Conclusion and Future Work

The work done so far projects the use of different classifiers along with the use of DWT, LPC and Formants which is used for feature extraction. Results show that K* and Random forest classifiers perform better than Neural network classifier. Future work can be carried out using different classifiers in combination or designing new set of features.

References

1. Reevy, G., Malamud Ozer, Y., Ito, Y.: Encyclopedia of Emotion, vol. 1. ABC-CLIO (2010)
2. Duck, S., McMahan, D.T.: The Basics of Communication: A Relational Perspective. SAGE (2011)
3. El Ayadi, M., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: features, classification schemes, and databases. Pattern Recognit. Elsevier **44**, 572–587 (2011)
4. Tisljár-Szabó, E., Pléh, C.: Ascribing emotions depending on pause length in native and foreign language speech. Speech Commun. Elsevier **56**, 35–48 (2014)
5. Grimm, M., Kroschel, K., Mower, E., Narayanan, S.: Primitives-based evaluation and estimation of emotions in speech. Speech Commun. Elsevier **49**, 787–800 (2007)
6. Wu, S., Falk, T.H., Chan, W.-Y.: Automatic speech emotion recognition using modulation spectral features. Speech Commun. Elsevier **53**, 768–785 (2011)
7. Ye, C., Liu, J., Chen, C., Song, M., Bu, J.: Speech emotion classification on a Riemannian manifold. In Proceedings of the International Conference on Advances in Multimedia Information Processing—PCM, pp. 61–69 (2008)
8. Rao, K.S., Kool-agudi, S.G.: Emotion Recognition Using Speech Features. Springer (2013)
9. Verma, G.K., Tiwary, U.S., Agrawal, S.: Multi-algorithm fusion for speech emotion recognition. Springer, Advances in Computing and Communications, Communications in Computer and Information Science, vol 192, pp. 452–459, 2011

10. Mahdhaoui, A., Ringeval, F., Chetouani, M.: Emotional speech characterization based on multi-features fusion for face-to-face interaction. In: Proceedings of International Conference on Signals, Circuits and Systems, pp. 1–6 (2009)
11. Shami, M., Verhelst, W.: An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech. Speech Commun. Elsevier **49**, 201–212 (2007)
12. Fourier, J.B.J.: The Analytical Theory of Heat. The University Press (1878)
13. Jensen, A., la Cour-Harbo, A.: Ripples in Mathematics: The Discrete Wavelet Transform, Springer Science & Business Media (2011)
14. Heisenberg, W.: 'Ueber den anschaulichen Inhalt der quantentheoretischen Kinematik and Mechanik' Zeitschrift für Physik, vol. 43, pp. 172–198. English translation in (Wheeler and Zurek, 1983), pp. 62–84 (1927)
15. Nehe, N.S., Holambe, R.S.: DWT and LPC based feature extraction methods for isolated word recognition. EURASIP J. Audio Speech Music Process., 1–7 (2012)
16. Witten, I.H., Frank, E.: Data Mining Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)

A Quantitative Error Map Generation for Modification of Manual Socket for Below-Knee Amputee

Arun Dayal Udai and Amarendra Nath Sinha

Abstract Traditional prosthetic socket manufactured through manual plaster-casting method for lower limb prosthesis consists of geometrical errors since the shape of the cast changes due to hand impressions, pressure involved while taking out the negative cast and finally during hand sculpting of the positive cast. As the Plaster of Paris (PoP) model is further utilized for manufacturing of prosthetic limb, the error gets carried over to the prosthetic limb, which results in improper fit of the artificial limb and ultimately wounding the amputees residual limb, particularly at the stress-prone knee. In the present work, an error map generation using reverse engineering (RE) technique has been proposed which assists in rectifying the manually prepared PoP model. The correction technique is quantitative and requires less hand-sculpting skills of the prosthetist. With this process, the accuracy increases to the level of a product manufactured by CNC/RP method and at a comparatively much cheaper rate.

Keywords Reverse engineering · Prosthetic socket · CAD

1 Introduction

The process of developing of artificial human limbs for below-knee (BK) amputees has nearly come up to the level where it is easily being manufactured through manual as well as advanced CAD/CAM techniques. The research work still has scope to achieve desired accuracy in the custom-fit design as well as new innovation in data

A. Dayal Udai (✉)
Department of Mechanical Engineering, Birla Institute of Technology, Mesra, Ranchi 835215, India
e-mail: arun_udai@bitmesra.ac.in
URL: http://www.bitmesra.ac.in

A. Nath Sinha
School of Engineering and Technology, Jagran Lakecity University, Bhopal, India
e-mail: dransinha@jlu.edu.in
URL: https://www.jlu.edu.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_36

acquisition and process refinement for its production where it nearly matches the CAD data. Such a design process is expected to improve the quality, shorten lead time, reduce cost and improve data acquisition process. The overall process should be simple so that technicians can be easily trained and widespread in the developing countries. Several attempts have been made by researchers to implement some of the stated qualities of a custom-fit artificial limb ranging from data acquisition process to the manufacturing process. Some of the works improve the data acquisition process while a few of them refine the post-processing methods of the cloud data to generate a 3D CAD model. Kurt et al. [4] discuss a stump measuring system and socket mold is prepared using NC milling machine and CAD workstation. A 3D surface scanner for lower limb prosthetics is discussed by [3] that uses four CID cameras and three white light projectors. Walsh et al. [9] developed a complete system for manufacturing prostheses using RE.

The conventional manual manufacturing of artificial limbs by first developing negative and positive cast of residual limbs does not give accuracy which ultimately results in pain and difficulty to amputees putting artificial limbs on his residual part of live limb [2]. Both measurement and manual manufacturing results in inaccuracy because of the irregular shape and size of the amputees residual limb. RE is utilized by [5] for CAD model preparation of customized artificial joint, which used a Coordinate Measuring Machine (CMM). An alternative means of 3D reconstruction is attempted by [6] using computerized tomography (CT) scanning, image processing and RE technique. A similar process is utilized by [1] using spiral X-ray CT scanning and a different CAD package for surface development. In the present work a process is developed which uses an RE technique to prepare a quantitative error map which is utilized to modify the manually prepared Plaster of Paris (PoP) model of the amputees residual limb. The modified PoP model is utilized to manufacture the fiber-reinforced composite socket which is to be fitted to the manually prepared artificial limb.

This ongoing multistage endeavor involves acquisition of cloud points through non-contact white light 3D scanner, development of wire frame model and finally the surface model of the residual part of the limb as well as the manually prepared PoP cast. The processes of RE followed in steps are preparing the residual limb surface, fixing registration marks, 3D scanning, cloud registration and alignment, data filtering, segmentation, curve fitting to build wire frame geometry and building of the surface model. These steps thoroughly applied with few innovative techniques and modifications to the traditional techniques in use. Each subsequent transformation, like cloud points to curve and curves to surface are analyzed and modified iteratively, to obtain a final shape with least possible error. The surface model for both the PoP cast and the live residual limb is aligned together and an error map is prepared. This process supplements traditional method of manufacturing artificial limb with more quantitative techniques for rectification of artificial limb.

2 Methodology

The error map generation procedure involves (a) digitizing the surfaces of amputee limb and the manually prepared positive plaster model; (b) generating 3D surface model for both the digitized data set using RE method; (c) aligning the surface model to a common axis and (d) generating a quantitative error data set (a color-coded three-dimensional image) with reference to a geometrical position on plaster model. UGS Imageware® 12.0 is used for the above all processes as it is widely available and easier for any less skilled prosthetists to learn. Since RE is more of an art work than an engineering it depends very much on individual skill. The method discussed quantifies the prosthetists job with a more scientific data and supplements the existing methods for the modification of manual artificial sockets.

2.1 Digitizing Surfaces

The amputees residual limb consists of soft tissues, uneven surfaces and texture, that makes it difficult to scan using most precise white light scanner. So, the surfaces were sprayed with a non-toxic white color and scanned with a minimum possible number of views in very short interval, within which, the limb remains virtually stable over the fixture. The cloud points were obtained at constant room temperature of 25 °C, so as to have an uniform optical properties of the air medium. Proper registration marks were placed at appropriate locations (as shown in Fig. 1), which assist in alignment of multiple view cloud sets and in referencing of error map over the plaster model in later stages of the process. Each registration mark was made visible from at least

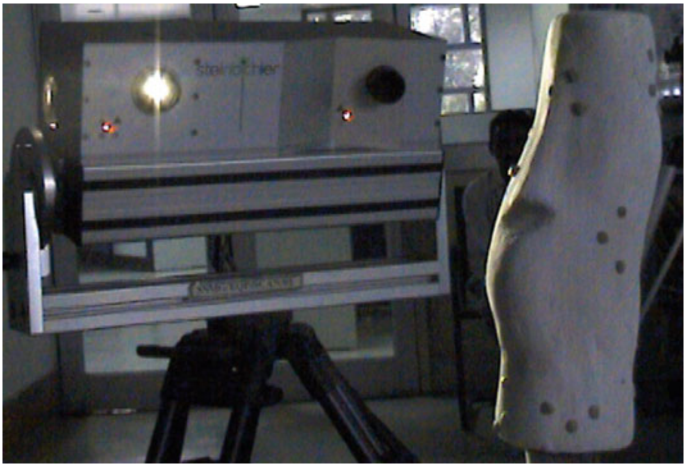


Fig. 1 Scanning process with registration marks

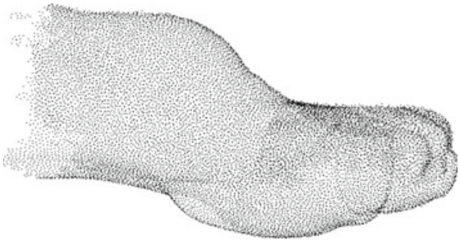


Fig. 2 Completed scanned cloud data of amputee's residual limb

two views of the scanner. This made it easier to identify any particular view cloud data during alignment of all the cloud point to a single coordinate system.

The cloud points for the limb and the plaster model were followed with cordial deviation and space sampling method for data reduction, smoothening and filtering of any redundant points. Similar procedure was adapted for scanning of the manually prepared plaster model. The completed cloud points as shown in Fig. 2, of the plaster and live limb was processed using Imageware® package to form surface models of each in standard RE steps of segmentation, curve fitting and surface generation.

Geometrical feature-based cloud segmentation method was adapted, which segregates the tubular cloud data along the length of the limb and the dome-shaped clouds near to the lower end of the limb. Any remaining shapes were classified as transition shapes which were covered with edge-bounded surface patches.

2.2 Generating 3D Surface Model Using RE Technique

Along the tubular cloud data a curve aligned 30–40 transverse sections was made, which was fitted with closed B-Spline curves as discussed in [8]. This is shown in Fig. 3. Curves being the base for surface generation, more effort was made at the curves level to rectify any possible chances of surface error. The start points of all the curves were aligned and curve parameters were made to flow in same direction. Any unnecessary control points were removed to a certain tolerance and the knots are redistributed uniformly along the length of the curve. These set of curves were

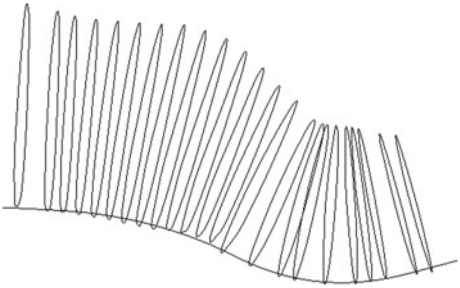
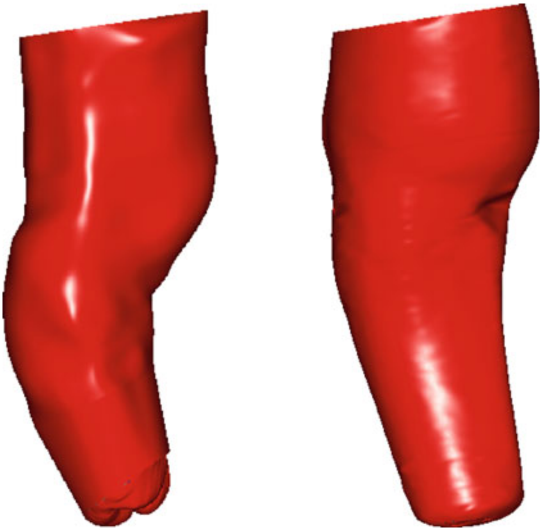


Fig. 3 Set of closed B-Spline curves fitted over sectioned cloud points

Fig. 4 Surface models of live limb and plaster cast



(a) Surface model of live limb

(b) Surface model of plaster cast

utilized to generate a tubular loft surface. The dome-shaped caps at the end of the lower limb was fitted with loft surfaces as shown in Fig. 4.

In Fig. 4, at each stage of cloud-to-curve and curve-to-surface generation, the deviation from the original cloud was inspected, and the curves or surfaces were rectified to diminish the error within permissible limits. Modifications were made by moving the control points normal to the curves or surface. A first order and second order of continuity was established between any joining surfaces as in [7]. Surface models for the live limb and the plaster casting were generated with similar steps. As the cloud points for plaster were more uniform, the surfaces generated were also smoother.

2.3 Aligning the Surfaces to a Common Axis

Axis of the prepared CAD surface model of the live limb or the plaster was derived from the best fit line through the centers of the circle fitted over the transverse slices, along the length of the completed scanned cloud data. The axis may also be formed with the centroid of the sectioned cloud data. But the former method was adopted as it is simple and also has the desired accuracy. The axes along with the surfaces for live limb and the plaster model was brought together to form a single axis. When fixing one of the surfaces (live model in the current case), the other one was moved linearly along the axis and rotated about the axis to obtain a best fit through and

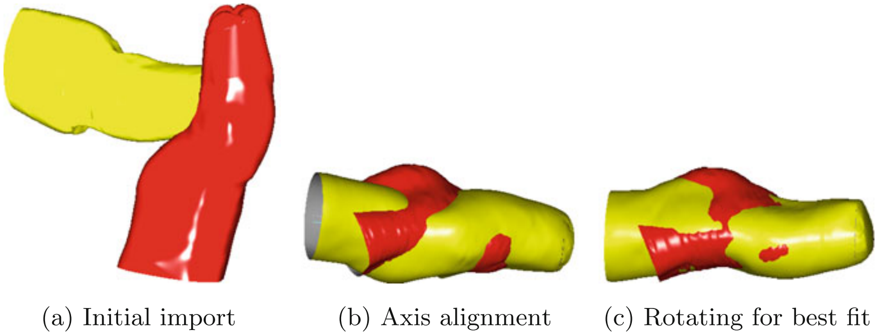


Fig. 5 Alignment of surfaces along the common axis

automated iterative process. Some of the intermediate steps are shown in Fig. 5. The other degrees of freedom (DOF) for the moving cloud was constrained.

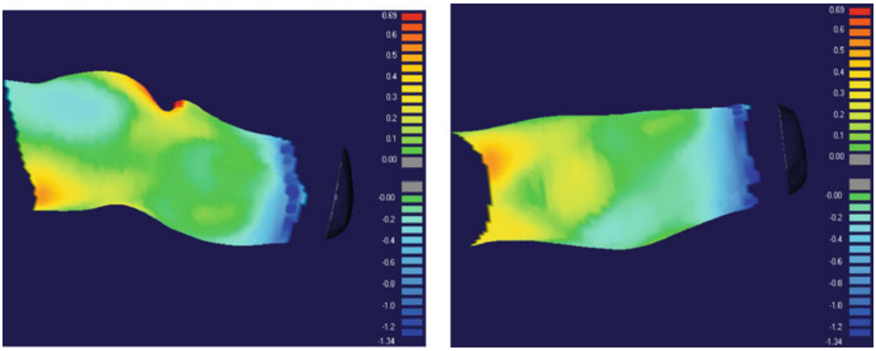
2.4 Generating Quantitative Error Map and Its Implementation

Considering the live model as the standard reference, the deviation of plaster model surface was inspected. A three-dimensional color plot was obtained which shows the maximum and minimum deviations from the standard reference. The distance between the surfaces was calculated, based on the surface normal to the reference surface to the plaster model surface. The deviations corresponding to any location was outlined for manual modification to the plaster cast. The location on the plaster model corresponding to any location on its CAD counterpart was found with respect to the registration points fixed while scanning. The deviations were marked on the plaster and outlined for manual modification. In the automated process the plaster model may be directly modified on CAD platform and manufactured using NC, CAM or any rapid prototyping system.

In the current work, a manually modified plaster cast was taken for socket preparation using fiber-reinforced epoxy resin socket. The original plaster cast was lost while taking off the socket. However, the record may be maintained on any CAD database.

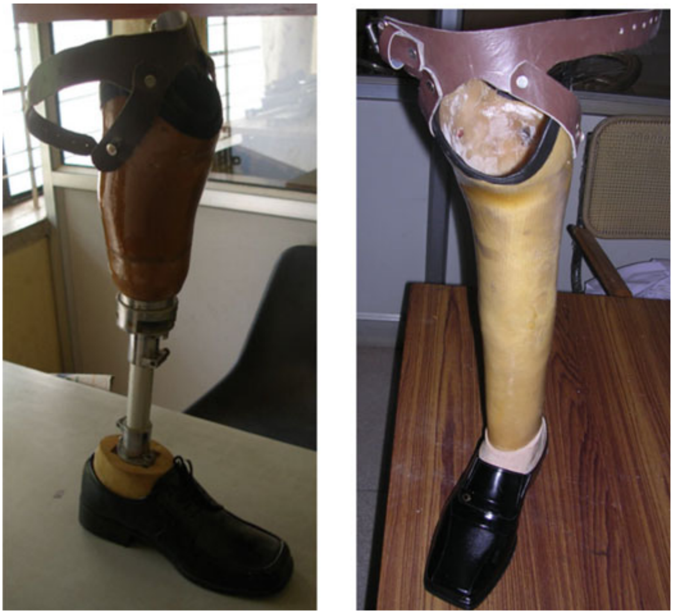
3 Results and Discussion

Surfaces for the CAD model were compared with the original cloud obtained after scanning and 90 % of the surfaces were within ±3.0 mm of deviation which was manually corrected up to ±0.5 mm accuracy. A slope variation between the surfaces joined below 5° was accepted. A maximum gap of 0.05 mm existed between



(a) Error-map front view (b) Error-map top view

Fig. 6 Color-coded error map



(a) Artificial leg of first case (b)Artificial leg of second case

Fig. 7 Prepared artificial legs with the modified sockets

the surface joints. Volume measurements of the solid models were found to correspond within 3–5 % of surface models and direct determinations was made using Archimedean weighing (as in [1]), higher being the live limb error. Such a model can be exported to any automated manufacturing machine like RP or a CNC machine. The surface difference was within limits of −1.34 mm to +0.69 mm over the working area of the sockets as shown in Fig. 6.

Two below-knee amputee cases were fully rehabilitated with manually prepared artificial limbs socket modified using the current technique. This is shown in Fig. 7. Figure 7a shows the exoskeleton and Fig. 7b shows the endoskeleton type of artificial limb structure.

The method not only supplements the existing manual plaster-casting technique with a quantitative method for modification, but also eliminates skill dependency or any art work of prosthetist, with minor additions to the infrastructure cost of the CAD systems. The proposed method is expected to be useful for clinical evaluation, recordkeeping, education/research and transferring manual modification skills to CAD system. As the work has been carried out for below-knee amputee cases only, the discussions are made with reference to BK amputee. This may be extended for any above-knee amputee or hand amputation case also.

4 Conclusions

The paper presents a methodology to enhance the precision of a prosthetic socket manufactured using hand-sculpting technique. The proposed method was demonstrated using a below-knee amputee case. However, this may be extended for any other amputee cases as well. The paper used a CAD-based reverse engineering technique to quantify the errors existing on a hand-sculpted PoP cast and generated an error map that may be used to enhance the precision of the final socket manufactured out of it.

Acknowledgments This work was carried out with the financial support from DST, Govt. of India, through one of its project at BIT, Mesra, Ranchi. The authors would also like to thank District Disabled Rehabilitation Center, Ranchi, for their support during fabrication of the artificial limb. Scientists at CMERI, Durgapur, are also acknowledged for their contributions in obtaining the cloud data.

References

1. Bhatia, G.H., Commean, P.K., Smith, K.E., Vannier, M.W.: Automated lower limb prosthesis design. Proc. SPIE, Visual. Biomed. Comput. **2359**, 493–503 (1994)
2. Cochrane, H., Orsi, K., Reilly, P.: Lower limb amputation, Part 3: prosthetics a 10 year literature review. J. Prosthet. Orthot. **25**, 21–28 (2001)
3. Commean, P.K., Smith, K.E., Vannier, W.: Design of a 3-D surface scanner for lower limb prosthetics: a technical note. J. Rehab. Res. Dev. **33**(3), 267–278 (1996)
4. Kurt, O., Jonathan, K., Arne, K., Billy, L., Goran, S.: The CAPOD system a scandinavian CAD/CAM System for prosthetic sockets. J. Prosthet. Orthot. **1**(3), 139–148 (1989)
5. Lin, Y.P., Wang, C.T., Dai, K.R.: Reverse engineering in CAD model reconstruction of customized artificial joint. J. Med. Eng. Phys. Elsevier **27**(2), 189–193 (2005)
6. Shuxian, Z., Wanhua, Z., Bingheng, L.: 3D reconstruction of the structure of a residual limb for customising the design of a prosthetic socket. J. Med. Eng. Phys. Elsevier **27**(1), 67–74 (2005)

7. Sinha, A.N., Udai, A.D.: Development of artificial limb surface using reverse engineering method. In: Proceedings of International Conference on Advances in Machine Design & Industry Automation (ICAMDIA 2007), College of Engineering, Pune, India, pp. 519–522. http://web.iitd.ernet.in/~mez108088/files/sinhai_udai_icamdia_2007.pdf
8. Sinha, A.N., Udai, A.D.: Computer Graphics, 1st edn. Tata Mc-Graw Hill Publishing Co., Ltd., New Delhi, India (2007)
9. Walsh, N.E., Lancaster, J.L., Faulkner, V.W., Rogers, W.E.: A computerized system to manufacture prostheses for amputees in developing countries. J. Prosthet. Orthot. **1**(3), 165–181 (1989)

Quality of Services-Aware Routing Protocol for Mobile Ad Hoc Networks for Multimedia Communication

Hiren Kumar Deva Sarma

Abstract A routing protocol for mobile ad hoc networks with Quality of Service consideration is proposed in this paper. The protocol ensures user-specified Quality of Service level. The proposed protocol is intended for multimedia communication (e.g., video) over mobile ad hoc networks. The protocol discovers multiple paths between a pair of source and destination node by using existing routing protocols like multi-path routing protocol based on Ad hoc On demand Distance Vector (AODV) or Dynamic Source Routing (DSR). Then discovered paths are prioritized as per the proposed protocol. The most suitable path for ensuring user-specified Quality of Service level is selected for further data transmission. The analytical results presented in the paper show the performance of the proposed protocol.

Keywords Mobile ad hoc networks • QoS-aware routing • Multimedia communication • Multi-path routing

1 Introduction

Mobile ad hoc networking is one of the latest paradigms in the mobile wireless communication world. Mobile ad hoc networks are formed by a set of mobile nodes that are capable of communicating with each other through wireless medium like radio. Mobile ad hoc networks are self-organized and self-starting in nature. Wireless interconnections of communication devices happen without any centralized authority. Such networks either extend or operate along with the wired networking infrastructure and depending on applications these may evolve as autonomous networks as well. Mobile ad hoc networks suffer from different constraints such as limited computing resources of the devices, limited communication bandwidth in the network, and also limited battery power which is the source of

H.K.D. Sarma (✉)
Department of Information Technology, Sikkim Manipal Institute of Technology,
Majitar, Rangpo, East Sikkim 737 136, Sikkim, India
e-mail: hirenkdsarma@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_37

energy of the communicating devices in the network. It also may not be feasible to renew the battery of a device due to harsh environment. Therefore, energy efficiency is one of the major objectives considered at the time of the protocol design for such networks. Mobile ad hoc networks have potential to apply in different areas like civil life, industries, agriculture, post disaster recovery operations and also military and defense. Multimedia communication over mobile ad hoc networks is a challenging task. There shall be many applications which may demand multimedia, e.g., video transport over the network. Since there is no fixed infrastructure and the topology is frequently changing due to node mobility, the links are continuously established and broken. Multimedia (video) transport typically needs stringent bandwidth and delay guarantee [1, 2]. But it is extremely hard in mobile ad hoc networks to maintain some specific end-to-end route which is both stable and bandwidth sufficient. In spite of all these constraints, it is essential to maintain certain level of Quality of Service (QoS) while multimedia transport is under consideration over a mobile ad hoc network. Quality of Service provisioning over mobile ad hoc networks remains an open issue. Traditional Quality of Service management techniques cannot be applied directly to mobile ad hoc networks due to the several constraints such networks suffer from. It is also important to mention that Quality of Service provisions in a network are not dependent only on one layer like network layer, rather they depend on all the layers in the protocol stack. Improved Quality of Service can be offered by a joint and coordinated effort of all or multiple layers.

In this work, a novel routing protocol for mobile ad doc networks is developed. The case of multimedia communication over mobile ad hoc networks is considered. The proposed protocol is energy efficient and also Quality of Service aware. The protocol tries to optimize the energy utilization in the network in the process of ensuring user specified Quality of Service parameters while multimedia (e.g., video) communication takes place. The mathematical model proposed here tries to optimize the resource utilization in the network considering the constraints in a given situation. Some preliminary analytical results are also reported and analyzed. Future scopes of the work are outlined. The rest of the paper is organized as follows: Sect. 2 highlights the background of this work followed by Sect. 3, in which related works are outlined. The protocol is proposed in Sect. 4. Section 5 presents some analytical results and associated discussions. Finally, the paper is concluded in Sect. 6.

2 Background

Mobile ad hoc networking is gradually becoming mature. However, challenges still exist for supporting multimedia data transportation, for example, video communication over mobile ad hoc networks. Video communication is a kind of soft real time application in which correctness with respect to the data transmission like task execution does not only depend on the delivery of the data at the destination node, but also on the time at which the data packets have been delivered. Timely delivery of data packets at the destination node is essential in order to meet certain level of Quality of Service by an application like video communication. Overall Quality of Service of an application is determined by several parameters like time-bound delivery of the packets of a video stream at the destination, high throughput in terms of successful packet delivery against total number of packets transmitted by the sender node, etc. The topology of mobile ad hoc network is highly dynamic and the links between nodes may break up abruptly. This may affect the video transmission and may cause interruptions and jerkiness in the video signal in the receiver end. The quality of the received video signals may get also affected by the high packet loss during transmission through a multi-hop route. Again transmission bit error rate is very high in mobile ad hoc networks due to channel fading and interference from other transmissions. Multimedia communication demands high communication bandwidth and also cannot tolerate delay. In the event of video transmission over mobile ad hoc networks, the stream of frames must flow through the network in a steady manner. It is desirable to have a uniform level of jitter. The term jitter signifies the delays between consecutive frames. Jitter should not vary too much during a session of video transmission. Jitter above a threshold value in a particular flow may result in a frame at being available at the receiver after the previous frame has been played out. Thus this phenomenon may lead to impairments in video quality [2]. Considering all the above observations and the resource constraints mobile ad hoc networks suffer from, it is concluded that ensuring user specified Quality of Service in mobile ad hoc networks is a highly challenging task.

3 Related Work

Video transport over wireless ad hoc networks has been studied in [3–5]. These techniques exploit the presence of multiple paths between a pair of sender and receiver in a mobile ad hoc network and try to distribute the video traffic through these multiple paths. Dynamic Source Routing is an on demand source routing protocol for mobile ad hoc networks [1]. It obtains multiple paths between a pair of source and destination. Split Multi path Routing (SMR) is an extension to DSR. In [3], a cros- layer QoS provisioning algorithm is proposed. This scheme also makes use of the multi-path characteristics of DSR protocol in order to provide multiple source-to-destination node-disjoint and also loop-free paths. In [4] an optimization over on-demand routing protocol is proposed to facilitate video transmission over mobile ad hoc networks. The protocol tries to reduce the routing overhead by limiting the route discovery process confined to a reasonable region. Thus it minimizes the flow of control packets in the network and available bandwidth is saved to some extent. It tries to improve the real time video transmission quality. The proposed protocol constructs an overlay by using Multicast Overlay Spanning Tree (MOST) protocol. The solution also creates multiple video description of a given video stream by making use of Multiple Description Coding (MDC).

Step 1: User specified input regarding Quality of Servicelevel
Step 2: Discover multiple paths in between a source-destination pair
Step 3: Compute Reliability matrix for each route
Step 4: Compute Mobility matrix for each route
Step 5: Compute Energy matrix for each route
Step 6: Prioritize the Quality of Service satisfying routesfor use

Fig. 1 Components of the proposed QoS-aware routing protocol for multimedia communication

4 Proposed Algorithm

The objective is to ensure Quality of Service specified by users optimizing the energy utilization in the mobile ad hoc network. The Quality of Service parameters are specified by users. The parameters are *minimum bandwidth expected*, *minimum life time of the network*, *maximum delay that can be tolerated*, *maximum jitter that can be tolerated*, and *maximum packet loss percentage that can be accepted*. The protocol tries to ensure the user-specified Quality of Service level. Multiple paths are discovered between a pair of source and destination nodes. Each path is characterized by three different metrics namely *Reliability matrix*, *Mobility matrix* and *Energy matrix*. Figure 1 shows various components of the protocol in sequence.

Reliability matrix gives an estimation regarding the probability of transmission error of packets in a route. Mobility matrix gives an idea about the mobility of the nodes in a route and the stability of the route. Energy matrix gives estimation in regard to the available energy in a route. In order to meet the specified Quality of Service level, the protocol finds the suitable routes based on characteristics specified by Reliability matrix, Mobility matrix and Energy matrix. Multiple paths between a pair of source destination nodes which are discovered before are now prioritized and used alternately such that the overall life of the network is extended.

(a) **User-specified input regarding Quality of Service level**: User of the system specifies various Quality of Service parameters. These parameters are minimum bandwidth expected (W_{min}), minimum lifetime of the network (T_{min}), maximum delay that can be tolerated ($\mathrm{delay}_{\mathrm{max}}$), maximum jitter that can be tolerated ($\mathrm{jitter}_{\mathrm{max}}$), and maximum packet loss percentage that can be tolerated ($\mathrm{Ploss}_{\mathrm{max}}$). Minimum lifetime of the network indicates the minimum time users want the system to run. Again this minimum lifetime, T_{min}, is dependent on a threshold number of nodes ($N_{\mathrm{threshold}}$) that have to remain alive at least till the end of $\mathrm{T}_{\mathrm{min}}$, for successful operation of the network. Depending on the user specified $T_{\min}$, the value of $N_{\text{threshold}}$ can be determined. For a given setup of mobile ad hoc network, this mapping $T_{\min} \rightarrow N_{\text{threshold}}$, can be determined by simulation. Thus the routing protocol must be energy efficient and life time of the network (in terms of number of active nodes) must be prolonged. Therefore, user-specified QoS requirement may be expressed as:

$$\text{Requirement}_{\text{QoS}} = \{W_{\min},\ T_{\min},\ \text{delay}_{\max},\ \text{jitter}_{\max},\ \text{Ploss}_{\max}\} \tag{1}$$

The above-mentioned QoS parameters are related to some other parameters of the network setup, e.g., reliability of the routes discovered by the routing protocol, mobility of the nodes in the network, and also energy availability in the nodes of the network. Reliability of a route is influenced by the percentage of packet loss in that route and also it is an indication of stability of the route in terms of links' availability. Therefore, Reliability matrix, Mobility matrix and Energy matrix with respect to a network setup are computed. The procedures for computing Reliability matrix, Mobility matrix and Energy matrix are discussed in detail in the following part of this paper.

(b) **Discover multiple paths in between a source–destination pair**: There are several routing protocols in literature that can discover multiple paths in a mobile ad hoc network setup [2]. Such protocols are generally extensions of well-established routing protocol like AODV [2] or DSR [2]. In this proposed protocol, multiple paths between a specific source–destination pair are assumed to be discovered by some existing energy efficient multi-path routing protocol. Then processing is done on the discovered path set in order to find out QoS aware path to ensure Quality of Service specified by users.

(c) **Computing reliability matrix**: A Reliability matrix for each of the routes discovered is computed. Each route is consisting of multiple links connecting different nodes. Each link connects two nodes. First a *reliability parameter* for each link is computed. Various parameters considered for computing reliability of a link connecting node i and j are:

the probability that a link is bad → $p(x)$,
link failure probability → $(p_{\text{link}})_{i,j}$,
node failure probability of node i → $(p_{\text{node}})_i$,
node failure probability of node j → $(p_{\text{node}})_j$.

The *reliability parameter* ($P_{\text{reliability}}$) of each link is computed in terms of cumulative probability considering the above-mentioned four probabilities. This cumulative probability tells the probability that at least one of the four events happens (i.e., node failure i, node failure j, link failure $i \rightarrow j$, bad link $i \rightarrow j$). Thus the cumulative probability regarding reliability of a route is computed as mentioned below:

Table 1 Reliability matrix for various links (e.g., $i \rightarrow j$, etc.) in a Route R

i	j	–	k
(–, –)	$((P_{\text{reliability}})_{i \to j}, \text{SINR}_{i \to j})$		$((P_{\text{reliability}})_{i \to k}, \text{SINR}_{i \to k})$
$((P_{\text{reliability}})_{j \to i}, \text{SINR}_{j \to i})$	(–, –)		$((P_{\text{reliability}})_{j \to k}, \text{SINR}_{j \to k})$
		(–, –)	
$((P_{\text{reliability}})_{k \to i}, \text{SINR}_{k \to i})$	$((P_{\text{reliability}})_{k \to j}, \text{SINR}_{k \to j})$		(–, –)

$$\left(P_{\text{reliability}}\right)_{i \to j} = 1 - (1 - p(x))\left(1 - (p_{\text{link}})_{i,j}\right)\left(1 - (p_{\text{node}})_i\right)\left(1 - (p_{\text{node}})_j\right) \tag{2}$$

Then another parameter, SINR of each link in the route, (e.g., $\text{SINR}_{i \to j}$, etc.) is considered and following matrix is formed (as shown in Table 1).

The above matrix gives the reliability parameter and SINR of various links in route R. Now in order to select a particular route with QoS assurance, the reliability matrix is studied and only those links and therefore, only those routes are selected which possess the reliability value above a threshold value. For example, the link $i \rightarrow j$, will be selected only if the values of the pair $((P_{\text{reliability}})_{i \to j}, \text{SINR}_{i \to j})$ are above a threshold value. This threshold value may be determined by simulation and it also depends on the application and QoS requirement level.

(d) **Computing mobility matrix**: Mobility of nodes has significant impact on the Quality of Service. Relatively less mobile nodes are preferred in a route. Such a work to select relatively less mobile nodes for a route is available in [5]. This model uses received signal strength as the basis for computing relative mobility matrix. Again in [6], this relative mobility model is used for finding relatively less mobile nodes and classifying the nodes as motionless, low mobility, high mobility, and very high mobility. We use this part of the work reported in [6] as the mobility matrix.

(e) **Computing energy matrix**: The available energy in a route should be sufficient in order to have smooth routing of packets through that route. If sufficient energy is not available in the intermediate nodes of a route, then these nodes may become down even at the middle of a particular transmission session. Such instability is not desired and therefore, before identifying a route the residual energy levels of the intermediate nodes are analyzed.

Table 2 describes different routes available between a pair of nodes. For example, there are three routes available between N_1 and N_2 as per Table 2 and those are R_i, R_j and R_k. Various entries in the table shows the residual energy level of different routes between a selected pair of nodes as source and destination, respectively.

For each route R_i, between a pair of nodes (N_x, N_y), where $1 \leq i \leq n$ and n is the maximum number of unique routes between $(N_x, N_y) \rightarrow R_i(E_{x,y})$ is the residual energy of the route R_i between the nodes N_x and N_y. Here, $R_i(E_{x,y})$ is

Table 2 Energy matrix showing residual energy level of different routes

	N_1	N_2	–	–	N_n
N_1	0	$R_i(E_{1,2})$, $R_j(E_{1,2})$, $R_k(E_{1,2})$	–	–	$R_i(E_{1,n})$
N_2	$R_i(E_{2,1})$	0	–	–	$R_i(E_{2,n})$
–	–	–	–	–	–
–	–	–	–	–	–
N_n	$R_i(E_{n,1})$	$R_i(E_{n,2})$	–	–	0

the total residual energy of all the intermediate nodes of the route R_i between N_x and N_y.

Now issue is which route to select if multiple routes are available between a pair of source and destination. Following computation is done.

Consider that the total energy required to transmit a bit from source to destination through route R_i is equal to $(E_i)_{\text{req}}$. Also consider that the total energy available (residual) in the route $R_i = (E_i)_{\text{res}} = R_i(E_{x,y})$. Then compute,

$$E = (E_i)_{\text{req}} / (E_i)_{\text{res}} \tag{3}$$

Now, the route with minimum value of E is selected as the appropriate route for data transmission.

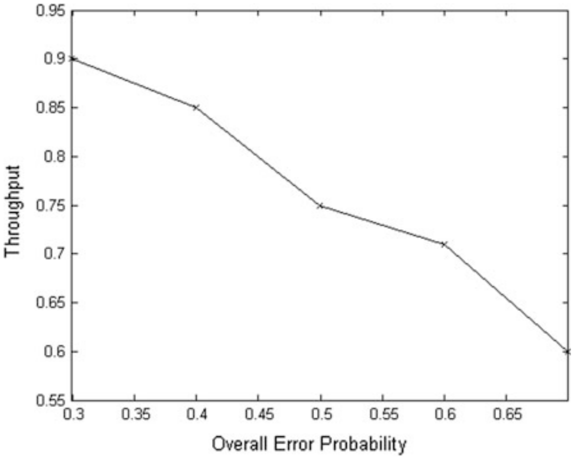
(f) **Prioritize the Quality of Service satisfying routes for use**: After all the previous steps suitable routes satisfying QoS requirements are identified. Then the routes are prioritized depending on the user requirements, kinds of user application undertaken and the quality levels of the available routes. This prioritization is postponed till the implementation phase. The prioritization can be done based on the situation and the final selection of the most useful route for a particular situation it is left for the implementer. How to do it, is simple process. For example, different available routes satisfying QoS requirements as mentioned in earlier steps are enlisted. Then considering steps (a), (c), (d) and (e), a normalization process (by using a suitable normalization function) can be carried out since the parameters considered in these steps are of heterogeneous nature. And finally most appropriate route can be selected.

5 Results and Discussion

The performance of the proposed protocol is discussed through some analytical results as given in this section.

The proposed protocol is simulated in MATLAB using the analytical model described in this paper. The results obtained here depict the performance of the protocol under different circumstances.

Fig. 2 Throughput level under different error conditions



A mobile ad hoc network of 20 nodes is considered. Each node moves with different velocity level. The throughput in terms of number of packets delivered against the number of packets transmitted is computed after considering different error probabilities like link error probability, node error probability and bad link probability. The throughput indicates the ration between the numbers of packets delivered at the destination to the actual number of packets sent by the sender. This throughput also indicates the number of packets delivered by meeting user-specified QoS level.

Figure 2 shows the throughput analysis of the proposed protocol under different error condition. It is observed that with increase in the error probability, there is degradation in the throughput level. This increased error probability indicates higher probability of link failure, node failure, and bad link.

The throughput level degrades along with increase in the mobility level. This fact is demonstrated in Fig. 3. Five mobility levels are shown and these are mobility level 1, 2, 3, 4, and 5. Mobility level 1 indicates nodes move with a velocity in the range of (1–2) m/s. Similarly, mobility level 2 indicates a velocity in the range of (3–4) m/s, mobility level 3 indicates a velocity level in the range of (5–6) m/s, mobility level 4 indicates a velocity level in the range of (7–8) m/s and mobility level 5 indicates a velocity level in the range of (9–10) m/s.

Figure 4 shows average length of the path finally chosen between a pair of source and destination nodes. Number of intermediate hops is the measure of the length of the path. In this situation, the mobile nodes move with different velocities such as 1, 2, 3, 4, and 5 m/s. Moreover, the number of nodes considered is 20. It is observed that the average path length increases along with the increased mobility level; although the average length of the path at the speeds of 2 m/s and 4 m/s is same. This is mainly due to the random breakage of the links because of the high mobility of the nodes.

Fig. 3 Throughput level under different mobility conditions

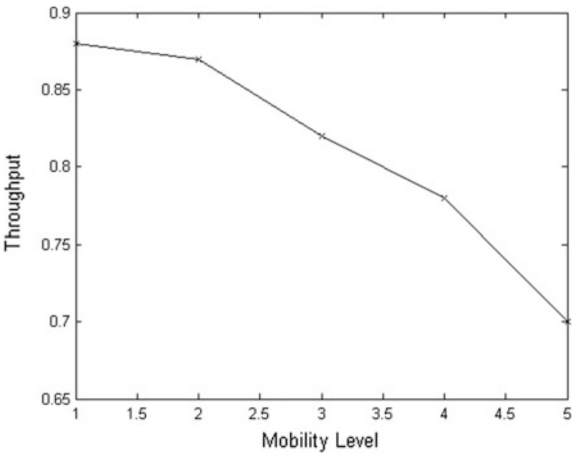


Fig. 4 Average path length under different mobility conditions

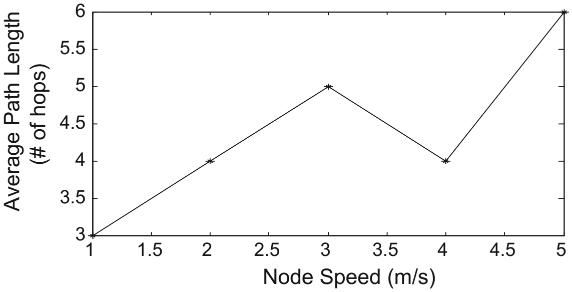


Fig. 5 Average path length in different sizes of network

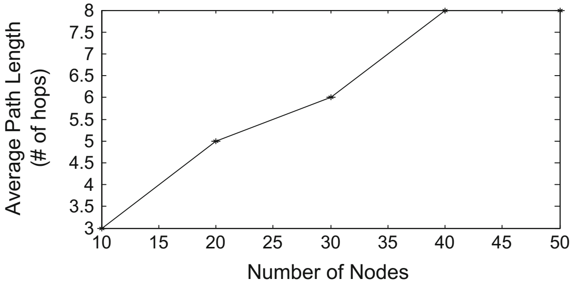


Figure 5 shows the average path length between a pair of sender and receiver in different networks of sizes 10, 20, 30, 40 and 50 nodes. The average length of the path either increases or remains the same along with the increased size of the network. This is so, because with higher number of nodes more links are available between a pair of sender–receiver.

6 Future Scope and Conclusion

In future, test bed implementations can be planned for observing the performance of the protocol. The proposed protocol tries to ensure the user-specified Quality of Service level in a mobile ad hoc network. This kind of protocol is more suitable when multimedia transmission, e.g., video communication, over mobile ad hoc network is considered. Multiple paths are found out between a pair of source and destination by using the well-known multi-path routing protocols based on AODV or DSR. Then discovered multiple paths are prioritized as per proposed protocol. Then most suitable path that can ensure user specified Quality of Service level is selected for further data communication. The analytical results demonstrate the performance of the proposed protocol in terms of throughput.

Acknowledgments This work is carried out as a part of the research grant issued by All India Council for Technical Education (AICTE), Government of India, with file number-8023/BOR/RID/RPS-216/2007-08. The authors acknowledge the support of AICTE, Government of India.

References

1. Siva Ram Murthym, C., Manoj, B.S.: Ad hoc Wireless Networks: Architectures and Protocols, 2nd edn. Pearson Education (2004). ISBN: 81-297-0945-7
2. Leon Garcia, Widjaja: Communication Networks. Tata McGraw Hill edition (2000). ISBN: 0-07-040235-3
3. Adam, G., Bouras, C., Gkamas, A., Kapoulas, V., Kioumourtzis, G., Tavoularis, N.: Cross-layer mechanism for efficient video transmission over mobile ad hoc networks. In: Third International Workshop on Cross Layer Design, 2011 (IWCLD 2011)
4. He, Y., Onyang, Y., Li, C., Xiong, Z.: Routing optimized video transmission over mobile ad hoc networks. In: Proceedings of Workshop on Power Electronics and Intelligent Transportation System, 2008, (PEITS'08), 2–3 Aug 2008, pp. 18–22. ISBN: 978-0-7695-3342-1
5. Basu, P., Khan, N., Little, T.D.C.: A mobility based metric for clustering in mobile ad hoc networks. In: Proceedings of 21st International Conference on Distributed Computing Systems Workshop (ICDCSW'01), pp. 413–418 (2001)
6. Carrascal Frias, V., Diaz Delgado, G., AguilerIgartua, M., Alins Delgado, J., Mata Diaz, J.: QoS provision for video-streaming applications over ad hoc networks. In: Proceedings of IEEE EUROCON 2005, held at Serbia & Motenegro, Belgrade, 22–24 Nov 2005, pp. 640–643

Collectively Find Spatial Objects in Time-Dependent Spatial Network

Abdul Wahid and Md. Tanwir Uddin Haider

Abstract With the rapid revolution of GPS-equipped devices, discovery of spatial objects with user's requirement in a city that takes minimum travelling time is a very big challenge. To retrieve spatial objects, we use some spatial queries that have both textual description and location of objects. All existing queries in spatial network focus on finding single objects rather than group of objects, and also assume that each edge of spatial network has constant weight. But in real-world scenario weight of each edge varies and depends on various factor for, e.g., congestion, road blocking, etc. In this paper, we propose a query that collectively finds a group of spatial objects in such a way that group's keyword matches with query's keyword and sum of their traveling time is minimum using objects near to the query location and also has lowest inter-object distance with considering the lowest congestion in their path. The aim of this query is to help users in application like emergency services, trip planning, etc.

Keywords Spatial database • Spatial query • Spatial objects • Time-dependent spatial network

1 Introduction

The spatial database management system is a database management system that handles spatial data as well as non-spatial data. In many fields, there is need of spatial data. Spatial data mean data related to space and space may be any point of interest like tourist point, hospital, restaurant, mall, etc. To store spatial data in computer system, we use points, line, segments and region. In spatial database,

A. Wahid (✉) · Md.T.U. Haider
Department of Computer Science & Engineering,
National Institute of Technology Patna, Patna 800 005, Bihar, India
e-mail: awahid.nitp@gmail.com

Md.T.U. Haider
e-mail: tanwir99@yahoo.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_38

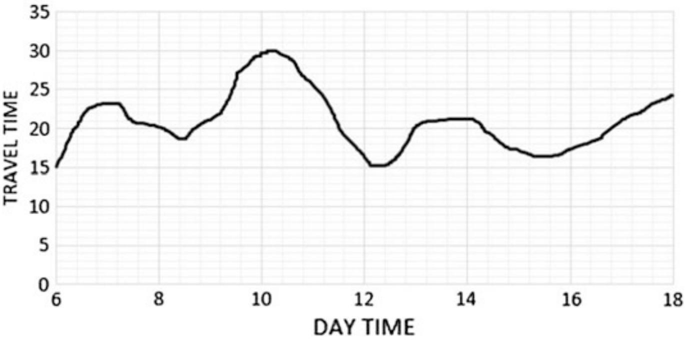


Fig. 1 Actual travel time of a segment of road network

there are various types of spatial query that handle spatial data. Spatial database is widely used in GIS (Geographic Information System), CAD (Computer-aided design), MMIS (multimedia information system and many more fields).

Rapid revolution of GPS-equipped devices (like smart phones) and online map based services (like Google Maps, Bing Maps etc.) allow users to find location based services, points of interests (for e.g. Hospitals, hotels, shopping malls etc.) and their directions with their travelling time. There are various algorithms (e.g., [1–3]) that compute these types of query.

But most of these algorithms assume that each edge of road network has constant weight (traveling time), while in real-world scenario, weight (i.e., traveling time) of each path of a road network depends on various factor for, e.g., congestion on that edge, road blocking, etc., and it is a function of the time of the day. It means that travel time depends on the time at which the query is issued.

In Fig. 1, graph shows the actual travel time (in min) of an edge of a road network between 6:00 A.M. to 6:00 P.M. From Fig. 1, we observe that travel time significantly changes in very short duration of time. In Fig. 1 from 9:00 A.M. to 10:00 A.M. travel time changes from 22 to 30 min and from 10:00 A.M. to 12:00 P. M. travel time changes from 30 to 16 min approximately. Hence, our query returns different result at different time and it mainly depends on the time at which the query is issued.

Figures 2 and 3 show a road network where an ambulance issues a query at 6:00 A.M. and 10:00 A.M., respectively, on same path of road network on the same day. And it shows that travel time on the path of a same road network is different and in both cases query would return different outputs.

After observing user needs, we present here a query that collectively meets the user requirement. For example, a person may have multiple particular point of interests like—a hospital (where a number of beds available are more than 100, and specialist in orthopedics), restaurant (where special menu available), and accommodation (with some condition according to user's requirement) that may be met by several spatial objects.

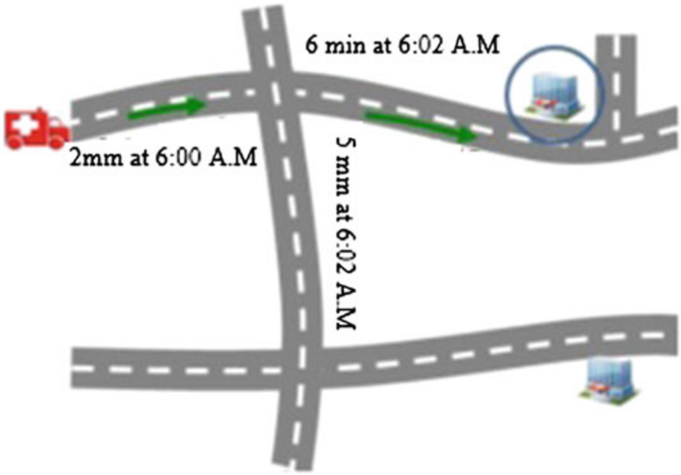


Fig. 2 Query at 6:00 A.M.

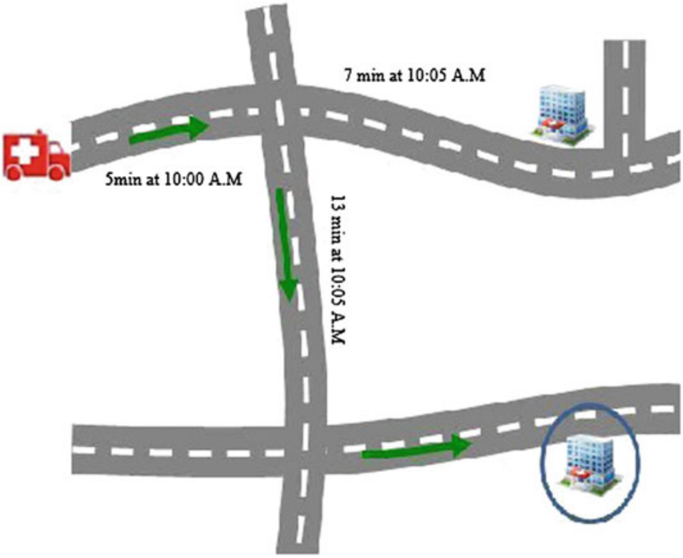


Fig. 3 Query at 10:00 A.M.

To solve this type of user's need, we have a spatial database that stores spatial as well as non-spatial data and then considers how to retrieve group of spatial objects that collectively meet the user's requirement as location and set of keywords [4] in such a way that (i) groups' keyword matches with query's keyword, (ii) the distance of spatial objects should be minimum from query point and also have less congestion in their path that minimizes the traveling time and (iii) inter-objects distance should be minimum (i.e., spatial objects or point of interest close to each other) also with less congestion in their specified path.

Consider a set of objects O, and a query q = (q.l, q.k), where ‘q.l’ is a location and ‘q.k’ represents a group of keywords. Now our aim is to find a group of spatial objects o (o is subset of O) that match with query’s keyword, such that the sum of their traveling time should be minimum. To solve this problem, first we solve the problem of finding single spatial object (like mall or hospital or any point of interest), whose distance from query point ‘q’ is minimum and has less congestion in their path. After selecting one region of interest, then we move to find second region of interest whose distance should be minimum from both query point ‘q’ and first point of interest also with less congestion in their path (1st path from query point to 2nd region of interest and 2nd path from first region of interest to 2nd region of interest) and repeat this process to find next spatial objects.

The rest of the paper is as follows. Section 2 presents related work. Section 3 describes problem definition. Section 4 describes processing of type 1 spatial query. Section 5 describes processing of type 2 spatial query. Section 6 shows analytical results of type 1 and type 2 spatial query. Section 7 presents conclusion and future work.

2 Related Work

In spatial query processing, spatial database uses an indexing to prune the search space. To achieve high efficiency during query processing, various index structures like R-tree [5], kd-tree [6] and quad-tree [7] have been developed. In the nearest neighbor query, its aims to find objects near to the query point. Nearest neighbor (NN) query may process in Euclidean space [8], in spatial network [9], and in higher dimensional spaces (e.g., [10, 11]).

We have studied various types of NN search query like Aggregate k-NN monitoring [12], Trip planning queries [13], and Continuous NN queries [8, 10, 14, 15]. Where CNN query continuously reports the k-NN while user moving along the path. Mouratidis et al. [16] investigate a Continuous Nearest Neighbor (CNN) problem in the road network where both query point and data’s objects change their location. In [17], Cong presents retrieval of top k most relevant spatial object that matches with query.

The mCK query [18, 19] retrieves group of spatial objects where mCK query takes m keywords and returns m object that matches with query keywords. Here, it is assumed that each spatial object corresponds to a unique keyword of query. But in this paper, we take a query that retrieves a group of spatial objects that have both location and set of keywords which are different from mCK query.

3 Problem Definition

In this paper, we assume that ‘S’ be a set of keywords. This keyword takes user input and depends on type of the applications or user’s requirement. It may be number of beds available in hospital, department of hospital should be orthopedics, or menu in the restaurant, etc. Let ‘D’ be a database that contains ‘N’ spatial objects and each object ‘O’ which belongs to database ‘D’ contain location ‘O.l’ and group of keywords ‘O.k’. Where ‘O.k’ is the subset of ‘S’.

Consider a spatial query q = <q.l, q.k>, where ‘q.l’ represents location and ‘q.k’ represents group of keywords. This spatial query finds a group of objects ‘X’ (where ‘X’ is the subset of ‘D’) in such a manner that to cover all the spatial objects or point of interest in minimum time, i.e., ‘f(X)’ should be minimum.

We present the total traveling time function as follows:

$$f(X) = \beta \cdot d1(q, X)/v1 + (1 - \beta) \cdot d2(X)/v2. \quad (1)$$

where d1(q, X) represents network distance between query point ‘q’ and object in X, and v1 is the average speed between query point ‘q’ and object in X on the road network. d2(X) represents network distance between objects in X, (i.e., object should be near to each other), and v2 denotes the speed at their specified path between objects in road network. Speed v1 and v2 vary according to the congestion in their specified path. β is a real value between 0 and 1. To minimize the traveling time ‘f(X)’ should be minimum. We consider two types of function.

TYPE 1 traveling time function:

$$f1(X) = \beta \cdot \max_{(r \in x)} d1(q, r)/v + (1 - \beta) \cdot \max_{(r1, r2 \in x)} \cdot d2(X)/v. \quad (2)$$

In Type 1 time traveling cost function, calculate maximum time to travel all the spatial objects or point of interest. First part calculates maximum time to travel from query point to any object in X and second part calculates maximum time to travel between objects in X, without considering any congestion in their path and ‘v’ is the free flow speed.

Type 2 time traveling cost function:

$$f2(X) = \beta \cdot \max_{(r \in x)} d1(q, r)/v1 + (1 - \beta) \cdot \max_{(r1, r2 \in x)} \cdot d2(X)/v2. \quad (3)$$

In Type 2 time traveling cost function, calculate maximum time to travel all the spatial objects or region of interest. First part calculates maximum time to travel from query point to any object in X and second part calculates maximum time to travel between objects in X, with considering congestion in their path. Where v1 and v2 vary according to the congestion in their specified path.

4 Processing of Type 1 Spatial Query

To process type 1 spatial query where query finds the set of objects that are near to the query location and also close to other spatial object, such that time to travel all the spatial objects or point of interest should be minimum. To calculate time to travel all point of interest in this query, we assume that there is no congestion in their path that means user can travel with speed of ‘v’ on each path of a road network.

To process this query, we mainly focus on:

1. Find set of objects (m) that cover the query keywords from query location.
2. After getting the set of spatial objects

(a) if set of spatial object contains only one type of object (e) (like hospital or restaurant, or mall), then minimum traveling time depends only on minimum distance from query point, because we assume that its speed will remain constant because there is no congestion or other disturbance in their path. Traveling time (q, point e) = min Distance (q, point e)/v. (i.e., sum of network distance of all paths of road network from query point ‘q’ to spatial object divided by speed of vehicle on that path).

(b) if set of spatial objects (m) contains more than one type of objects, then traveling time depends on two factors: (i) distance from query point should be minimum and (ii) objects should be close to each other (i.e., inter-object distance should be minimum).

5 Processing of Type 2 Spatial Query

To process type 2 spatial query where query finds the set of objects that are near to the query location and also closer to other spatial object, such that the time to travel all the spatial objects or point of interest should be minimum. To calculate time to travel all regions of interest in this query, we assume that there is congestion in their path that affects the speed of vehicle on those paths of a road network. Here, we take the membership value of their specified path in congestion (μ). Due to congestion, speed changes from the previous value as:

$$V_{(\text{new})} = V_{(\text{old})}\{1 - \mu\} \tag{4}$$

where μ is the membership value of path in congestion at the time of query.

To process this query, we mainly focus on:

1. Find set of objects (m) that cover the query keywords from query location.
2. After getting the set of spatial objects

(a) if set of spatial object contains only one type of object (e) (like hospital or restaurant, or mall), then minimum traveling time depends on both (i) minimum distance from query point, (ii) less congestion in their path. Due to congestion, the time to travel all the point of interest increases. But our aim is to cover all the points in less traveling time. So that is why we consider only those points whose distance from the query point is less and those paths that have less congestion in the road network.

(b) if set of spatial objects (m) contains more than one type of objects, then traveling time depends on two factors: (i) distance from query point should be minimum with considering congestion in their path and (ii) objects should be close to each other (i.e., inter-object distance should be minimum) also with less congestion in their path.

6 Analytical Result

Consider a query where user wants to cover all spatial objects (like mall, hospital and restaurant) where hospital has at least 100 beds and whose specialty in orthopedics, and restaurant should be veg restaurant, and a shopping mall from query point ‘q’ with less traveling time.

6.1 *Type 1 Spatial Query*

To process this type of query, first we find all the spatial objects in a city that cover the query keywords. In this section, we assume that the speed (assume V km/h) remains same in their path and does not change due to any congestion.

Time to cover all the points is equal to the total network distance divided by speed of vehicle on each path of the road network. So here, the time to travel all the points depends only on distance between them, because we have already assumed that the speed of vehicle on each path of a road network is constant.

To process this, we use the dataset of ‘hospital’, ‘restaurant’, and ‘mall’. After applying query on the hospital dataset (i.e. hospital/s that have at-least 100 beds and whose specialty should be orthopedics in a city C from query point q) we got some hospitals which is shown in Table 1.

After getting the list of hospitals, we apply the query on dataset of restaurant that provides a list of restaurants which is shown in Table 2 that cover the query keywords.

Table 1 Hospitals in a city

Hospital	H1	H2	H3	H4	H5	H6
Network distance from ‘q’ (km)	0.92	0.8	0.7	0.4	0.8	0.6

Table 2 Restaurants in a city

Restaurant	R1	R2	R3	R4	R5
Network distance from ‘q’ (km)	0.65	0.75	0.9	0.85	0.4

Table 3 Malls in a city

Mall	M1	M2	M3	M4	M5	M6
Network distance from ‘q’ (km)	0.7	0.3	0.46	0.9	0.9	0.2

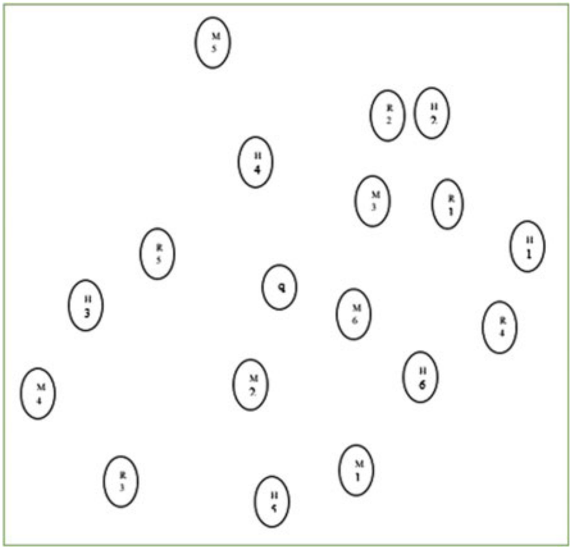


Fig. 4 All spatial objects in a city

Table 3 shows some malls which cover the query keywords. All spatial objects that satisfy the query are shown in Fig. 4.

Now, we proceed to calculate the network distance of each spatial object from query point q. After calculating network distance, we found that ‘M6’ (M6 are spatial object of mall) is close to query point. After selecting 1st object (i.e., mall), now we move to find other objects whose distance from query point ‘q’ and 1st spatial object should be minimum. Then now, we calculate network distance of each object of hospital and restaurant from ‘M6’. Tables 4 and 5 show network distance from ‘M6’.

After calculating distance of each object of hospital and restaurant from query point q and 1st spatial object ‘M6’, we found that spatial object ‘H6’ is close to both query point ‘q’ and spatial object ‘M6’.

To find third object (restaurant in this example) that should be near to the query point, 1st spatial object ‘M6’ and 2nd spatial object ‘H6’ we calculate network

Table 4 Hospital dataset

Hospital	H1	H2	H3	H4	H5	H6
Network distance from 'M6' (km)	0.74	0.79	1.1	0.6	0.6	0.37

Table 5 Restaurant dataset

Restaurant	R1	R2	R3	R4	R5
Network distance from 'M6' (km)	0.55	0.78	1.3	0.55	0.62

Table 6 Dataset of restaurant (distance from H6)

Restaurant	R1	R2	R3	R4	R5
Network distance from 'H6' (km)	0.62	1.4	1.2	0.2	1.3

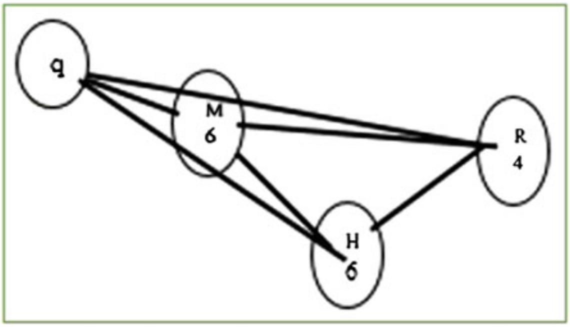


Fig. 5 Three spatial objects (hospital, mall, and restaurant) that cover the query keywords

distance of each 'restaurant' from point 'M6' and 'H6'. Table 6 shows network distance of each object of restaurant (or restaurant) from 'H6'.

After overall calculation, we found third object 'R4' which is shown in Fig. 5. Figure 5 shows set of spatial objects that cover the query keywords and connecting lines show that there is a path that connects spatial objects.

6.2 Type 2 Spatial Query

Consider a typical type of query where user wants to cover all spatial objects with considering the specification of these objects (like spatial objects are mall, hospital and restaurant where user wants such type of hospitals whose specialty should be orthopedics and have at-least 100 beds available and next try for restaurant that should be vegetable, and a mall within 1 km range from query point 'q' with less travelling time.

To process this type of query first, we find all spatial objects that cover the query keywords. In this section, speed varies due to congestion in their path from $V_{(old)}$ to $V_{(new)}$ as

Table 7 Hospitals in a city with congestion in their path from query point ‘q’

Hospital	H1	H2	H3	H4	H5	H6
Network distance from ‘q’ (km)	0.92	0.8	0.7	0.4	0.8	0.6
Value of μ from q	0.2	0.8	0.1	0.8	0.6	0.7

Table 8 Restaurants in a city with congestion in their path from ‘q’

Restaurant	R1	R2	R3	R4	R5
Network distance from ‘q’ (km)	0.65	0.75	0.9	0.85	0.4
Value of μ from q	0.3	0.2	0.2	0.8	0.7

Table 9 Malls in a city with congestion in their path from ‘q’

Mall	M1	M2	M3	M4	M5	M6
Network distance from ‘q’ (km)	0.7	0.3	0.46	0.9	0.9	0.2
Value of μ from q	0.4	0.8	0.7	0.2	0.3	0.8

$$V_{(new)} = V_{(old)}\{1 - \mu\} \tag{5}$$

where μ is the membership value of path in congestion at the time of query.

Assume $V_{(old)} = V$ km/h.

Time to cover all the points = distance/$V_{(new)}$.

To process this type of query, we use datasets of ‘hospital’, ‘restaurant’, and ‘mall’. After applying query on hospital dataset (i.e. list of hospitals whose specialty should be orthopedics and have at-least 100 beds available within 1 km range from query point ‘q’) we get some hospitals, mall and restaurant which is shown in Tables 7, 8, and 9 respectively. Where Table 7 shows a list of hospitals that cover the query keywords with congestion in their path from query point q. Table 8 shows a list of restaurants that cover the query keywords with congestion in their path from query point q.

First, we calculate network distance of each spatial object from query point ‘q’, with considering congestion in their path. After calculating distance, we found that point ‘H3’ is closer to query point and also has less congestion in the path from q to H3. After selecting 1st spatial object ‘H3’ next, we move to find 2nd spatial object (like mall or restaurant) but it depends on both (i) closer to query point ‘q’, and (ii) closer to 1st object (i.e., H3). To find 2nd object, we calculate the distance of each object of mall and restaurant from ‘H3’ with congestion in their path. Tables 10 and 11 show distance of each object from ‘H3’ with congestion.

After calculation, we get point ‘M6’, whose network distance from both query point q and 1st spatial object is minimum and also has less congestion in their path. After getting 2nd spatial object, now our aim is to find out the third spatial object. To find 3rd spatial object, it should be near to query location ‘q’ and from both 1st and 2nd spatial objects also with less congestion in their path. To find 3rd object, we calculate the distance of each object of restaurant from ‘H3’ and ‘M6’ with

Table 10 Malls whose distance from 'H3' and congestion between them

Mall	M1	M2	M3	M4	M5	M6
Network distance from 'H3' (km)	1.1	0.4	1.2	0.3	0.8	1.1
Value of μ from 'H3'	0.2	0.3	0.5	0.6	0.5	0.2

Table 11 Restaurants whose distance from 'H3' and congestion between them

Restaurant	R1	R2	R3	R4	R5
Network distance from 'H3' (km)	1.5	1.3	0.6	1.7	0.2
Value of μ from 'H3'	0.6	0.2	0.1	0.3	0.5

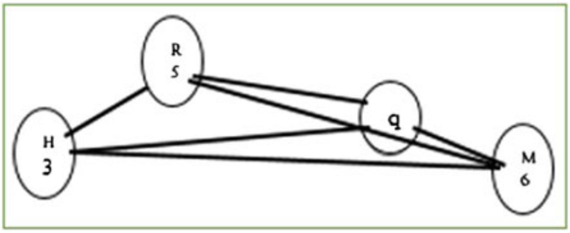


Fig. 6 Three spatial objects that cover the query's keywords

congestion in their path. Tables 5 and 10 show distance of each object from 'H3' and 'M6' with congestion.

After calculation traveling time from query point q and other spatial objects, we found that 'R5' is the spatial object that is closer to both query point as well as from other spatial objects which is shown in Fig. 6.

7 Conclusion and Future Work

In this paper, we describe a problem of finding group of spatial objects that are near to query point and near to each other and also have less congestion in their path such that it takes less time to cover all the points. This type of query helps tourist who wants to travel all the tourist places with less travel time. Also this query helps patient who needs emergency service, then in this case this query retrieves all the hospitals near the query point where emergency service is available.

Here, we consider a user who is static and query to find region of interest within some region from this query point, with consideration road congestion but user may be moved on the path and then query to find spatial objects also with the consideration of congestion in their path.

References

1. Samet, H., Sankaranarayanan, J., Alborzi, H.: Scalable network distance browsing inspatial databases. In: Proceedings of SIGMOD (2008)
2. Kolahdouzan, M., Shahabi, C.: Voronoi-based k-NN search in spatial networks. In: Proceedings of VLDB (2004)
3. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query processing in spatial networks. In: Proceedings of VLDB (2003)
4. Xin, C., Cong, G., Jensen, C.S., Ooi, B.C.: Collective spatial keyword querying. In: Proceedings of the 2011 International Conference on Management of Data—SIGMOD 11 (2011)
5. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of SIGMOD (1984)
6. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)
7. Finkel, R., Bentley, J.: Quad trees: a data structure for retrieval on composite keys. Acta Informatica **4**(1), 1–9 (1974)
8. Kolahdouzan, M., Shahabi, C.: Voronoi-based k nearest neighbor search for spatial network databases. In: Proceedings of 30th International Conference Very Large Data Bases, pp. 840–851 (2004)
9. Zhang, D., Ooi, B.C., Tung, A.K.H.: Locating mapped resources in web 2.0. In: ICDE, pp. 521–532 (2010)
10. Jagadish, H., Ooi, B., Tan, K.-L., Yu, C., Zhang, R.: iDistance: An adaptive b+-tree based indexing method for nearest neighbour search. ACM Trans. Database Syst. **30**(2), 364–397 (2005)
11. Zhang, D., Chee, Y.M., Mondal, A., Tung, A.K.H., Kitsuregawa, M.: Keyword search in spatial databases: towards searching by document. In: ICDE, pp. 688–699 (2009)
12. Cho, H.-J., Chung, C.-W.: An efficient and scalable approach to CNN queries in a road network. In: Proceedings of 31st International Conference Very Large Data Bases, pp. 865–876 (2005)
13. Mouratidis, K., Papadias, D., Hadjieleftheriou, M.: Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 634–645 (2005)
14. Jensen, C.S., Kolarvr, J., Pedersen, T.B., Timko, I.: Nearest neighbor queries in road networks. In: Proceedings of 11th ACM International Symposium on Advances in Geographic Information Systems, pp. 1–8 (2003)
15. Deng, K., Zhou, X., Shen, H.T., Xu, K., Lin, X.: Surface k-NN query processing. In: Proceedings of 22nd International Conference on Data Engineering, p. 78 (2006)
16. Qin, L., Yu, J.X., Ding, B., Ishikawa, Y.: Monitoring aggregate k-NN objects in road networks. In: Proceedings of SSDBM, pp. 168–186 (2008)
17. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.-H.: On trip planning queries in spatial databases. In: Proceedings of SSTD, pp. 273–290 (2005)
18. Cho, H.-J., Chung, C.-W.: An efficient and scalable approach to cnn queries in a road network. In: Proceedings of VLDB, pp. 865–876 (2005)
19. Mouratidis, K., Papadias, D., Hadjieleftheriou, M.: Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In: Proceedings of SIGMOD, pp. 634–645 (2005)

Algorithm for Representation of Call-Duration Graphs of Telephone Graph Using Multi-layer Graph Techniques

Bapuji Rao, S.N. Mishra and H.S. Maharana

Abstract Sometimes social networks frequently require multiple distinct types of connectivity information which can finally be derived as layers. This type of information retrieval can be represented as multi-layer graphs. Each layer has its own set of edges over the same underlying vertices or nodes of actual (base) layer. So the edges of different layers typically are related but unique in behavior which is based on particular applications. In this paper, we propose a telephone graph for representing as a multi-layer graph. From base layer, we try to retrieve the call history as layers based on their characteristics. Finally, the algorithm has been implemented in C++ programming language and observed satisfactory results.

Keywords Multi-layer graph · Directed weighted graph · Weight · Base layer · Sub-layer

1 Introduction

The fundamental study of networks in the study of complex network systems in biological, social, information, engineering, and physical sciences can be seen in [1–3]. In traditional studies of networks, generally the nodes are connected to each other by a single type of static edge that forms all connections between them. It is overall a gross over simplification. This kind of static connection leads to misleading results and lack of addressing certain problems. So the inter-layer

B. Rao (✉)
ITIMS, iNurture Education Solutions Pvt. Ltd., Bangalore 560052, Karnataka, India
e-mail: rao.bapuji@gmail.com

S.N. Mishra
Department of CSE & Applications, IGIT, Sarang 759146, Odisha, India
e-mail: sarose.mishra@gmail.com

H.S. Maharana
Department of Electrical Engineering, EATM, Bhubaneswar 752060, Odisha, India
e-mail: hsmaharana@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_39

connections have the ability to generate new structure in the form of dynamic correlations between components of a system. Hence, it is necessary to develop a framework that solves all the common problems related to the network [4].

For example, a social network user may currently be logged into Facebook but not logged into some other social networking site such as Google+ and Twitter. The presence of nodes in multi-layer systems of social network guarantees the possibility of self-interactions in a particular social network. This feature is not similar to the functions in interdependent networks, which were the ideas in interconnected communities within a single and larger network [5, 6].

When the source of connectivity information for a group of users is more than one, then there is a possibility of existence of multi-layer networks. The direct communication link in a social network is the knowledge, considered as relational information. From this, we can derive behavioral relationships based on user interests [7].

The layer which has "relational" information between users is an extrinsic layer which is considered as the base layer and the layer which has "behavioral" information between users is considered as the sub-layer of the social network [7]. An Email network has been used by [7] which is an example of social network. In this network, two layers can be created. The 1st layer is considered as relational network is created from the headers of emails by identifying the sender and receiver(s) of each message, which includes CC and BCC recipients, and the 2nd layer is considered as behavioral network is recovered from using the contents of email messages.

2 Basics of Multi-layer Network

2.1 Multi-layer Network

A multi-layer network is a quadruplet $M = (V_M, E_M, V, L)$. The first two elements V_M and E_M in M form a graph $G_M = (V_M, E_M)$. So, one can interpret a multi-layer network as a graph whose nodes are labeled [8]. A weighted multi-layer network M can be defined as its edges by weights in the underlying graph G_M (i.e., mapping each edge with a real number using a function $W: E \rightarrow R$). So, a multi-layer network M can be undirected or directed if the underlying graph G_M is undirected or directed.

An example of the most general type of multi-layer network $M = (V_M, E_M, V, L)$ in Fig. 2a can be seen from [8]. Another example of Zachary Karate Club (ZKC) network is represented as a multi-layer network in Fig. 3a can be seen from [8].

2.2 Tensor Representations

Generally, an ordinary graph (i.e., monoplex network) can be represented using adjacency matrices. For a node-aligned multi-layer network, it can be represented using a order of 2 × $(d + 1)$ adjacency tensor $\mathbf{A} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{|V|X|V|X|L_1|X|L_1|\ldots X|L_d|X|L_d|}}$ [4, 9]. In social networks, multi-dimensional arrays are sometimes called super-sociomatrices and adjacency matrices are called sociomatrices [9].

3 Proposed Model

Based on [7], we propose a telephone graph as another example of social network for representing as multi-layer graph. This telephone graph is considered as directed weighted graph. The edge between two phone numbers has a direction and a weight (i.e., total number of calls made). From this graph, we try to create four numbers of layers. The first layer is the relational layer which is considered as the base layer, whereas the remaining three layers are behavior layers which are considered as the sub-layers. The base layer holds the actual telephone graph. The sub-layers hold the behavior of the telephone graph based on particular characteristics. These behavior layers are formed with the characteristics of call duration of the user, i.e., low-call durations, medium-call durations and high-call durations. Hence, this proposed telephone graph has four numbers of layers.

Figure 1 shows the representation of proposed telephone graph in multi-layer graph form. The base layer is the actual telephone graph layer which has twenty numbers of users ID. The user IDs are ranging from 1 to 20. Here, the telephone graph is the directed weighted telephone graph because the edges with a specific direction assign with a weight (i.e., the total number of calls made by the user). The calls made by the user can be characterized into three groups, such as low-call durations, medium-call durations, and high-call durations respectively. Due to these characteristics of base layer graph, three sub-layers are formed. The 1st sub-layer is low-call durations graph, the 2nd sub-layer is medium-call durations graph, and the last one is high-call durations graph.

Each layer can be represented as weighted adjacency matrix. Based on the definition of multi-layer network, this telephone graph can be represented as a multi-dimensional array having the order 4 × 20 × 20, where 4 is the total number of layers and 20 × 20 is the order of each layer. It means that the above telephone graph has four numbers of double dimension arrays of order 20 × 20. Its weighted adjacency matrix representations are shown in Fig. 2.

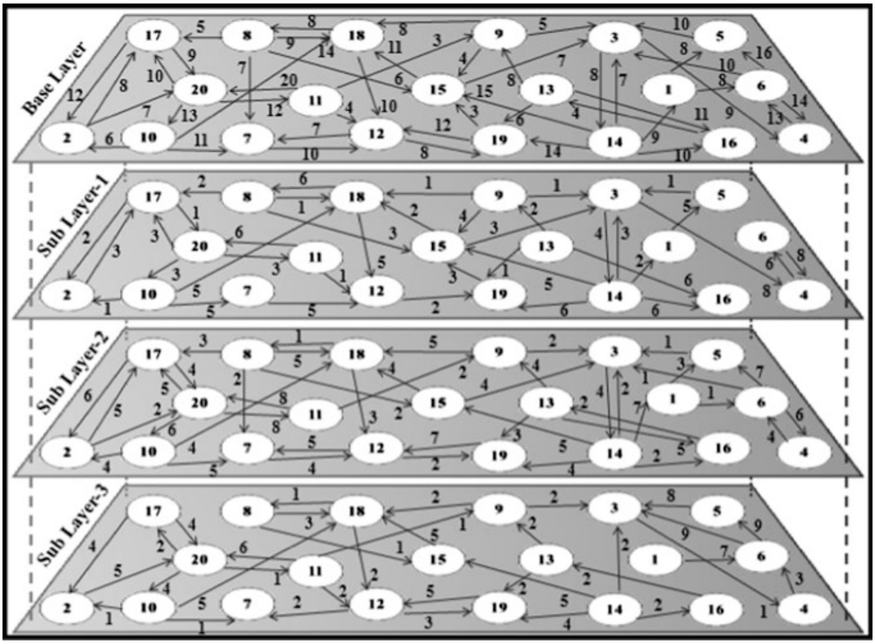
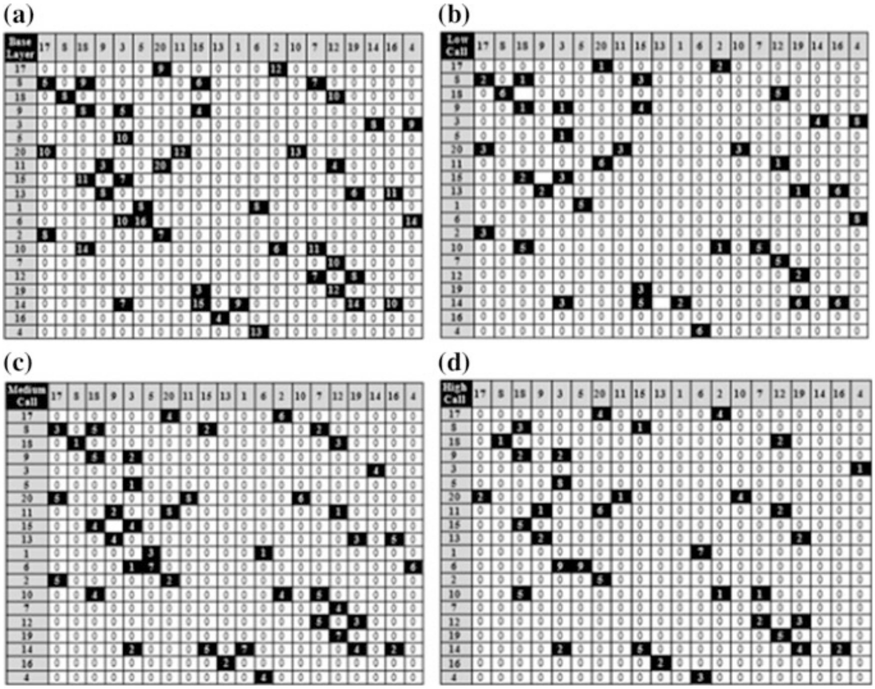


Fig. 1 Multi-layer network of telephone graph



(a)

Base Layer	17	8	18	9	3	5	20	11	15	13	1	6	2	10	7	12	19	14	16	4
17	0	0	0	0	0	0	9	0	0	0	0	0	12	0	0	0	0	0	0	0
8	5	0	9	0	0	0	0	0	6	0	0	0	0	0	7	0	0	0	0	0
18	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
9	0	0	8	0	5	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	9
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	10	0	0	0	0	0	0	12	0	0	0	0	0	13	0	0	0	0	0	0
11	0	0	0	3	0	0	20	0	0	0	0	0	0	0	0	4	0	0	0	0
15	0	0	11	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	6	0	11	0
1	0	0	0	0	0	8	0	0	0	0	0	8	0	0	0	0	0	0	0	0
6	0	0	0	0	10	16	0	0	0	0	0	0	0	0	0	0	0	0	0	14
2	8	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	14	0	0	0	0	0	0	0	0	0	6	0	11	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	0	0	0
19	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	12	0	0	0	0
14	0	0	0	0	7	0	0	0	15	0	9	0	0	0	0	0	14	0	10	0
16	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0

(b)

Low Call	17	8	18	9	3	5	20	11	15	13	1	6	2	10	7	12	19	14	16	4
17	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0
8	2	0	1	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
18	0	6		0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
9	0	0	1	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	8
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	3	0	0	0	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	0
11	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	2		3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	6	0
1	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	5	0	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
19	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	3	0	0	0	5		2	0	0	0	0	0	6	0	6	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0

(c)

Medium Call	17	8	18	9	3	5	20	11	15	13	1	6	2	10	7	12	19	14	16	4
17	0	0	0	0	0	0	4	0	0	0	0	0	6	0	0	0	0	0	0	0
8	3	0	5	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0
18	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
9	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	5	0	0	0	0	0	0	8	0	0	0	0	0	6	0	0	0	0	0	0
11	0	0	0	2	0	0	8	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	4		4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	3	0	5	0
1	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	6
2	5	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	4	0	0	0	0	0	0	0	0	0	4	0	5	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	3	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0
14	0	0	0	0	2	0	0	0	5	0	7	0	0	0	0	0	4	0	2	0
16	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0

(d)

High Call	17	8	18	9	3	5	20	11	15	13	1	6	2	10	7	12	19	14	16	4
17	0	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0	0	0	0
8	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
9	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	2	0	0	0	0	0	0	1	0	0	0	0	0	4	0	0	0	0	0	0
11	0	0	0	1	0	0	6	0	0	0	0	0	0	0	0	2	0	0	0	0
15	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0
6	0	0	0	0	9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	5	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
14	0	0	0	0	2	0	0	0	5	0	0	0	0	0	0	0	4	0	2	0
16	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0

Fig. 2 **a** Base layer weighted adjacency matrix. **b** Sub-layer-1 weighted adjacency matrix. **c** Sub-layer-2 weighted adjacency matrix. **d** Sub-layer-3 weighted adjacency matrix

4 Proposed Algorithm

4.1 Algorithm CDG ()

```
Structure Telephone
{
  Number id;
  Structure CallerPhone
  {
   [Caller id, no. of calls, and call durations]
      Number cid, nc;
      Real cd[100]; } CallerId[100]; }
nl: Number of layers.
tid: Number of users.
PhoneId.Txt: Text file contains number of layers, total
number of users and their ids.
ANetData.Txt: Text file contains the telephone graph's
edge (no. of calls) data.
CallData.Txt: Text file contains the telephone graph
details such as user id, caller id, number of calls, and
call durations of each call.
i. Open file "PhoneId.Txt"
ii. [Read no. of layers, no. of users, and ids]
    Read nl, tid
iii. Define array Id[tid]
iv. Repeat For i:=1, 2, .........,tid:
      Read Id[i]
    End For
v. Close file "PhoneId.Txt"
vi. Define layer of network matrices ANM[nl][tid][tid]
vii. Open file "ANetData.Txt"
viii. a.[Read data from file and assign to the 1st layer]
          Repeat For i:=1, 2, ........,tid:
             Repeat For j:=1, 2, ........,tid:
               Read ANM[1][i][j]
             End For
          End For
      b. Close file "ANetData.Txt"
ix. a. Define array Degree[tid]
    b. [Assign 0s to Degree[]]
       Repeat For i:=1, 2, .......,tid:
            Set Degree[i]:=0
       End For
   c. [Count outgoing calls of all users]
     Repeat For i:=1, 2, ........,tid:
      Repeat For j:=1, 2, ........,tid:
       If ANM[1][i][j]≠0,Then: Set Degree[i]:=Degree[i]+1
      End For
     End For
x. Define array of structure of Telephone kind Phone[tid]
xi. Open file "CallData.Txt"
xii. Repeat For i:=1, 2, .........,tid:
      a. [Read user id]
          Read Phone[i].id
      b.[Read caller id, no. of calls and calls duration]
             Repeat For j:=1, 2, ..........,Degree[i]:
                1. Read Phone[i].CallerId[j].cd, nc
                2. Set Phone[i].CallerId[j].nc:=nc
                3. Repeat For k:=1, 2, .........,nc:
                       Read Phone[i].CallerId[j].cd[k]
                   End For
             End For
     End For
xiii. Call Layers_Creation(ANM, Phone)
xiv. Call Layers_Show(ANM)
xv. Exit
```

4.2 Procedure Layers_Show (ANM)

```
i. Repeat For i:=1, 2, ......., nl:
     Repeat For j:=1, 2, .......,tid:
       Repeat For k:=1, 2, .......,tid:
            Display ANM[i][j][k]
       End For
     End For
   End For
ii. Return
```

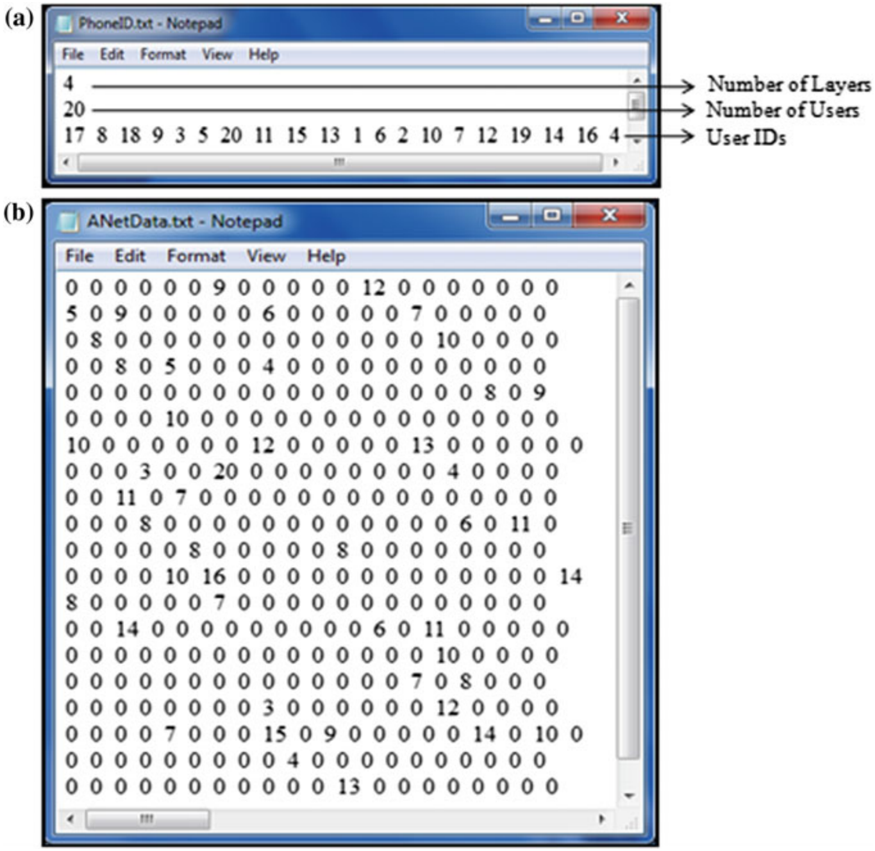
4.3 Procedure Layers_Creation (ANM, Phone)

```
i. [Assign 0s to from 2nd to 'nl' layers of ANM[][][]]
   Repeat For i:=2, 3, ......., nl:
      Repeat For j:=1, 2, .......,tid:
         Repeat For k:=1, 2, .......,tid:
            Set ANM[i][j][k]:=0
         End For
      End For
   End For
ii. Repeat For i:=1, 2, .......,tid:
       a. [Find the index position of user id]
           Repeat For rindex:=1, 2, ......,tid:
              If Id[rindex]=Phone[i].id, Then: Break.
           End For
       b. [No. of callers from Degree[i]]
           Repeat For j:=1, 2, ........,Degree[i]:
            c. Set nc:=Phone[i].CallerId[j].nc
            d. [Finding the index position of caller id]
               Repeat For cindex:=1, 2, .....,tid:
     If Id[cindex]=Phone[i].CallerId[j].cid, Then: Break.
               End For
            e. [Counting call durations]
               Repeat For k:=1, 2, ........,nc:
    1. Set Duration:=Phone[i].CallerId[j].cd[k]
    2. If Duration>0 and Duration<=5 [2nd layer counting]
       Then
     Set ANM[2][rindex][cindex]:=ANM[2][rindex][cindex]+1
       End If
    3. If Duration>5 and Duration<=10 [3rd layer counting]
       Then
     Set ANM[3][rindex][cindex]:=ANM[3][rindex][cindex]+1
       End If
    4. If Duration>10 [4th layer counting]
       Then
     Set ANM[4][rindex][cindex]:=ANM[4][rindex][cindex]+1
       End If
               End For
           End For
    End For
iii. Return
```

5 Experimental Results

Three data files, namely “PhoneId.txt”, “ANetData.txt”, and “CallData.txt” are considered as datasets to the proposed algorithm; and shown in Figs. 3 and 4. The 1st data file contains total number of layers, number of ids, and the ids. The 2nd data file contains the actual number of calls between the users as edge weight. The 3rd data file contains the Ids, caller Ids, numbers of calls made, and the call durations. The algorithm has been implemented in C++ programming language and the output is satisfactory. The screen shots are shown in Figs. 5 and 6.

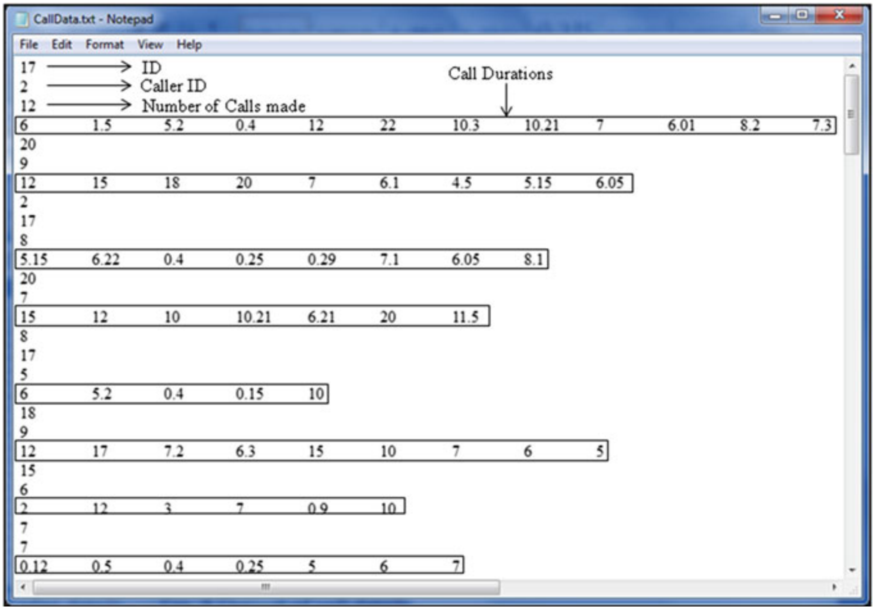
Finally, the Phase III of algorithm reads the details of user Id, caller Id, number of calls made, and calls durations of each call from text file “CallData.Txt” which is shown in Fig. 4. These details are assigned to array of structure Phone[]. Based on the details of Phone[], the layers of telephone graph are created by calling the procedure Layers_Creation().



```
4
20
17 8 18 9 3 5 20 11 15 13 1 6 2 10 7 12 19 14 16 4
```

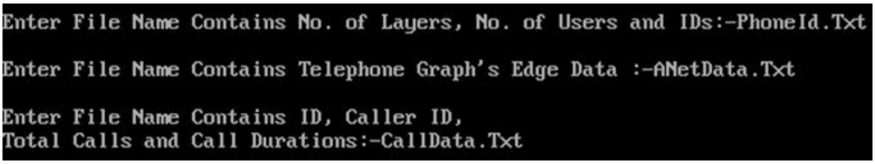
```
0 0 0 0 0 0 9 0 0 0 0 0 12 0 0 0 0 0 0 0
5 0 9 0 0 0 0 0 6 0 0 0 0 0 7 0 0 0 0 0
0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0
0 0 8 0 5 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 9
0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 12 0 0 0 0 0 13 0 0 0 0 0 0
0 0 0 3 0 0 20 0 0 0 0 0 0 0 0 4 0 0 0 0
0 0 11 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0 6 0 11 0
0 0 0 0 0 8 0 0 0 0 0 8 0 0 0 0 0 0 0 0
0 0 0 0 10 16 0 0 0 0 0 0 0 0 0 0 0 0 0 14
8 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 14 0 0 0 0 0 0 0 0 0 6 0 11 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 8 0 0 0
0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 12 0 0 0 0
0 0 0 0 7 0 0 0 15 0 9 0 0 0 0 0 14 0 10 0
0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 13 0 0 0 0 0 0 0 0
```

Fig. 3 **a** Dataset of phone Id details. **b** Dataset of edge details



```
17
2
12
6       1.5     5.2     0.4     12      22      10.3    10.21   7       6.01    8.2     7.3
20
9
12      15      18      20      7       6.1     4.5     5.15    6.05
2
17
8
5.15    6.22    0.4     0.25    0.29    7.1     6.05    8.1
20
7
15      12      10      10.21   6.21    20      11.5
8
17
5
6       5.2     0.4     0.15    10
18
9
12      17      7.2     6.3     15      10      7       6       5
15
6
2       12      3       7       0.9     10
7
7
0.12    0.5     0.4     0.25    5       6       7
```

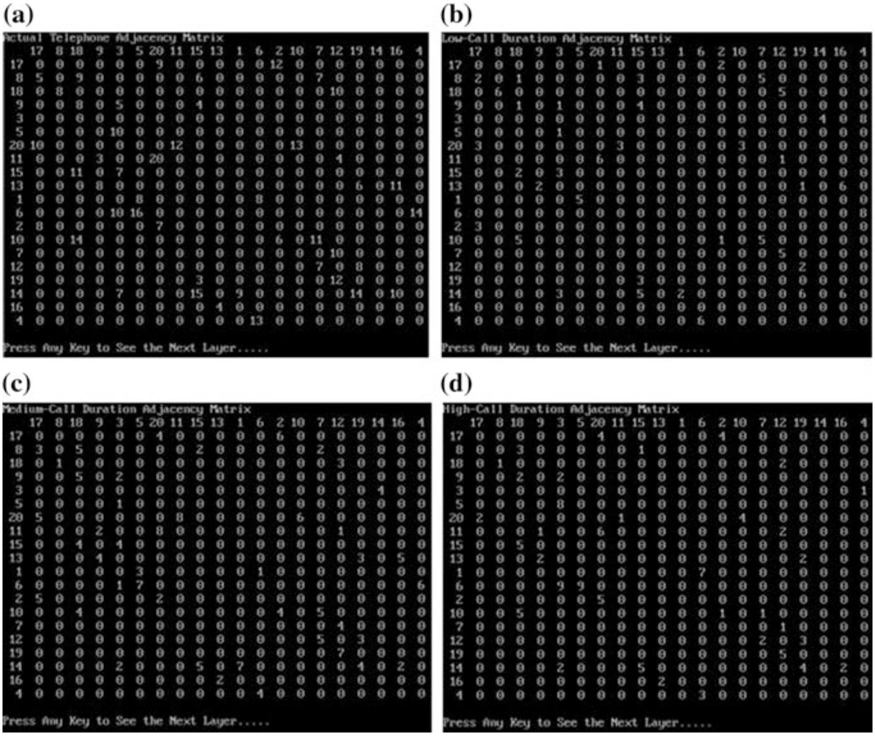
Fig. 4 Dataset of call details



```
Enter File Name Contains No. of Layers, No. of Users and IDs:-PhoneId.Txt

Enter File Name Contains Telephone Graph's Edge Data :-ANetData.Txt

Enter File Name Contains ID, Caller ID,
Total Calls and Call Durations:-CallData.Txt
```

Fig. 5 Dataset input

Hence, the text files “PhoneId.Txt”, “ANetData.Txt”, and “CallData.Txt” are considered as datasets to the proposed algorithm. The algorithm has been implemented in C++ programming language and the output is satisfactory. The screen shots are shown in Figs. 5 and 6.



(a)

```
Actual Telephone Adjacency Matrix
      17   8  18   9   3   5  20  11  15  13   1   6   2  10   7  12  19  14  16   4
  17   0   0   0   0   0   0   9   0   0   0   0   0  12   0   0   0   0   0   0   0
   8   5   0   9   0   0   0   0   0   6   0   0   0   0   0   7   0   0   0   0   0
  18   0   8   0   0   0   0   0   0   0   0   0   0   0   0   0  10   0   0   0   0
   9   0   0   8   0   5   0   0   0   4   0   0   0   0   0   0   0   0   0   0   0
   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   8   0   9
   5   0   0   0   0  10   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  20  10   0   0   0   0   0   0  12   0   0   0   0   0  13   0   0   0   0   0   0
  11   0   0   0   3   0   0  20   0   0   0   0   0   0   0   0   4   0   0   0   0
  15   0   0  11   0   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  13   0   0   0   8   0   0   0   0   0   0   0   0   0   0   0   0   6   0  11   0
   1   0   0   0   0   0   8   0   0   0   0   0   8   0   0   0   0   0   0   0   0
   6   0   0   0   0  10  16   0   0   0   0   0   0   0   0   0   0   0   0   0  14
   2   8   0   0   0   0   0   7   0   0   0   0   0   0   0   0   0   0   0   0   0
  10   0   0  14   0   0   0   0   0   0   0   0   0   6   0  11   0   0   0   0   0
   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  10   0   0   0   0
  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   7   0   8   0   0   0
  19   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0  12   0   0   0   0
  14   0   0   0   0   7   0   0   0  15   0   9   0   0   0   0   0  14   0  10   0
  16   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0   0   0   0   0   0
   4   0   0   0   0   0   0   0   0   0   0   0  13   0   0   0   0   0   0   0   0

Press Any Key to See the Next Layer.....
```

(b)

```
Low-Call Duration Adjacency Matrix
      17   8  18   9   3   5  20  11  15  13   1   6   2  10   7  12  19  14  16   4
  17   0   0   0   0   0   0   1   0   0   0   0   0   2   0   0   0   0   0   0   0
   8   2   0   1   0   0   0   0   0   3   0   0   0   0   0   5   0   0   0   0   0
  18   0   6   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0
   9   0   0   1   0   1   0   0   0   4   0   0   0   0   0   0   0   0   0   0   0
   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   8
   5   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  20   3   0   0   0   0   0   0   3   0   0   0   0   0   3   0   0   0   0   0   0
  11   0   0   0   0   0   0   6   0   0   0   0   0   0   0   0   1   0   0   0   0
  15   0   0   2   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  13   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   1   0   6   0
   1   0   0   0   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   8
   2   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  10   0   0   5   0   0   0   0   0   0   0   0   0   1   0   5   0   0   0   0   0
   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0
  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
  19   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0
  14   0   0   0   0   3   0   0   0   5   0   2   0   0   0   0   0   6   0   6   0
  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   4   0   0   0   0   0   0   0   0   0   0   0   6   0   0   0   0   0   0   0   0

Press Any Key to See the Next Layer.....
```

(c)

```
Medium-Call Duration Adjacency Matrix
      17   8  18   9   3   5  20  11  15  13   1   6   2  10   7  12  19  14  16   4
  17   0   0   0   0   0   0   4   0   0   0   0   0   6   0   0   0   0   0   0   0
   8   3   0   5   0   0   0   0   0   2   0   0   0   0   0   2   0   0   0   0   0
  18   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0
   9   0   0   5   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0
   5   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  20   5   0   0   0   0   0   0   8   0   0   0   0   0   6   0   0   0   0   0   0
  11   0   0   0   2   0   0   8   0   0   0   0   0   0   0   0   1   0   0   0   0
  15   0   0   4   0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  13   0   0   0   4   0   0   0   0   0   0   0   0   0   0   0   0   3   0   5   0
   1   0   0   0   0   0   3   0   0   0   0   0   1   0   0   0   0   0   0   0   0
   6   0   0   0   0   1   7   0   0   0   0   0   0   0   0   0   0   0   0   0   6
   2   5   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0
  10   0   0   4   0   0   0   0   0   0   0   0   0   4   0   5   0   0   0   0   0
   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0
  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   3   0   0   0
  19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   7   0   0   0   0
  14   0   0   0   0   2   0   0   0   5   0   7   0   0   0   0   0   4   0   2   0
  16   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0
   4   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0   0   0   0

Press Any Key to See the Next Layer.....
```

(d)

```
High-Call Duration Adjacency Matrix
      17   8  18   9   3   5  20  11  15  13   1   6   2  10   7  12  19  14  16   4
  17   0   0   0   0   0   0   4   0   0   0   0   0   4   0   0   0   0   0   0   0
   8   0   0   3   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
  18   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0
   9   0   0   2   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
   5   0   0   0   0   8   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  20   2   0   0   0   0   0   0   1   0   0   0   0   0   4   0   0   0   0   0   0
  11   0   0   0   1   0   0   6   0   0   0   0   0   0   0   0   2   0   0   0   0
  15   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  13   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0
   1   0   0   0   0   0   0   0   0   0   0   0   7   0   0   0   0   0   0   0   0
   6   0   0   0   0   9   9   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   2   0   0   0   0   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   0
  10   0   0   5   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0
   7   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   3   0   0   0
  19   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0
  14   0   0   0   0   2   0   0   0   5   0   0   0   0   0   0   0   4   0   2   0
  16   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0
   4   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0

Press Any Key to See the Next Layer.....
```

Fig. 6 **a** Actual telephone adjacency matrix. **b** Low-call duration adjacency matrix. **c** Medium-call adjacency matrix. **d** High-call adjacency matrix

6 Conclusion

Sometimes social networks necessitate information retrieval from a multi-layer graphs. This type of information retrieval can be represented as multi-layer graphs. So each layer has its own set of edges over the same underlying vertices or nodes of actual (base) layer. So the edges of different layers typically are related but unique in behavior which is based on particular applications. This paper has proposed a telephone graph and algorithm for representing a telephone graph as a multi-layer graph. The algorithm has successfully retrieved the call history as layers based on their characteristics from the base layer of the proposed telephone graph. Finally, the algorithm has been implemented in C++ programming language and observed satisfactory results.

References

1. Estrada, E.: The Structure of Complex Networks: Theory and Applications. Oxford University Press, Oxford, UK (2011)
2. Newman, M.E.J.: Networks: An Introduction. Oxford University Press (2010)
3. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Phys. Rep. **424**, 175 (2006)
4. De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M.A., Gómez, S., Arenas, A.: Mathematical formulation of multilayer networks. Phys. Rev. X (2013)
5. Dickison, M., Havlin, S., Stanley, H.E.: Phys. Rev. E **85**, 066109 (2012)
6. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Nature **464**, 1025 (2010)
7. Oselio, B., Kulesza, A., Hero, A.O.: Multi-layer graph analytics for social networks. Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. http://www.cs.cmu.edu/~enron
8. Kivelä, Mikko, Arenas, Alex, Barthelemy, Marc, Gleeson, James P., Moreno, Yamir, Porter, Mason A.: Multilayer networks. J. Complex Netw. **2**, 203–271 (2014). doi:10.1093/comnet/cnu016
9. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)

Weibull Probability Distribution Function-Based Matched Filter Approach for Retinal Blood Vessels Segmentation

Nagendra Pratap Singh and Rajeev Srivastava

Abstract Retinal blood vessels contain an important information that is useful for computer-aided diagnosis of various retinal pathologies such as hypertension, diabetes, glaucoma, etc. Therefore, a retinal blood vessel segmentation is a prominent task. In this paper, a novel Weibull probability distribution function-based matched filter approach is introduced to improve the performance of retinal blood vessel segmentation with respect to prominent matched filter approaches and other matched filter-based approaches existing in literature. Moreover, to enhance the quality of input retinal images in pre-processing step, the concept of principal component analysis (PCA)-based gray scale conversion and contrast-limited adaptive histogram equalization (CLAHE) are used. To design a proposed matched filter, the appropriate value of parameters are selected on the basis of an exhaustive experimental analysis. The proposed approach has been tested on 20 retinal images of test set taken from the DRIVE database and confirms that the proposed approach achieved better performance with respect to other prominent matched filter-based approaches.

Keywords Weibull probability distribution function · Matched filter · Retinal blood vessels segmentation · Entropy-based optimal thresholding

1 Introduction

The automated retinal blood vessel segmentation is a prominent task for computer-aided diagnosis of retinal pathologies such as glaucoma, hypertension, diabetes, obesity, etc. Vessel structures are similar to a cluster of lines, hence retinal blood vessel segmentation is a line detection problem. After an in-depth literature survey, it was found that various methods have been proposed for retinal blood vessel

N.P. Singh (✉) · R. Srivastava
Department of CSE, Indian Institute of Technology (BHU), Varanasi 221005, U.P., India
e-mail: npsingh.rs.cse13@itbhu.ac.in

R. Srivastava
e-mail: rajeev.cse@iitbhu.ac.in

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_40

segmentation. According to Fraz et al. [1], the retinal blood vessel segmentation approaches are mainly classified into seven categories, namely, the intensity-based pattern recognition techniques, mathematical morphology based, vessel tracking based, model based, parallel hardware based, multi-scale-based techniques and matched filter-based approach.

In this paper, we concentrate on matched filter-based retinal blood vessel segmentation approach, because it provides the best retinal vessel structure with respect to other filtering approaches [2, 3]. The matched filter-based approaches compare the variations of the intensity level of the cross-section profile of the retinal image with the predetermined template or kernel. To design a matched filter kernel, three points are important which include limited curvature of vessels which may be approximated by piece-wise linear segments and the width of the vessels gradually decreases when moving away from the optical disk of the retinal image, and cross-sectional intensity profile of retinal blood vessels.

For the first time Chaudhuri et al. [2] stated that the cross-sectional profile of retinal blood vessels has an approximate Gaussian shape. Hence the Gaussian-shaped match filter kernel is required for the detection of retinal blood vessels. According to the literature survey on matched filter-based approach, it was found that various authors improved the performance of matched filter-based approach by improving the thresholding techniques rather than changing the Gaussian-shaped matched filter kernel. The matched filter-based approaches use the prior knowledge about the shape and symmetry of the cross-section profile of the retinal blood vessel with respect to their shape, location and scale parameter. Therefore, Zolfagharnasa and Naghsh-Nilchi [4] replaced the Gaussian function-based matched filter by Cauchy probability distribution function (PDF) and reported that the accuracy of retinal blood vessel detection was substantially improved. Due to this reason, we examined other PDFs to improve the accuracy of retinal blood vessel segmentation and found the Weibull PDF more suited to the purpose. So in this paper, we propose a novel Weibull PDF-based matched filter approach for retinal blood vessel segmentation.

If we analyze the cross-section profile and well-shaped cross-section profile of two randomly selected gray scaled retinal images from DRIVE database, as shown in Fig. 1a, c, and Fig. 1b, d respectively, with the actual Gaussian, Cauchy, and Weibull PDFs curves as shown in Fig. 1e, it is found that the Weibull PDF provides a better match to vessel cross-section profiles in comparison to Gaussian and Cauchy PDF curves. Therefore, the Weibull PDF-based matched filter approach may lead to a better result with respect to other existing matched filter-based approaches. This fact is also examined and validated through experimental result shown in results and performance analysis section.

The rest of the paper is organized as follows: Sect. 2 presents the proposed method and model, Sect. 3 presents result analysis of proposed approach that is tested on DRIVE database and results are encouraging. Finally, Sect. 4 presents the conclusion of the proposed approach.

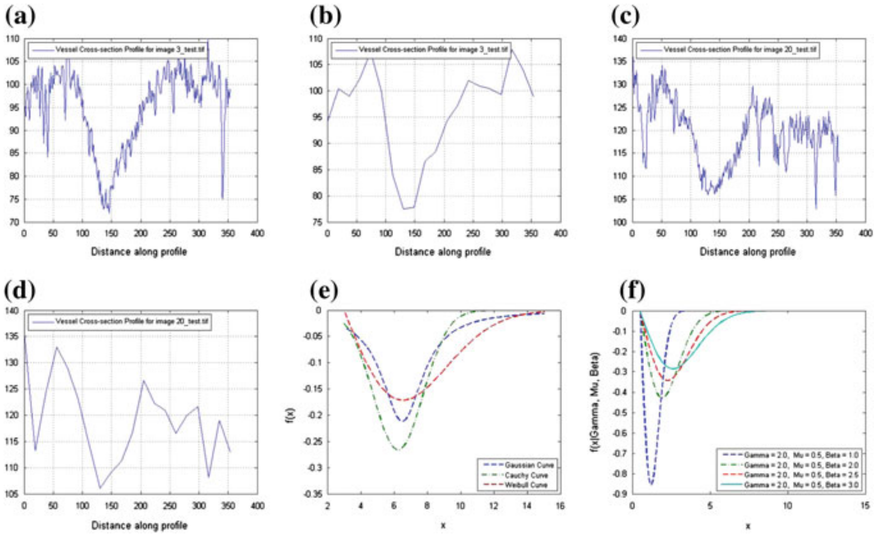


Fig. 1 **a**, **c** Cross-section profile. **b**, **d** Approximate well-shaped cross-section profile of ‘3_test.tif’ and ‘20_test.tif’ images taken from DRIVE dataset, respectively. **e** Actual Gaussian, Cauchy, and Weibull PDF curve. **f** Weibull PDF curve with different γ, β, and μ

2 Proposed Method and Model

Our proposed Weibull PDF-based matched filter approach for retinal blood vessel segmentation is a combination of pre-processing, Weibull PDF-based matched filtering, entropy-based optimal thresholding, vessel length filtering and removing outliers artifacts.

2.1 Pre-processing

The contrast difference of the retinal blood vessels is low with respect to their background and decreases when moving away from the optical disk of the retinal image. Therefore, to improve the accuracy of retinal blood vessel segmentation, contrast enhancement is an essential pre-processing task. In the pre-processing step, various authors [2, 3, 5–8] used only green channel image of RGB color retinal image in the successive segmentation process. Because the contrast of the blood vessels in the green channel image is better in comparison to the red and blue channels of the RGB color retinal image. After that the green channel image is converted into gray scale image and then the successive step of particular retinal blood vessel segmentation approach has been applied on gray scale image generated from respective green channel image.

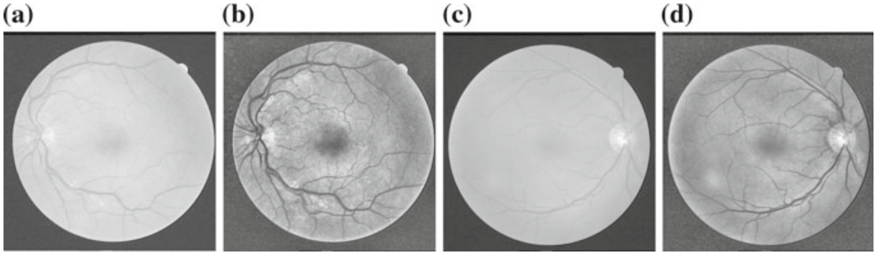


Fig. 2 **a**, **c** Gray scale image. **b**, **d** Contrast-enhanced gray scale image of '3_test.tif' and '20_test.tif', respectively

In this paper, principal component analysis (PCA)-based color-to-gray image conversion method is used to generate a gray scale image of a color retinal image, because the PCA-based color-to-gray conversion method effectively preserves both the texture and color discriminability by using simple linear computations in subspaces with low computational complexity [9]. The generated gray scale images of two randomly selected retinal images from the DRIVE database are shown in Fig. 2a, c, respectively. After that, contrast-limited adaptive histogram equalization (CLAHE) is applied on the gray scale image and the contrast-enhanced gray scale image is generated, as shown in Fig. 2b, d for the respective gray scale retinal images.

2.2 *Weibull PDF-Based Matched Filter*

The matched filter-based retinal blood vessel segmentation approach compares the gray scale cross-section profile of retinal blood vessels with the predefined kernel. Chaudhuri et al. [2] proposed a first matched filter approach based on Gaussian function and claimed the vessel cross-section profile is in approximate Gaussian shape. The actual Gaussian curve is shown in Fig. 1e. Zolfagharnasa and Naghsh-Nilchi [4] proposed a Cauchy PDF-based matched filter approach and claimed the vessel cross-section profile matched better with a Cauchy PDF curve. As shown in Fig. 1e, the Gaussian and Cauchy PDFs are both equally diminishing toward their truncated values on both sides of their peak, but the Gaussian function reaches its truncated value faster with respect to the Cauchy PDF. If we analyze the well-shaped vessel cross-section profile of the gray scale retinal image as shown in Fig. 1b, d, then it is clear that the cross-section profile is not equally diminished toward their truncated values on both sides of their peak value.

The Weibull PDF curve with different values of parameters as shown in Fig. 1f is diminished towards their truncate value, but reached to its truncate value with different diminishing rates on both sides of its peak value. This characteristic of Weibull PDF better matched the cross-section profile of a gray scale retinal image with respect to the Gaussian and Cauchy PDF. Therefore, in this paper, we propose the novel Weibull PDF-based matched filter approach. The Weibull PDF is an extreme valued probability distribution function and defined as:

$$f(x) = \frac{\gamma}{\beta}\left(\frac{x-\mu}{\beta}\right)^{\gamma-1} e^{-\left(\frac{x-\mu}{\beta}\right)^{\gamma}}, \tag{1}$$

where x is the perpendicular distance between points (x, y) and straight line passing through the center of retinal blood vessel, μ is the location parameter, β is the scale parameter, and γ is the shape parameter. The length of retinal blood vessels is assumed as piece-wise line segment in the kernel. The length of piece-wise line segment (L) is determined experimentally, by analyzing the blood vessels in both normal and abnormal retinal images. According to the experimental result, for L = 1 to 15, as shown in Table 1, and Fig. 3a, it is clear that the average accuracy of retinal blood vessel segmentation is continuously increasing from L = 1 to 11 after that decreases from L = 12 to onward. The highest average accuracy is achieved at L = 11. So in this paper the length of piece-wise line segment L = 11 is selected. This step reduces the possibility of false vessel detection in non-ideal environment [2].

In case of matched filtering approach to cover the piece-wise line segments of a whole retinal image, a set of kernel is required in every direction. In the proposed approach 12 different kernels of size 15 × 15 are used to detect the retinal blood vessels in every direction by rotating the kernel 15° with respect to the previous one. Three main parameters are used in Weibull PDF known as the scale parameter (β), location parameter (μ), and shape parameter γ as used in $Eq^n - 1$. The scale parameter (β) must be a positive real number. The behavior of β used in Weibull PDF is similar to the variance and scale parameter used in Gaussian and Cauchy PDF, respectively. For small value of β, the Weibull PDF curve narrows down and for large value, the Weibull curve expands as shown in Fig. 1f. However, the behavior of scale parameter (β) used in Weibull PDF is different from the variance and scale

Table 1 Average accuracy of proposed approach w.r.t. various sizes of piece-wise line segment (L) for image taken from DRIVE database

Length of piece-wise line segment (L)	Average accuracy	Length of piece-wise line segment (L)	Average accuracy
1	0.8953	9	0.9440
2	0.9041	10	0.9494
3	0.9186	11	0.9513
4	0.9171	12	0.9482
5	0.9306	13	0.9463
6	0.9344	14	0.9467
7	0.9402	15	0.9426
8	0.9422		

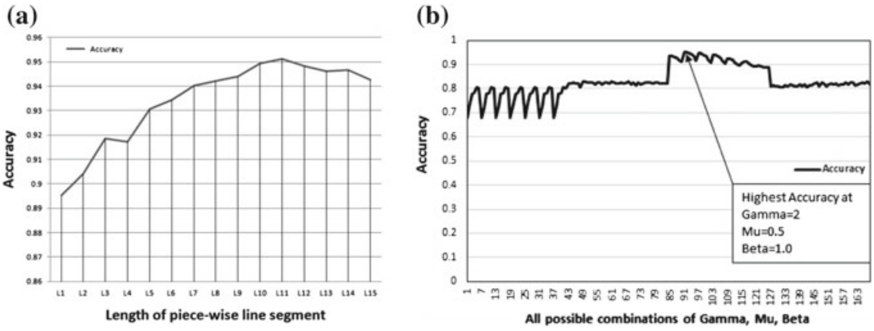


Fig. 3 **a** Graphical presentation of comparative analysis. **b** Accuracy graph of proposed approach with respect to various combination of γ, μ, and β

parameter of Cauchy PDF with respect to their diminishing behavior towards their truncated value as previously discussed and clearly visible in Fig. 1e.

In Weibull PDF the location parameter (μ) is used to shift the graph on horizontal X-axis. All retinal blood vessels pass through the center of the retinal fundus image. Therefore, the selection of location parameter (μ) is important. The shape parameter (γ) used in Weibull PDF manages the shape of the Weibull curve. This parameter is important because it is able to select the more appropriate shape of the curve that is well matched with the vessel cross-section profile of the retinal images and provides better accuracy.

The proposed approach has been tested for some selected values of the shape parameter (γ) from 1.0 to 2.5 with 0.5 intervals at different values of scale parameter β (from 0.5 to 3 with 0.5 intervals), and at different locations ($\mu = 0$ to 3 with 0.5 intervals). After the analysis of obtained accuracy for all 168 different possible combinations of γ, μ and β as shown in Fig. 3b, it is clear that the highest average accuracy of retinal blood vessel segmentation is achieved at $\gamma = 2.0$, $\mu = 0.5$, and $\beta = 1.0$. So these values are used in the proposed Weibull PDF-based matched filter approach. On the basis of above selected parameters, constructing the filter set using the Weibull PDF-based kernel, then convolved them and find the matched filter response (MFR) image. The MFR image of two randomly selected retina images taken from DRIVE data set are shown in Fig. 4a, f.

2.3 Entropy-Based Optimal Thresholding

To obtain the binary image of the retinal blood vessel structure from the Weibull PDF-based matched filter response (MFR) image, an efficient thresholding algorithm is required. For this purpose, the entropy-based optimal thresholding algorithm [4] is used. The entropy-based optimal thresholding algorithm is better than an other

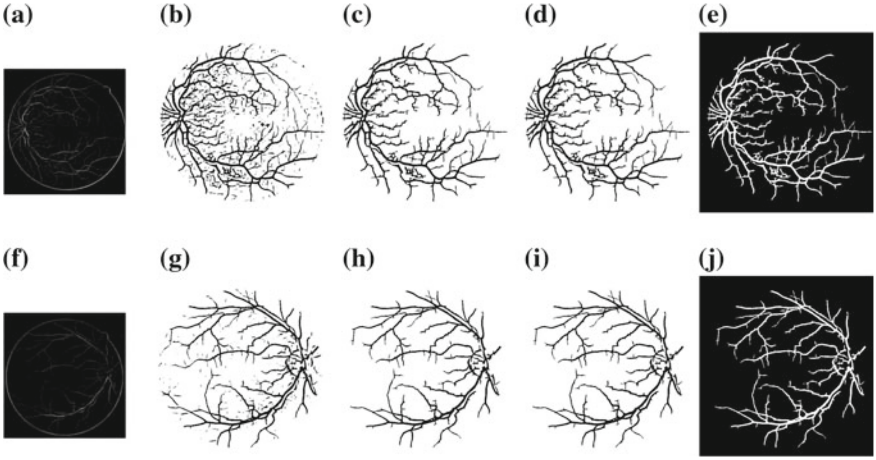


Fig. 4 **a**, **f** MFR image. **b**, **g** Classified retinal vessels from their backgrounds. **c**, **h** Retinal vessel after removing isolated and misclassified pixels. **d**, **i** After removing outer artifacts. **e**, **j** Complimented segmented image of '3_test.tif' and '20_test.tif', respectively

thresholding algorithm. Reason for using the above thresholding algorithm is that the dependencies between the pixel's intensity of MFR image allows us to preserve the spatial structure of the thresholding images and capable of correctly classifying the blood vessels of retinal image from their backgrounds as argued in [4]. The segmented retinal blood vessels from their background are shown in Fig. 4b, g for image '3_test.tif' and '20_test.tif' taken from a test set of DRIVE database, respectively.

2.4 Length Filtering and Removing Outer Artifacts

Some isolated and misclassified pixels are present in Fig. 4b, g. Therefore, to remove them length filtering is applied on Fig. 4b, g by using an eight-connected neighborhood and pixel label propagations after that find a retinal blood vessels structure without isolated and misclassified pixels, as shown in Fig. 4c, h. In Fig. 4c, h some artifacts may be present outside the retinal boundary. Therefore, to remove them the mask generated by particular retinal image is applied, as shown in Fig. 4d, i. Finally, a compliment of the segmented image is generated as shown in Fig. 4e, j which is used for evaluating the performance measures such as TPR, FPR and accuracy with respect to ground truth image available in DRIVE database.

3 Results and Performance Analysis

The proposed Weibull PDF-based matched filter approach has been implemented on a publicly available DRIVE database of retinal image [10]. The images taken from the DRIVE database were obtained from a diabetic retinopathy screening program in the Netherlands [10]. The ground truth image was manually segmented by observers and computer science students. They have been trained by an experienced ophthalmologist, Abramoff et al. [11]. The proposed vessel-segmentation approach has been implemented on Matlab R2013a in a PC having AMD E-450 APU, with 1.65 GHz processor and 2 GB RAM. The average execution time of entire process for 20 images of DRIVE database took 2.34 min.

To compare the performance of the proposed approach, the quantitative performance measures, an average accuracy, average true positive rate (ATPR), and the average false positive rate (AFPR) are evaluated. The accuracy of the segmentation algorithms is defined as the ratio of total correctly classified pixels with total number of pixels in the selected image. The TPR is the ratio of total correctly classified vessel pixels in total number of vessel pixels in the respective ground truth image and similarly the FPR is the ratio of total correctly classified non-vessel pixels in total number of non-vessel pixels in the respective ground truth image. These quantitative performance measures are widely used and defined in the literature [2, 3, 5–8, 10, 11].

In this paper, the proposed approach is compared with three prominent matched filter approaches, namely a classical Gaussian matched filter approach introduced by Chaudhuri et al. [2], improved matched filter approach introduced by Al-Rawi et al. [3], and Cauchy matched filter approach introduced by Zolfagharnasab and Naghsh-Nilchi [4] as well as some other segmentation approach based on matched filter, proposed by Zhang et al. [6], Jiang and Mojon [5], Cinsdikici and Aydın [7], and Amin and Yan [8]. The values of quantitative performance measures for 20 retinal images taken from a test set of DRIVE database are shown in Table 2 and the comparative analysis with respect to matched filter-based segmentation approaches exist in the literature are shown in Table 3. The graphical presentation of comparative analysis is shown in Fig. 5a. On the basis of result analysis the performance measures (ATPR, AFPR, average accuracy) of the proposed approach are found better with respect to prominent matched filter approaches as well as other segmentation approaches based on matched filter.

Other than the quantitative performance measures, the receiver operating characteristic (ROC) curve is used to justify the effectiveness of the segmentation approach. ROC curve is a graphical representation of TPR (sensitivity) on the y-axis and FPR (1-specificity) on the x-axis. The area under the curve (AUC) is an important component of ROC which is used for justifying the effectiveness of the segmentation approach. The maximum value of AUC is 1 which indicates the segmentation approach is perfect [12] that is practically not possible in case of retinal blood vessel segmentation. The value of AUC closer is to 1 indicating better performance of the segmentation approach [12].

Table 2 Quantitative performance measures of proposed approach

Image	TPR	FPR	Accuracy	Image	TPR	FPR	Accuracy
01_test.tif	0.8105	0.0343	0.9518	11_test.tif	0.7325	0.0372	0.9422
02_test.tif	0.7532	0.0210	0.9558	12_test.tif	0.7345	0.0269	0.9525
03_test.tif	0.7427	0.0386	0.9396	13_test.tif	0.7190	0.0306	0.9449
04_test.tif	0.6501	0.0110	0.9578	14_test.tif	0.7611	0.0295	0.9536
05_test.tif	0.7543	0.0249	0.9544	15_test.tif	0.8136	0.0503	0.9399
06_test.tif	0.5493	0.0075	0.9494	16_test.tif	0.6736	0.0178	0.9543
07_test.tif	0.7196	0.0284	0.9486	17_test.tif	0.6020	0.0148	0.9529
08_test.tif	0.6883	0.0322	0.9438	18_test.tif	0.6957	0.0172	0.9601
09_test.tif	0.6259	0.0149	0.9560	19_test.tif	0.8326	0.0330	0.9559
10_test.tif	0.7043	0.0246	0.9531	20_test.tif	0.7791	0.0270	0.9587
Average	TPR	FPR	Accuracy	=>	0.7171	0.0261	0.9513

Table 3 Comparative analysis of proposed approach w.r.t. existing matched filter approach for DRIVE database

Matched filter approach	ATPR	AFPR	Average accuracy
Chaudhuri et al. [2]	0.6326	0.0936	0.8709
Al-Rawi et al. [3]	0.5993	0.0447	0.9096
Zolfagharnasab and Naghsh-Nilchi [4]	0.6239	0.0286	0.9269
Zhang et al. [6]	0.7120	0.0286	0.9382
Jiang and Mojon [5]	–	–	0.9212
Cinsdikici and Aydın [7]	–	–	0.9293
Amin and Yan [8]	–	–	0.9200
Proposed approach	0.7171	0.0261	0.9513

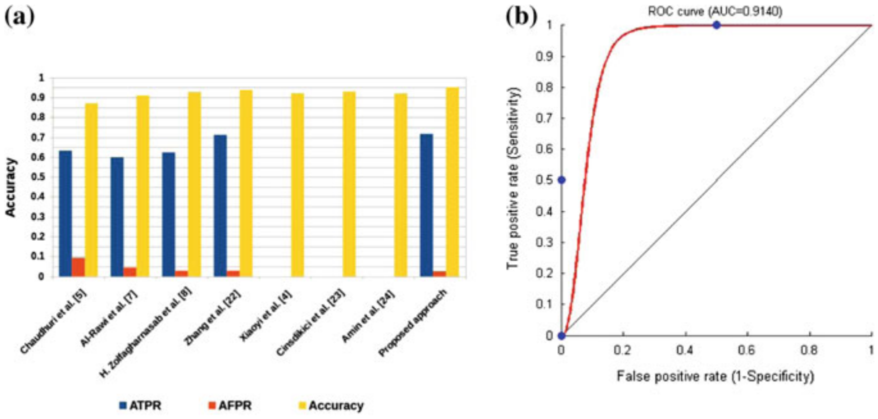


Fig. 5 **a** Graphical presentation of comparative analysis. **b** ROC curve for DRIVE data set

Therefore, to justify the effectiveness of the proposed approach, plot the ROC curve and calculate the AUC by using the ATPR and AFPR for the 20 retinal images taken from the DRIVE database as shown in Fig. 5b. The value of AUC of our proposed approach is 0.9140 which is closer to 1 that indicates the proposed Weibull PDF based-matched filter approach is good and efficient for retinal blood vessel segmentation.

4 Conclusion

An accurate retinal blood vessel segmentation is a prominent task in computer-aided diagnosis of various retinal pathologies. In this paper, we proposed a novel matched filter approach with the Weibull probability distribution function as its kernel. In the proposed approach, a popular Gaussian function used by prominent researchers [2, 3] and the Cauchy probability distribution function used by [4] could be replaced by Weibull probability distribution function and a higher accuracy obtained. The proposed approach achieved higher accuracy because, the gray scale cross-sectional profile of retinal image matched better with Weibull PDF-based kernel. In addition, to design a new matched filter an appropriate value of parameters by exhaustive experimental analysis and enhancing the retinal image in pre-processing step by using the concept of principal component analysis (PCA) and contrast limited adaptive histogram equalization (CLAHE) is selected. The performance of the proposed approach is compared with the prominent Gaussian distribution function and Cauchy PDF-based matched filter approach and achieved better performance for all 20 retinal images selected from the test set of DRIVE database.

References

1. Fraz, M.M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A.R., Owen, C.G., Barman, S.A.: Blood vessel segmentation methodologies in retinal images—a survey. Comput. Methods Program. Biomed. **108**, 407–433 (2012)
2. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Trans. Med. Imaging **8**, 263–269 (1989)
3. Al-Rawi, M., Qutaishat, M., Arrar, M.: An improved matched filter for blood vessel detection of digital retinal images. Comput. Biol. Med. **37**, 262–267 (2007)
4. Zolfagharnasab, H., Naghsh-Nilchi, A.R.: Cauchy based matched filter for retinal vessels detection. J. Med. Signals Sensors **4**, 1 (2014)
5. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 131–137 (2003)
6. Zhang, B., Zhang, L., Zhang, L., Karray, F.: Retinal vessel extraction by matched filter with first-order derivative of Gaussian. Comput. Biol. Med. **40**, 438–445 (2010)

7. Cinsdikici, M.G., Aydın, D.: Detection of blood vessels in ophthalmoscope images using MF/ant (matched filter/ant colony) algorithm. Comput. Methods Program. Biomed. **96**, 85–95 (2009)
8. Amin, M.A., Yan, H.: High speed detection of retinal blood vessels in fundus image using phase congruency. Soft Comput. **15**, 1217–1230 (2011)
9. Seo, J.W., Kim, S.D.: Novel pca-based color-to-gray image conversion. In: 2013 20th IEEE International Conference on Image Processing (ICIP), pp. 2279–2283. IEEE (2013)
10. Niemeijer, M., Staal, J.J., Ginneken, B., Loog, M., Abramoff, M.D.: DRIVE: digital retinal images for vessel extraction. In: Methods for Evaluating Segmentation and Indexing Techniques Dedicated to Retinal Ophthalmology (2004)
11. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Trans. Med. Imaging **23**, 501–509 (2004)
12. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)

Assessment of Values of Time-Domain and Frequency-Domain Parameters for ECG Signals Through HRV Analysis Using Symlets for Arrhythmia Prediction

Alok Chakrabarty, Narottam Das and Dipankar Das

Abstract In this paper, we present a work on HRV analysis of ECG signals using three time-domain parameters namely SD ratio, pNN50 and RMSSD and one frequency-domain parameter namely LF/HF ratio. For this work we obtained the ECG signal data from the MIT-BIH database. A total of 40 ECG signals, 10 each for normal sinus rhythm, for atrial fibrillation, for supra-ventricular fibrillation and for premature ventricular contraction, were used for experimentation. All ECG signals are of 30 min duration. The R-peak was detected using the Symlet5 wavelet at second level of decomposition. The accuracy of R-peak detection was found to be 97 %. Through R-peak detection and through the determination of the RR intervals, we estimated the values of mean and of standard deviation of the 4 parameters. Using the information from RR intervals, we also obtained the Poincaré plots and the power spectral density plots for the 40 signals. Based on the obtained values of the parameters we comment on the nature of values of the parameters in the paper, for normal and abnormal conditions. We further observe that by visual inspection also of the Poincaré plot and of the power spectral density plot of an unclassified ECG signal, the signal can be classified as normal or abnormal.

Keywords ECG • Heart rate variability • Heart beat • Wavelet • Symlet • RMSSD • SD ratio • LF/HF • pNN50 • Arrhythmia • QRS complex •

A. Chakrabarty (✉)
Department of Computer Science and Engineering,
National Institute of Technology Meghalaya, Bijni Complex,
Laitumkhrah, Shillong 793003, India
e-mail: mcscalok@gmail.com

N. Das
Department of Electronics and Telecommunication Engineering,
Bharati Vidyapeeth College of Engineering Lavale, Pune 412115, India
e-mail: narottam01@gmail.com

D. Das
Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700032, India
e-mail: dipankar.dipnil2005@gmail.com

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9_41

R-peak • Atrial fibrillation • Supra-ventricular • Premature ventricular contraction • Poincaré plot • Power spectral density

1 Introduction

The heart rate, or pulse, is the speed of the heartbeat which is the number of contractions of the heart per unit of time. The heart rate is typically expressed in terms of beats per minute (bpm). The heartbeat rate or the heart rate or the pulse is an inexpensive and noninvasive symptomatic medical tool for disease determination. The heart rate conveys significant information for prediction of cardiovascular, non-cardiovascular as well as psychological diseases.

With the invention of electrocardiography, the scientific and technological monitoring of heart rate has advanced. Electrocardiography is a technique of recording the bioelectric current generated by the heart muscles. The graph of voltage versus time of this recording is called the electrocardiogram (ECG). As shown in Fig. 1, a typical ECG waveform of one heart beat comprises five deflections arbitrarily named as "P" to "T" waves. Sometimes there is an additional "U" wave which follows the "T" wave. The ECG waveform of one heart beat has an initial upward deflection called P wave followed by a downward deflection called Q wave followed by a sharp upward deflection called R wave followed by a downward deflection called S wave followed by an upward deflection called T wave [1]. The Q, R and S waves occur in quick succession and together these three waves reflect the single event of depolarization of the right and left ventricles of the human heart, and thus the waves are usually considered together under the name QRS complex. The QRS complex is the central and the most visually obvious part of an ECG recording. Thus the heart rate can also be defined as the number of QRS complexes in a minute. The count and shape of the QRS complex varies between a normal person and a diseased person. The heart rate becomes slow or fast according to situations. Different types of emotions, thoughts and changes in environmental conditions, also cause immediate change in the heart rate. The instantaneous heart rate is called the 'Heart Rate Variability' (HRV) and it varies between two consecutive heart beats [2]. The analysis of HRV, called the Heart Rate Variability analysis, is a recognized technique to obtain valuable information from ECG signals that can be used for prediction of abnormal cardiac conditions [3] and other non-cardiovascular diseases. The activity of detection of QRS complexes for the identification of the R waves is referred to as R-peak detection. It is the principal activity in HRV analysis of an ECG signal, based on which the RR intervals (inter-beat, i.e., one R wave and its adjacent R wave, intervals) are estimated and based on which the values of the time-domain and the frequency-domain parameters are estimated for the ECG signal.

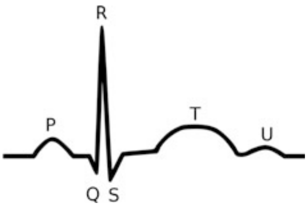


Fig. 1 A typical ECG waveform of one heart beat with P wave, QRS complex, T wave and U wave

2 The Time-Domain and the Frequency-Domain Parameters

For the HRV analysis, we used the time-domain parameters SD ratio, pNN50 and RMSSD and the frequency-domain parameter LF/HF ratio.

The SD ratio is the ratio of the length of the semi-minor axis and the length of the semi-major axis of the Poincaré plot of an ECG signal. The Poincaré plot of an ECG signal is a plot of each RR interval of the signal against its adjacent RR interval [4, 5]. Thus two consecutive RR intervals represent one point in the plot.

The parameter NN50 is the number of interval differences between adjacent RR intervals that are greater than 50 ms. The parameter pNN50 is obtained by dividing the value of NN50 by the total number of RR intervals in the ECG signal [2].

The parameter RMSSD is calculated as the square root of the mean squared differences of successive RR intervals. This parameter depends on the total number of RR intervals and the lengths of the RR intervals [2, 6].

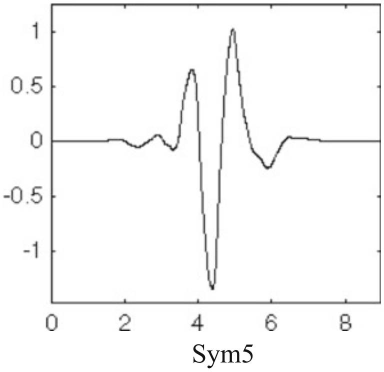
The parameter LF/HF ratio is a ratio of two spectral power values called the low-frequency (LF) power and the high-frequency (HF) power. To estimate the value of LF/HF ratio, the time series data of RR intervals are transformed from time-domain to frequency-domain using fast Fourier transformation (FFT). The value of the component LF is then obtained as the total spectral power of all RR intervals between 0.04 and 0.15 Hz and the value of the component HF is obtained as the total spectral power of all RR intervals between 0.15 and 0.4 Hz [7].

The parameter LF/HF ratio is generally not used for the analysis of long-term ECG recordings; this is why we have used SD ratio, pNN50 and RMSSD which can be used for HRV analysis of both long-term and short-term ECG recordings.

3 Experiments

The ECG data used for the experiments were obtained from the MIT-BIH database available at http://physionet.org [8]. We investigated a total of 40 ECG signals, 10 each for normal sinus rhythm (NSR), for atrial fibrillation (AF), for supra-ventricular fibrillation (SVF) and for premature ventricular contraction (PVC) from the database. Each ECG signal considered is of 30 min duration. The sampling frequency of normal sinus rhythm and supra-ventricular fibrillation is

Fig. 2 The Symlet 5 wavelet ‘Sym5’



128 Hz; 250 Hz for atrial fibrillation and 360 Hz for premature ventricular contraction. Since ECG signals can contain various types of noises such as baseline wander noise, power line interference, electrode contact noise, internal amplifier noise, muscle noise, motion artifacts, etc., the raw ECG signals were pre-processed using low-pass filters, high-pass filters and differentiators [9]. After noise removal, we decomposed every pre-processed ECG signal up to second level using the Symlet wavelet Sym5. The wavelets comprise a family of basis functions that describe signals in a localized time and frequency format. The Symlets are nearly symmetrical, orthogonal and bi-orthogonal wavelets proposed by Daubechies as modifications to the Daubechies ‘db’ family wavelets. The names of the Symlet family wavelets are written as ‘SymN’, where ‘N’ is the order, and ‘Sym’ is the surname [10]. In Fig. 2 the Symlet wavelet ‘Sym5’ is shown.

In a multiple-level wavelet decomposition process, a time-domain signal is broken down into many lower resolution components. At each level, filtering and sub-sampling is done which results in halving of the time resolution and doubling of the frequency resolution. The signal denoted by the sequence $x[n]$ is passed through several levels of low-pass and high-pass filtering. At each level of decomposition, the high-pass filter produces detailed information $g[n]$ and coarse approximations $h[n]$. Figure 3 shows a schematic diagram of the multiple-level wavelet decomposition process. Figure 4 illustrates the trend of R-peaks in normal sinus rythm signals and Fig. 5 illustrates the trend of true R-peaks and abnormal peaks in abnormal signals. We considered the second-level decomposition for better R-Peak detection yet with lesser computation [6]. By using a search algorithm and using a threshold of 0.75 × mean of all detected peaks, the R-peaks were found out with their respective locations. The detection accuracy was 97 % when we compared our R-peak counts for the 40 signals, against those available in the MIT-BIH database. The RR intervals were calculated both w.r.t. time and number of samples. Using the information from RR intervals, we also obtained the Poincaré plots and the power spectral density (PSD) plots for the 40 signals. We used the well-known MATLAB software for writing the programs.

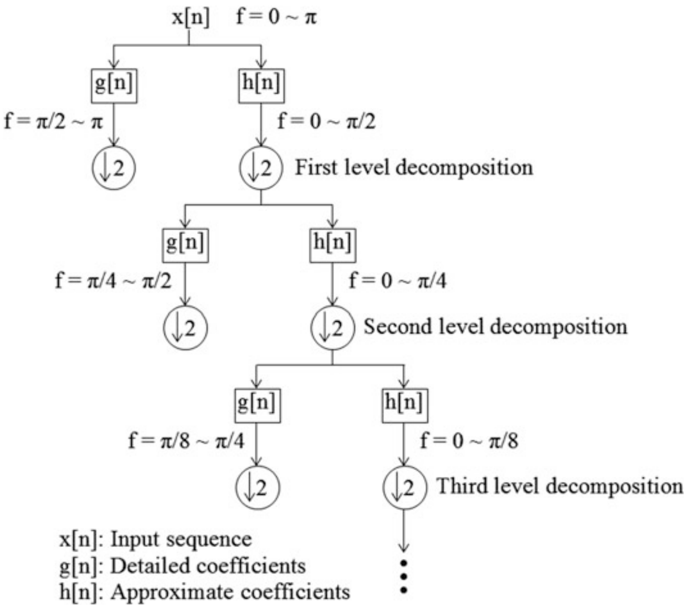


Fig. 3 Multiple-level wavelet decomposition

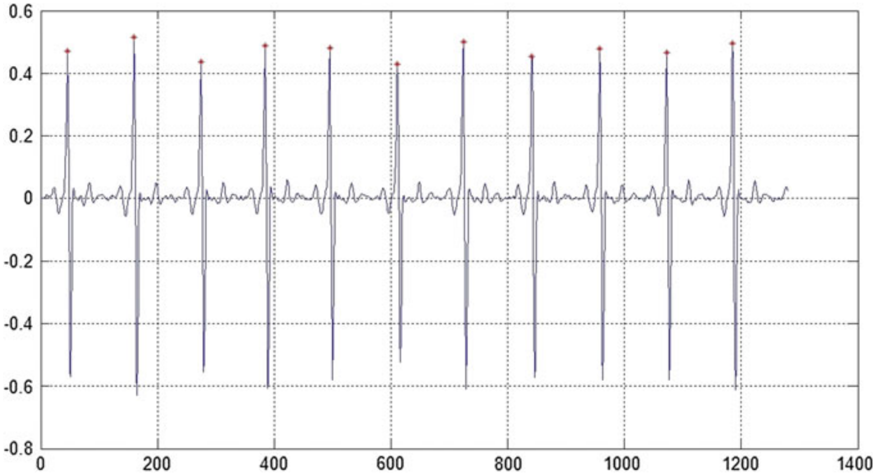


Fig. 4 R-peak detection of a sample NSR signal (R-peaks are indicated using '*' symbol)

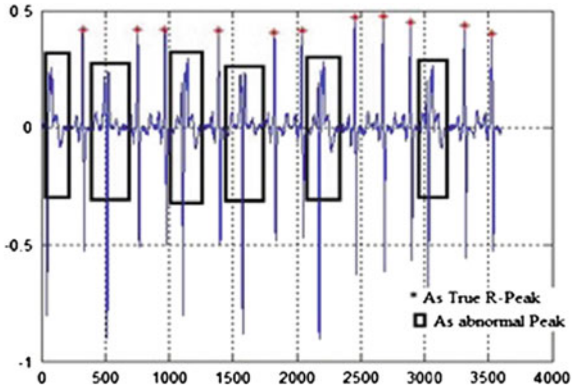


Fig. 5 R-peak detection of a sample abnormal signal (R-peaks are indicated using '*' symbol)

Table 1 Means and standard deviations for NSR

Signal type	Parameters			
	SD ratio	pNN50	RMSSD	LF/HF ratio
NSR	0.5461 ± 0.2014	7.8393 ± 1.995	4.4653 ± 1.7115	1.5868 ± 0.2645

Table 2 Means and standard deviations for AF

Signal type	Parameters			
	SD ratio	pNN50	RMSSD	LF/HF ratio
AF	0.9767 ± 0.0263	14.403 ± 6.268	10.8201 ± 3.0778	0.8699 ± 0.0435

Table 3 Means and standard deviations for SVF

Signal type	Parameters			
	SD ratio	pNN50	RMSSD	LF/HF ratio
SVF	0.8637 ± 0.0866	32.430 ± 9.358	7.004 ± 0.9067	0.9568 ± 0.1131

Table 4 Means and standard deviations for PVC

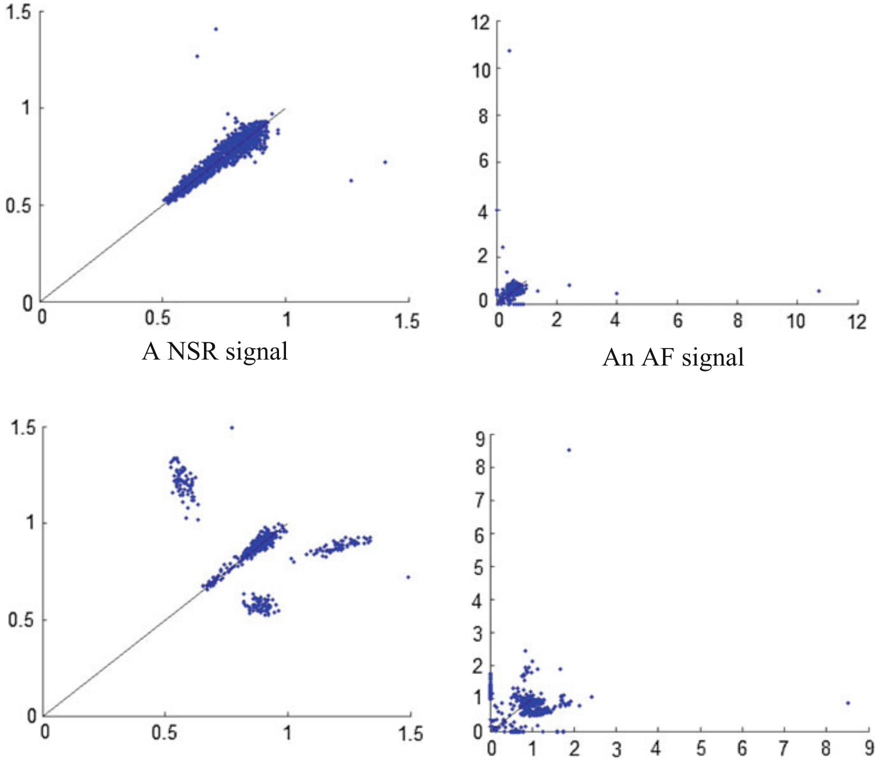
Signal type	Parameters			
	SD ratio	pNN50	RMSSD	LF/HF ratio
PVC	1.4059 ± 0.4602	61.792 ± 8.960	19.5751 ± 2.4868	0.7188 ± 0.1220

4 Results

Tables 1, 2, 3 and 4 show the values of mean and of standard deviation of the time-domain parameters SD ratio, pNN50 and RMSSD and the frequency-domain parameter LF/HF ratio obtained by HRV analysis of the 40 classified ECG signals obtained from the MIT-BIH database [11]. Figure 6 presents some Poincaré plots and Fig. 7, some PSD plots.

4.1 Plots

See Figs. 6 and 7.



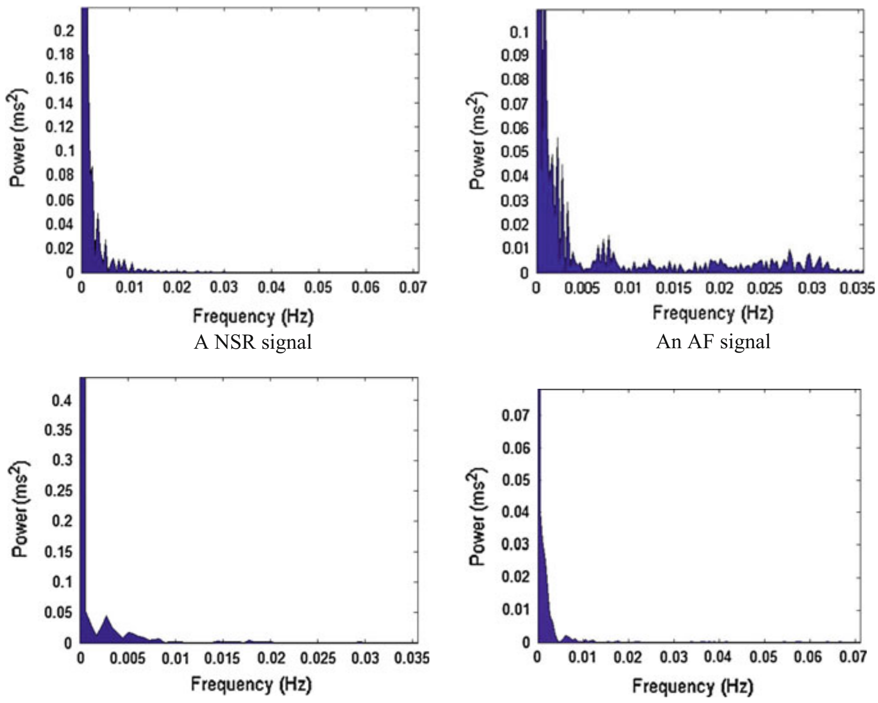
A NSR signal

An AF signal

A SVF signal

A PVC signal

Fig. 6 Poincaré plots



A NSR signal

An AF signal

A SVF signal

A PVC signal

Fig. 7 PSD plots

4.2 Tables

See Tables 1, 2, 3 and 4.

5 Conclusion

In this work, we obtained the values of mean and standard deviation of time-domain parameters such as like SD ratio, pNN50 and RMSSD and a frequency-domain parameter named LF/HF ratio which are useful parameters for HRV analysis of ECG signal. The HRV analysis of ECG signals helps in the prediction of abnormal heart conditions and of other non-cardiovascular diseases also. In this work, our aim of HRV analysis was related to prediction of the common cardiac ailment known as arrhythmia. We obtained the values of mean and standard deviation of the HRV analysis parameters by studying 40 classified ECG signals of the MIT-BIH database. From the obtained values in Tables 1, 2, 3 and 4, we can propose that the SD ratio should be higher for abnormal signals than for normal sinus rhythm. For pNN50 we comment that its value should be least for NSR. The value of RMSSD should be higher for abnormal signals than for NSR. The value of LF/HF ratio should be lower for abnormal signals compared to that for NSR. We also obtained Poincaré plots and power spectral density plots from the data collected through the HRV analysis. We make this comment that the Poincaré plots should look elliptical for NSR and scattered for abnormal signals. Regarding PSD plots, we comment that in PSD plots for NSR the maximum power distribution should be visible at low frequencies; whereas, for abnormal signals power distribution can be visible at higher frequencies also. Thus we can say that by visual interpretation also of the Poincaré plots and the PSD plots of ECG signals, ECG signals can be classified as normal or as abnormal.

References

1. Afonso, V.X., Tompkins, W.J., Nguyen, T.Q., Luo, S., et al.: ECG beat detection using filter banks. IEEE Trans. Biomed. Eng. **46**(2), 192–202 (1999)
2. Task Force of the European Society of Cardiology and The North American society of Pacing and Electrophysiology: Heart rate variability standards of measurement, physiological interpretation, and clinical use. Eur. Heart J. **17**, 354–381 (1996)
3. Taggart, P.: Arrhythmias in the normal human heart. In: Williams, E.M.V. (ed.) Antiarrhythmic drugs. Handbook of Experimental Pharmacology, vol. 89, pp. 279–301. Springer, Berlin, Heidelberg (1989)
4. D'Addio, G., Pinna, G., Maestri, R., Corbi, G., Ferrara, N., Rengo, F., et al.: Quantitative Poincaré plots analysis contains relevant information related to heart rate variability dynamics of normal and pathological subjects. Comput. Cardiol. **2004**, 457 (2004)
5. Piskorski, J., Guzik, P.: Filtering Poincaré plots. Comput. Methods Sci. Technol. **11**(1), 39–48 (2005)
6. Sifuzzaman, M., Islam, M.R., Ali, M.Z., et al.: Application of wavelet transform and its advantage compared to fourier transform. J. Phys. Sci. **13**, 121–134 (2009)
7. Miličević, G.: Low to high frequency ratio of heart rate variability spectra fails to describe sympatho-vagal balance in cardiac patients. Collegium Antropol. **29**(1), 295–300 (2005)
8. Moody, G.B., Mark, R.G., Glodberger, A.L., et al.: PhysioNet: a research resource for studies of complex physiologic and biomedical signals. Comput. Cardiol. **2000**, 179 (2000)
9. Pan, J., Tomkins, W.J.: A real-time QRS detection algorithm. IEEE Trans. Biomed. Eng. BME **32**(3), 230–236 (1985)
10. Chavan, M.S., Mastorakis, N., Chavan, M.N., Gaikwad, M.S., et al.: Implementation of SYMLET wavelets to removal of Gaussian additive noise from speech signal. In: Bojkovic, Z. S. (ed.) Proceedings of Recent Researches in Communications, Automation, Signal Processing, Nanotechnology, Astronomy and Nuclear Physics: 10th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications (EHAC'11), Cambridge, Feb 2011, p. 37. WSEAS Press (2011)
11. Mainardi, L.T.: On the quantification of heart rate variability spectral parameters using time–frequency and time-varying methods. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **367**(1887), 255–275 (2009)

Author Index

S.K. Sahana and S.K. Saha (eds.), *Advances in Computational Intelligence*,
Advances in Intelligent Systems and Computing 509,
DOI 10.1007/978-981-10-2525-9