Springer Tracts in Nature-Inspired Computing

Santosh Kumar Das
Thanh-Phong Dao
Thinagaran Perumal *Editors*

Nature-Inspired Computing for Smart Application Design

Springer

Springer Tracts in Nature-Inspired Computing

The book series is aimed at providing an exchange platform for researchers to summarize the latest research and developments related to nature-inspired computing in the most general sense. It includes analysis of nature-inspired algorithms and techniques, inspiration from natural and biological systems, computational mechanisms and models that imitate them in various fields, and the applications to solve real-world problems in different disciplines. The book series addresses the most recent innovations and developments in nature-inspired computation, algorithms, models and methods, implementation, tools, architectures, frameworks, structures, applications associated with bio-inspired methodologies and other relevant areas.

The book series covers the topics and fields of Nature-Inspired Computing, Bio-inspired Methods, Swarm Intelligence, Computational Intelligence, Evolutionary Computation, Nature-Inspired Algorithms, Neural Computing, Data Mining, Artificial Intelligence, Machine Learning, Theoretical Foundations and Analysis, and Multi-Agent Systems. In addition, case studies, implementation of methods and algorithms as well as applications in a diverse range of areas such as Bioinformatics, Big Data, Computer Science, Signal and Image Processing, Computer Vision, Biomedical and Health Science, Business Planning, Vehicle Routing and others are also an important part of this book series.

The series publishes monographs, edited volumes and selected proceedings.

More information about this series at http://www.springer.com/series/16134

Santosh Kumar Das · Thanh-Phong Dao ·
Thinagaran Perumal
Editors

Nature-Inspired Computing for Smart Application Design

Springer

Editors
Santosh Kumar Das
Department of Computer Science
and Engineering
National Institute of Science
and Technology (Autonomous)
Berhampur, Odisha, India

Thanh-Phong Dao
Division of Computational Mechatronics
Institute for Computational Science
Ton Duc Thang University
Ho Chi Minh City, Vietnam

Thinagaran Perumal
Department of Computer Science
Faculty of Computer Science
and Information Technology
Universiti Putra Malaysia
Seri Kembangan, Malaysia

ISSN 2524-552X ISSN 2524-5538 (electronic)
Springer Tracts in Nature-Inspired Computing
ISBN 978-981-33-6194-2 ISBN 978-981-33-6195-9 (eBook)
https://doi.org/10.1007/978-981-33-6195-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Several intelligent systems have been developed over the last few decades based on user needs and requirements. The need for regular usage is increased rapidly in all the criteria and specifications. The implementations are focused entirely on problems and real-life issues. Each application design is connected to one or more techniques based on the derived problem parameters. Some parameters are accurate, or some of them are not accurate depending on application design. Therefore, several natural-inspired computing techniques are applied to develop the method of smart design and are reflected in various areas in this book.

Objective of the Book

This book contains some smart design solutions based on different areas by using nature-inspired computing. It deals with modelling, analysis, design, and enhancement of different modules of the smart design. The main aim of this book is to enhance the applications of smart design by reducing complexity.

Organization of the Book

The book contains 13 chapters that are organized into three parts as follows. Before start the parts, Chapter "Smart Design and Its Applications: Challenges and Techniques" describes the overview of smart design and its related challenges, issues and applications. The first part contains four chapters that outline the modelling of smart city and environment. The second part contains four chapters that highlight some modelling of intelligent networking. The third part contains four chapters that outline the modelling of smart security and management based on different areas.

City and Environment

This part outlines some concepts that help in designing smart city and environment based on some areas. Short descriptions of these chapters are as follows.

Chapter "Automatic Generation Control Scheme for Power Quality Improvement of Multi-source Power Generating System with Secondary Controller Optimization Using Parameter-Setting-Free Harmony Search"

This chapter outlines an automatic generation control of single-area multi-source power generating system that is examined by applying proportional integral derivative controller as a secondary controller loop. This method uses the fusion of some nature-inspired techniques such as particle swarm optimization and harmony search.

Chapter "Environmental Sound Classification Using Neural Network and Deep Learning"

This chapter outlines the features of input audio signal which is extracted and used for training the neural network using deep learning algorithm for classifying environmental noise. This proposed method uses the fusion of artificial neural network, deep learning and Bayesian neural network for the purpose of smart output.

Chapter "Feature Selection Method Using CFO and Rough Sets for Medical Dataset"

This chapter outlines an efficient feature selection technique proposed by the fusion of central force optimization and rough set for the medical dataset. The combination of both methods helps to optimize the several parameters and establish the relationship between noisy and imprecise information for outperforming the results.

Chapter "Fuzzy-Based Optimal Solution for Minimization of Loss of Company Based on Uncertain Environment"

This chapter outlines an optimal solution to minimizing the losses of the company in an uncertain environment. In this method, the loss of the company is minimized by reducing uncertainty in the information with the fusion of quadratic programming and fuzzy logic. The combination of both helps to model the objective with related constraints and reduce imprecise information.

Intelligent Networking

This part outlines some concepts for designing intelligent networking for solving real-life issues and problems based on some variation of the wireless network. Short descriptions of these chapters are as follows.

Chapter "Impacts of Computational Techniques for Wireless Sensor Networks"

This chapter outlines some swam intelligence techniques that are compared with the context of the objective for giving a better assessment. This chapter uses some novel algorithms for intelligent networking with the context of computational intelligence and swarm intelligence for making the ad-hoc and sensor network more intelligent.

Chapter "Efficient Node Deployment Based on Immune-Inspired Computing Algorithm for Wireless Sensor Networks"

This chapter outlines two centralized energy-efficient deployment algorithms based on multi-objective immune algorithm for optimizing the trade-off between the network coverage and the energy cost. In this work, the first deployment algorithm is known as an immune-based node deployment algorithm.

Chapter "An Efficient Routing in Wireless Sensor Network: An Application of Grey Wolf Optimization"

This chapter outlines the investigation of wireless sensor network for determining an optimal path from source node to destination node. It is used to minimize energy consumption during data transmission. The proposed problem is formulated by the fusion of mixed-integer programming and simple branch and bound methods to making the network intelligent.

Chapter "Coverage Optimization using Nature-Inspired Algorithm for Directional Sensor Networks"

This chapter outlines the studies of various directional sensor network from the coverage point of view. It helps in directional sensing and coverage adjustment. It helps in the environment classified as target and barrier base coverage based on particle swarm optimization and memetic algorithm. The combination of both helps in efficient coverage optimization for the directional sensor network.

Security and Management

This part outlines some concepts of several types of security and management techniques based on the smart design in several applications. Short descriptions of these chapters are as follows.

Chapter "Flower Pollination Optimization-Based Security Enhancement Technique for Wireless Sensor Network"

This chapter outlines an efficient security enhancement and modelling technique for wireless sensor network by using flower pollination optimization. The proposed method uses several techniques for enhancing the network such as cryptography, encryption, decryption, data encryption standard, advanced encryption standard, and Rivest–Shamir–Adleman. The combination of all the stated techniques helps to model a robust and intelligent security technique in the wireless sensor network.

Chapter "Fuzzy Quadratic Programming Based Conflicting Strategy Management Technique for Company"

This chapter outlines an efficient strategy management technique for the company using quadratic programming. The quadratic programming plays the role of mathematical optimization based on the desire objective function along with constraints. Fuzzy logic is used to make the quadratic programming flexible which is used to maintain variations of the customer requirements and demands efficiently.

Chapter "A Novel Multilevel Classifier Hybrid Model for Intrusion Detection Using Machine Learning"

This chapter outlines a novel multi-level classifier hybrid model for intrusion detection using machine learning. This model contains two phases: In the first phase, the random tree classifier classifies the dataset into known attacks using the misuse detection approach, and the second phase classifies the novel attacks using the anomaly detection approach.

Chapter "Maintaining Manpower in Technical College Using Fusion of Quadratic Programming and Fuzzy Logic"

This chapter outlines a mathematical optimization technique by using quadratic programming and fuzzy logic for optimizing manpower of the technical college. In this work, quadratic programming is used to model the objective function in nonlinear formulation based on some constraints of manpower. Fuzzy logic is used to reduce the uncertainty of the problem by reducing imprecise information.

Berhampur, India — Santosh Kumar Das

Ho Chi Minh City, Vietnam — Thanh-Phong Dao, Assistant Professor

Selangor, Malaysia — Thinagaran Perumal, Assistant Professor

List of Reviewers

Abhishek Kumar, Swami Vivekananda Subharti University, Meerut
Amalendu Rana, NIST, Berhampur, Odisha
Amitesh Kumar Pandit, Dr. Rammanohar Lohia Avadh University, Ayodhya, Uttar Pradesh
Arun Prasad Burnwal, GGSESTC, Bokaro, Jharkhand
Ashish Kumar Dass, National Institute of Science and Technology, Brahmapur, Odisha
Chandan Kumar Shiva, S. R. Engineering College, Ananthsagar, Hasanparthy, Warangal, Telangana
Jaydeb Mondal, NIST, Berhampur, Odisha
Jayraj Singh, Indian Institute of Technology (ISM), Dhanbad
Mahesh Chaurasia, Dr. Rammanohar Lohia Avadh University, Ayodhya, Uttar Pradesh
Manoj Kumar Mandal, Jharkhand Rai University, Ranchi
Nabajyoti Mazumdar, Central Institute of Technology, Kokrajhar, Assam
Nabanita Dutta, VIT University, Vellore, Tamil Nadu
Priyanka Jaiswal, Indian Institute of Technology (ISM), Dhanbad
Ruchika Padhi, National Institute of Science and Technology, Brahmapur, Odisha
Sagar Samal, National Institute of Science and Technology, Brahmapur, Odisha
Shalini Mahato, B.I.T, Mesra, Ranchi, Jharkhand
Siba Prasada Tripathy, National Institute of Science and Technology, Brahmapur, Odisha
Smita Rani Sahu, BPUT, Odisha
Soumen Nayak, SOA University, Bhubaneswar
Sourav Samanta, University Institute of Technology, BU, Burdwan, W.B.
Subhra Priyadarshini Biswal, National Institute of Science and Technology, Brahmapur, Odisha
Susmita Mahato, NIST, Berhampur, Odisha
Vishwas Mishra, Swami Vivekanand Subharti University, Meerut, U.P.

Contents

Editors and Contributors

About the Editors

Santosh Kumar Das received his Ph.D. degree in Computer Science and Engineering from Indian Institute of Technology (ISM), Dhanbad, India, in 2018, and completed his M.Tech. degree in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology (erstwhile WBUT), West Bengal, India, in 2013. He is currently working as Assistant Professor at the School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Institute Park, Pallur Hills, Berhampur, Odisha, India. He has more than eight years teaching experience. He has authored/edited three books with Springer in series as Lecture Notes in Networks and Systems, Tracts in Nature-Inspired Computing, and Studies in Computational Intelligence. He has contributed more than 27 research papers. His research interests mainly focus on ad hoc & sensor network, artificial intelligence, soft computing, and mathematical modelling. His h-index is 13 with more than 480 citations.

Thanh-Phong Dao is currently Assistant Professor at the Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam. He received his B.S. degree in Mechanical Engineering from the Ho Chi Minh City University of Technology and Education, Vietnam, in 2008. He received his M.S. and Ph.D. degree in Mechanical Engineering from the National Kaohsiung University of Applied Sciences, Taiwan, ROC, in 2011 and 2015, respectively. His research interests include compliant mechanism, assistive technology and rehabilitation, computational intelligence, soft computing, and optimization algorithm.

Thinagaran Perumal is the recipient of 2014 Early Career Award from IEEE Consumer Electronics Society for his pioneering contribution in the field of consumer electronics. He completed his Ph.D. at Universiti Putra Malaysia, in the area of smart technology and robotics. He is currently an Associate Professor at the Department of Computer Science, Faculty of Computer Science and Information

Technology, Universiti Putra Malaysia. He is also currently appointed as Chair of IEEE Consumer Electronics Society for Malaysia Chapter. His research interests are towards interoperability aspects of smart homes and Internet of things (IoT), wearable computing, and cyber-physical systems. Thinagaran is also heading the National Committee on Standardization for IoT (IEC/ISO TC/G/16) as Chairman since 2018. Some of the eminent works include proactive architecture for IoT systems; development of the cognitive IoT frameworks for smart homes; and wearable devices for rehabilitation purposes. He is an active member of IEEE Consumer Electronics Society and its Future Directions Committee on Internet of Things. He has been invited to give several keynote lectures and plenary talk on Internet of things in various institutions and organizations internationally. He has published several papers in IEEE conferences and journals and is serving as a TPC member for several reputed IEEE conferences. He is an active reviewer for IEEE Internet of Things Journal, IEEE Communication Magazine, IEEE Sensors Journal, and IEEE Transaction for Automation Science and Engineering, to name a few.

Contributors

Mohammed Abo-Zahhad Electrical and Electronics Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt;
Department of Electronics and Communications Engineering, E-JUST-Alexandria, New Borg El Arab, Egypt

Rajeev Agrawal GL, BAJAJ Institute of Technology and Management, Greater Noida, India

Hifzan Ahmad A.P.J. Abdul Kalam Technical University (AKTU), Lucknow, India

B. Anand Department of EIE, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India

Haider Banka Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India

Samiran Bera Indian Institute of Technology (Indian School of Mines), Dhanbad, India

Arun Prasad Burnwal Department of Mathematics, GGSESTC, Bokaro, Jharkhand, India

Santosh Kumar Das School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Berhampur, Odisha, India;
National Institute of Science and Technology, Behrampur, India

Kumar Dixit Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India

Sunil Gautam Department Engineering and Computing, Institute of Advanced Research, Gandhinagar, India

Joydev Ghosh School of Computer Science and Robotics, National Research Tomsk Polytechnic University (TPU), Tomsk, Russia

Ramesh Kumar Huda Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India

K. Jagatheesan Department of EEE, Paavai Engineering College, Namakkal, Tamil Nadu, India

Narendra Kohli Harcourt Butler Technical University, Kanpur, India

Abhishek Kumar Department of Electronics & Communication Engineering, Swami Vivekananda Subharti University, Meerut, India

Ranjit Kumar Indian Institute of Technology (Indian School of Mines) Dhanbad, Dhanbad, Jharkhand, India;
GCET, Greater Noida, India

Ritesh Kumar Department of Electrical and Electronics Engineering, SR Engineering College, Ananthsagar, Hasanparthy, Warangal, Telangana, India

Dinesh Kumar Sah Indian Institute of Technology (ISM), Dhanbad, India

Sheila Mahapatra Department of Electrical and Electronics Engineering, Alliance University, Bangalore, India

B. K. Mahatha Amity School of Engineering and Technology, Amity University Jharkhand, Ranchi, Jharkhand, India

Manoj Kumar Mandal Department of Mathematics, Jharkhand Rai University, Ranchi, India

S. Mini National Institute of Technology Goa, Goa, India

Vishwas Mishra Department of Electronics & Communication Engineering, Swami Vivekananda Subharti University, Meerut, India

Swarnavo Mondal Department of Computer Science and Engineering, University Institute of Technology, BU, Burdwan, West Bengal, India

Hari Om Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India

Trilochan Panigrahi National Institute of Technology Goa, Goa, India

Saurav Raj Department of Electrical Engineering, Institute of Chemical Technology Mumbai, Marathwada Campus, Jalna, India

Dharma Rane National Institute of Technology Goa, Goa, India

Nabil Sabor Electrical and Electronics Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt;
Department of Electronics and Communications Engineering, E-JUST-Alexandria, New Borg El Arab, Egypt

Sourav Samanta Department of Computer Science and Engineering, University Institute of Technology, BU, Burdwan, West Bengal, India

Nikhil Saxena University of Cincinnati, Cincinnati, OH, USA

Soumadip Sen Department of PGDM, International Management Institute, Kolkata, West Bengal, India

Aditya Sharma Institute of Nanoengineering and Microsystems, National Tsing Hua University, Hsinchu, Taiwan, ROC

Pushkar Shirodkar National Institute of Technology Goa, Goa, India

Chandan Kumar Shiva Department of Electrical and Electronics Engineering, SR Engineering College, Ananthsagar, Hasanparthy, Warangal, Telangana, India

Chaya Shivalingagowda Kalsekar Engineering College, New Panvel Mumbai and GITAM university, Vizag, India

Sachin Tripathi Indian Institute of Technology (Indian School of Mines) Dhanbad, Dhanbad, Jharkhand, India

Basetti Vedik Department of Electrical and Electronics Engineering, SR Engineering College, Ananthsagar, Hasanparthy, Warangal, Telangana, India

Smart Design and Its Applications: Challenges and Techniques

Santosh Kumar Das

1 Introduction

In the last few decades, the people's requirements have increased exponentially where each of the requirement consists of some intelligent techniques that help solve several types of issues which may be same or different types. Smart design is used in several applications where each application is based on real-life scenarios such as smart cities, smart environment, smart water, smart metering, smart security and emergency, smart agriculture, smart industry control, eHealth smart home automation system, etc. There are several challenges for smart design; some of which are discussed as follows.

(i) **Infrastructure**: Infrastructure is the primary framework for designing any smart technique or device consisting of several components based on some features and behaviours. It gives the organization structure that helps to drive the system in real-life scenarios based on the applications.
(ii) **Privacy**: Privacy indicates the confidentiality of the information or system, like if the system or device is smart; it requires to maintain more privacy system. In the modern era, most of the devices or systems are intelligent or smart. However, it has several security issues with data and information because it is accessed by the global system to increase flexibility.
(iii) **Usability**: Usability is one of the challenges for smart design because most people are unaware of several technologies such as machine learning, artificial intelligence, soft computing, and data science, etc. It creates several types of difficulties for the person who wants to use these technologies or the device. So,

S. K. Das (✉)
School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Institute Park, Pallur Hills, Berhampur, Odisha 761008, India
e-mail: sunsantosh2014@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*,
Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_1

some knowledge is required for using this technology or handling this system for the user's easiness.

(iv) **Affordability**: It indicates the cost-related information that handles by the company or user or uses it for deployment in the market. If the cost of the product is very high or expensive, then it cannot be affordable. Its cost depends on the total designing cost, including several other types of expenses. Hence, the overall cost should be optimal that can affordable by the regular user or company.

Moreover, several challenges are available for smart design with the market and the real-life scenarios. Some of the challenges are discussed in this section. However, some of the other challenges are available in the literature mentioned in [1–3]. The stated works are based on several types of issues and challenges regarding the smart and innovative design and application. This is an introduction chapter that highlights some of the smart design based on nature-inspired computing, along with their essential elements, those are inherent in this smart design.

2 Some Applications for Smart Design

There are several applications and uses of smart design in our real life with respect to daily life, market, business, and any other aspect. In this section, some of the applications are discussed, which are covered in this book. The names of these applications are: city and environment, intelligent networking, security, and management. Descriptions of these applications are as follows.

2.1 City and Environment

Smart cities indicate a fusion of several intelligent methods in real-life applications, such as a smart road that indicates some alert systems in the road that highlight some precautions to the person and vehicle. It is also added to the modern vehicular networking including the smart traffic management, public transportation, parking spaces, etc., in the smart city and environment module which prove to be offering several advantages. Hence, this smart city and environment is helpful for automated generation control in any situation. There are several power quality management techniques that include multi-source, for example, handling load balancing, environment management, medical diagnosis using smart devices, and intelligent doctors which includes several feature selection techniques which are based on nature-inspired techniques. It also includes several strategic management techniques that help businesses grow and also help the customers.

2.2 Intelligent Networking

Intelligent networking is a smart networking technique, which is the fusion of several types of networking that overcomes the limitation of the traditional network over the context of several parameters. Several networks take part in this networking, such as wireless sensor network (WSN), wireless ad hoc network (WANET), mobile ad hoc network (MANET), vehicular ad hoc network (VANET), and hybrid ad hoc network (HANET). Sometimes, the combination of WANET and WSN is known as ad hoc & sensor networks. In this book, several works are presented based on the stated network for handling several types of issues and applications such as routing, node deployment, network lifetime enhancement, coverage, multi-path, and multi-cast. Many works are proposed based on several intelligent networking sites, such as in [4]; some applications are discussed in WSN based on underground coal mines and renewable energy harvesting systems in WSN [5, 6]. In [7], the author discusses some challenges and attacks in MANET. In [8], a systematic review is discussed based on VANET with the routing protocol context. In [9, 10], some works are presented based on delay-tolerant networks for modelling the network's strategy. Many works are based on wireless communication based on intelligent features that help in real life for every user and customer [11, 12]. The mentioned networks' smart techniques help in easy installation and maintenance, reduce the business costs, provide higher-level security, easily customization, 24-h accessibility, etc.

2.3 Security and Management

Security and management are significant paradigms for designing the smart device and making the system intelligent. There are several frauds and dynamic mobility. The intelligence behaviour creates several types of imprecise parameters that raise several types of uncertainties. So, there is a need for efficient security management during the design of a smart device or technique. The proposed book is based on a nature-inspired computing-based smart application design. It focuses on the company's strategic management so that the company will get benefits, and customers also get benefit. Security enhancement of several networks is also a part of this section. Moreover, it also covers several intrusion detection techniques and workforce management techniques for a technical college. Numerous works are proposed in the references: security and management for enhancing the smart applications [13–18].

3 Some Techniques for Smart Design

Smart design is based on smart and intelligent techniques because it has robust, reliable, efficient, and dynamic features. These features are available in several modern techniques such as swarm intelligence, based on swarm application and features of the nature-inspired algorithm. It has several examples such as particle swarm optimization, harmony search, dragonfly algorithm, grey wolf optimization, and flower pollination optimization. Some of the techniques are based on biological systems that also behave like nature-inspired techniques such as artificial neural networks and Bayesian neural networks. Due to the design's complexity, the decision-maker or researcher uses deep learning and a memetic algorithm in some situations. Some algorithms use multi-objective optimization to handle more than one objective simultaneously; it may be multi-objective linear or non-linear optimization or multi-objective immune algorithms [19].

Moreover, central force optimization, atom search optimization, machine learning, random tree, fuzzy logic, etc., are some other techniques used for the optimization. Now, information retrieval is also used for smart information retrieval for the local language with machine learning. Several works are proposed [20, 21] which mention that these techniques are used to solve some of the issues or problems to make the system or device as smart as possible. Each technique is based on nature-inspired computing mentioned in this book to design a smart design for real-life applications.

4 Conclusions

In this chapter, several usages and applications are discussed with the context of smart design. It also includes several challenges and issues related to smart design. This smart design may be any machine, device, or technique that helps in society. Each of the smart designs or techniques uses one or more intelligent techniques discussed in the sections mentioned above. Without any intelligent technique, smart applications cannot be designed. These intelligent techniques are mentioned named in the chapter: nature-inspired computing for solving the issues of real-life applications. Each of the nature-inspired techniques has its driving principle in running the algorithm and solving real-life issues discussed later in this book based on several areas.

References

1. Panteli, C., Kylili, A., Fokaides, P. A. (2020). Building information modelling applications in smart buildings: From design to commissioning and beyond A critical review. *Journal of Cleaner Production*, 121766 https://doi.org/10.1016/j.jclepro.2020.121766.
2. Nižetić, S., Šolić, P., González-de, D. L. D. I., & Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production, 274,* 122877. https://doi.org/10.1016/j.jclepro.2020.122877.
3. Xu, Y., Yan, C., Liu, H., Wang, J., Yang, Z., & Jiang, Y. (2020). Smart energy systems: A critical review on design and operation optimization. *Sustainable Cities and Society*, 102369. https://doi.org/10.1016/j.scs.2020.102369.
4. Muduli, L., Mishra, D. P., & Jana, P. K. (2018). Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review. *Journal of Network and Computer Applications, 106,* 48–67.
5. Sah, D. K., & Amgoth, T. (2020). Renewable energy harvesting schemes in wireless sensor networks: A survey. *Information Fusion, 63,* 223–247.
6. Sah, D. K., & Amgoth, T. (2020). A novel efficient clustering protocol for energy harvesting in wireless sensor networks. *Wireless Networks, 26,* 4723–4737. https://doi.org/10.1007/s11276-020-02351-x.
7. AlRubaiei, M., sh Jassim, H., Sharef, B. T., Safdar, S., Sharef, Z. T., & Malallah, F. L. (2020). Current vulnerabilities, challenges and attacks on routing protocols for mobile ad hoc network: a review. In *Swarm Intelligence for Resource Management in Internet of Things* (pp. 109–129).
8. Dua, A., Kumar, N., & Bawa, S. (2014). A systematic review on routing protocols for vehicular ad hoc networks. *Vehicular Communications, 1*(1), 33–52.
9. Singh, A. K., Bera, T., & Pamula, R. (2018, March). PRCP: Packet replication control based prophet routing strategy for delay tolerant network. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)* (pp. 1–5). IEEE.
10. Singh, A. K., & Pamula, R. (2020). An efficient and intelligent routing strategy for vehicular delay tolerant networks. *Wireless Networks*, 1–18.
11. Yang, W., Wang, X., Song, X., Yang, Y., & Patnaik, S. (2018). Design of intelligent transportation system supported by new generation wireless communication technology. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications* (pp. 715–732). IGI Global.
12. Jat, D. S., Bishnoi, L. C., & Nambahu, S. (2018). An Intelligent Wireless QoS Technology for Big Data Video Delivery in WLAN. *International Journal of Ambient Computing and Intelligence (IJACI), 9*(4), 1–14.
13. Khan, F. A., Asif, M., Ahmad, A., Alharbi, M., & Aljuaid, H. (2020). Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development. *Sustainable Cities and Society, 55,* 102018. https://doi.org/10.1016/j.scs.2020.102018.
14. Xiong, M., Chen, D., Chen, J., Chen, J., Shi, B., Liang, C., & Hu, R. (2019). Person re-identification with multiple similarity probabilities using deep metric learning for efficient smart security applications. *Journal of Parallel and Distributed Computing, 132,* 230–241.
15. Singh, A. K., & Pamula, R. (2018, February). IRS: Incentive based routing strategy for socially aware delay tolerant networks. In *2018 5th international conference on signal processing and integrated networks (SPIN)* (pp. 343–347). IEEE.
16. Chandrakar, P. (2019). A secure remote user authentication protocol for healthcare monitoring using wireless medical sensor networks. *International Journal of Ambient Computing and Intelligence (IJACI), 10*(1), 96–116.
17. Bera, S., Chattopadhyay, M., & Dan, P. K. (2018). A two-stage novel approach using centre ordering of vectors on agglomerative hierarchical clustering for manufacturing cell formation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 232*(14), 2651–2662.

18. Jayakumar Loganathan, J., & Subbiah, J. (2020). Energy aware dynamic mode decision for cellular D2D Communications by using integrated multi-criteria decision making model. *International Journal of Ambient Computing and Intelligence, 11*(3), 7 February 2020, IGI Global.
19. Khosravy, M., Gupta, N., Patel, N., & Senjyu, T. (Eds.). (2020). *Frontier Applications of Nature Inspired Computation*. Berlin: Springer.
20. Rasheed, I., Banka, H., & Khan, H. M. (2020). A hybrid feature selection approach based on LSI for classification of Urdu text. In *Machine Learning Algorithms for Industrial Applications* (pp. 3–18). Cham: Springer.
21. Rasheed, I., Gupta, V., Banka, H., & Kumar, C. (2018, September). Urdu text classification: A comparative study using machine learning techniques. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)* (pp. 274–278). IEEE.

City and Environment

Automatic Generation Control Scheme for Power Quality Improvement of Multi-source Power Generating System with Secondary Controller Optimization Using Parameter-Setting-Free Harmony Search

K. Jagatheesan, B. Anand, Soumadip Sen, Swarnavo Mondal, and Sourav Samanta

1 Introduction

The power balance between total load demands with power generation keeps the quality of generated power and stability of power generating system. It can be obtained with help of implementing load frequency control (LFC)/automatic generation control (AGC) scheme in power system. The role of LFC/AGC scheme in power generating system is to keep system stability and quality of power during emergency loading circumstance. In power system, two control loops are available such as primary and secondary control loops. In these, primary control is a slow control

K. Jagatheesan
Department of EEE, Paavai Engineering College, Namakkal, Tamil Nadu, India
e-mail: jaga.ksr@gmail.com

B. Anand
Department of EIE, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India
e-mail: b_anand_eee@yahoo.co.in

S. Sen
Department of PGDM, International Management Institute, Kolkata, West Bengal, India
e-mail: soumadip.95@gmail.com

S. Mondal (✉) · S. Samanta
Department of Computer Science and Engineering, University Institute of Technology, BU, Burdwan, West Bengal, India
e-mail: swarnavog1@gmail.com

S. Samanta
e-mail: sourav.uit@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_2

loop, and it will keep system stability and quality of power at nominal loading situations only. When higher load demand occurs in system, the control loop generated control signal is not sufficient. At this time, secondary control loop generates the required signal to keep system operation in the predetermined value. From the literature survey, it clearly shows that several controllers are designed and implemented in power system successfully. Simultaneously, several optimization techniques for tuning of controller gain values are utilized for secondary controller gain values tuning for getting desired good power quality.

Improved sine cosine algorithm (SCA)-based adaptive fuzzy-aided proportional–integral–derivative (AFPID) controller is proposed for LFC of an autonomous power generation system (fuel cells along with energy storing units, diesel energy generator, wind turbine generator, solar photovoltaic units) [1]. Kharitonov theorem-based fuzzy logic approach designed proportional–integral controller is considered and proposed for LFC of single-area power system. Simulation result of the proposed controller is compared with conventional PID controller response [2]. Elephant herding optimization algorithm (EHOA) optimized PID controller is considered and utilized for LFC of single-area non-reheat power system. The proposed controller response is compared with genetic algorithm (GA), bacterial foraging optimization algorithm (BFOA), teaching learning-based optimization algorithm-tuned PID controller and quasi-oppositional gray wolf optimization algorithm (QOGWOA) response [3]. A hybrid stochastic fractal search plus pattern search (hSFS-PS) technique-tuned cascade PI-PD controller is proposed and investigated for AGC of multi-source single-area power system (thermal unit, hydro unit and gas power unit with plug-in electrical vehicle (PEV)). The superiority of the proposed controller is shown by comparing with optimal control, DE and TLBO techniques-tuned PI, PID and cascade PI-PID response for the identical power system [4].

Imperialist competitive algorithm (ICA) based optimization technique designed Fractional-order fuzzy PID (FOFPID) controller is applied in multi source power System. The response of controller is examined for AGC of multi-source single-/ two-area power unit (hydrothermal, gas) interconnected power system with/ without considering RFB unit in power system. This response is compared with hybrid stochastic fractal search (hSFS), pattern search (PS), differential evolution (DE), teaching learning-based optimization (TLBO) technique-tuned I, PI and PID controller response [5]. Gray wolf optimizer algorithm-tuned PID controller was designed and demonstrated for three-area interconnected AGC in power system incorporated with solar thermal power plant (STPP). The simulation response is compared with GWO-I, PI and PID controller response [6].

Adaptive set-point modulation (ASPM) method-tuned PI controller is proposed and investigated for LFC of two-area power system with HVDC link. The proposed controller response is compared with conventional controllers (PI, PID) performance [7]. Modified harmony search algorithm (MHSA)-tuned PID controller is proposed and examined for LFC of interconnected two-area hydrothermal power generating system using ITAE objective function. Simulation response of the proposed controller is compared with GA–PID response [8]. BAT algorithm-based PID controller is proposed and demonstrated for LFC of interconnected multi-area power generating system. The proposed controller simulation result is compared with fuzzy gain scheduling technique-tuned PI and conventional controller response [9].

Cuckoo search (CS) algorithm-tuned PID controller is applied in LFC of three-area interconnected power system (two-area reheat thermal systems and hydro system) which is investigated. Effectiveness of the projected technique is compared with GA, PSO-tuned I controller [10]. Genetic algorithm fuzzy system (GAF)-based polar fuzzy logic controller is investigated for LFC of three-area connected (hydro, nuclear and thermal) power generating systems. The performance was compared with fuzzy and conventional PI controller [11]. BAT algorithm (BA)-based cascade PD–PID controller is applied for multi-area reheated thermal power generating system for AGC. The performance of P, PI and PID controllers is investigated and compared with PD–PID cascade controller [12].

Beta wavelet neural network (BWNN)-supported proportional-integral plus (PI+) controller is investigated for interconnected two-area thermal power systems with and without fast-acting energy storage device such as HAE–FC and RFB in load frequency control. The performance of power system is compared with BWNN-based PID controller simulation result [13]. Proportional–integral–derivative with filter (PIDF) is designed and proposed into AGC of a multi-area thermal power system in deregulated environment. Simulation result of designed controller was compared with fuzzy logic controller response [14]. Fruit fly optimization algorithm (FFOA)-tuned I, PI, IDD, PID and PIDD controllers are designed and investigated for AGC of multi-area multi-source (combination of reheat thermal hydro, nuclear) power generating units. The proposed controller response is compared with I, PI, PID and IDD controller response [15].

The above discussed literature review analysis clearly manifests that recently, optimization techniques play a vital role in generating good quality power to consumer. The optimization techniques are used to tune gain values of secondary controllers in power generating system. Many optimization techniques are utilized to solve engineering optimization problems of power systems such as generic algorithm (GA) [16], particle swarm optimization (PSO) [16], gravitational search algorithm (GSA) [17, 18], bacteria foraging optimization (BFO) [19], firefly algorithm (FA) [20, 21], bat algorithm (BA) [12, 22], cuckoo search algorithm (CSA) [10, 23], artificial bee colony [24], hybrid genetic firefly algorithm [25], modified harmony search algorithm [26], hybrid firefly algorithm and pattern search [27], stochastic particle swarm optimization [28], ant colony optimization [29–31] and different combinations of controller are variable structure control [32], discrete-mode control [33], classical controller [34], fuzzy IDD controller [35], FOPID controller [36], 2DOF-PID controller [37], robust PID controller [38], fuzzy logic controller [39, 40], adaptive controller [41], PI+ controller [42], optimal control [43], dual-mode gain scheduling of PI controller [44], fractional-order controller [45], 2DOF controllers [46], PD-PID cascade controller [47], integral controller [48], conventional controller [49], fuzzy logic controller [50], PID controller [28–31] and various optimization techniques are utilized to optimize a variety of engineering issues in real-time application [51–56]. The investigation of the literature analysis effectively shows that in recent years, evolutionary computational techniques are implemented to tune controller gain values of power system application for obtaining desired performance of system. In this work, the proposed system has investigated the effectiveness and performance of PSF-HS optimization technique in multi-source power generating system. Also, the performance of the proposed method is compared with PSO, GA and HS optimized controller performance in similar multi-source single-area power generating system.

The association of the remaining proposed chapter is as follows. Sect. 1 clearly discusses about the literature review of the proposed research in detail and Sect. 2 gives the clear transfer function Simulink model of investigated single-area multi-source power generating system (comprises thermal, hydro and gas system). Sects. 3 and 4 give details about the controller and proposed optimization technique tuning process. In Sect. 5, the performance of the proposed technique is compared with GA, PSO and HS tuning method-based controller response to show the efficiency of the proposed technique-tuned secondary controller response under emergency load situation. Finally, the performance of the proposed optimization technique in proposed system is clearly presented in conclusion section.

2 Single-Area Multi-source Power System Modeling

In this proposed work, AGC control of single-area multi-source power generating system is considered for investigation. Multi-source power system comprises thermal unit, hydro unit and gas generating unit. The Simulink transfer function with Simulink model of investigated power system model is shown in Fig. 1. The thermal power system includes single-stage reheat turbine. The simulation work carried out for the single-stage reheat turbine, hydro turbine and gas turbines equipped power system is considered for investigation. In this work, PID controller is applied as a secondary controller for frequency regulation of generated power by generating unit. The primary control loop available in power system does not generate required control signal at the time of emergency situation. To overcome this crisis, secondary PID controller is implemented to keep system stability and quality of generated power system. The nominal parameter values of investigated single-area multi-source power

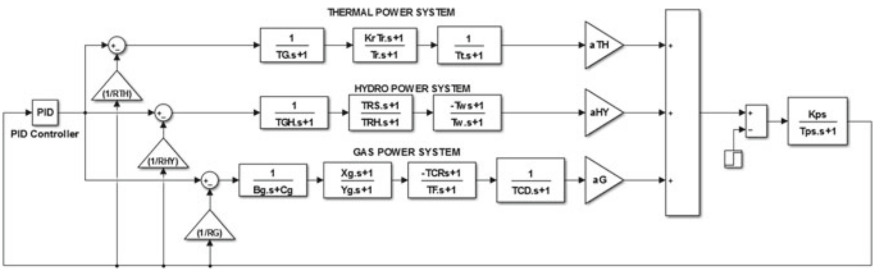


Fig. 1 Transfer function Simulink model of single-area multi source power system

Table 1 Nominal parameter values of analyzed power system

Symbol	Abbreviation	Nominal value
PID	Proportional–integral–derivative controller	–
R_{TH}	Speed regulation parameter for thermal power system	2.4 Hz/pu MW
R_{HY}	Speed regulation parameter for hydro power system	2.4 Hz/pu MW
R_{G}	Speed regulation parameter for gas power system	2.4 Hz/pu MW
T_{G}	Reheat time constant for thermal system	0.08 s
K_{r}	Reheater gain for thermal system	0.3
T_{r}	Time constant of reheater for thermal system	10 s
T_{t}	Time constant of turbine for thermal system	0.3 s
T_{GH}	Time constant of hydro turbine speed governor main servo	0.2 s
T_{R}	Steam turbine reheat time constant	10 s
T_{RH}	Hydro turbine speed governor transient droop time constant	28.75 s
B_{g}	Gas turbine constant of valve positioned	0.05 s
C_{g}	Gas turbine valve positioned	1
X_{g}	Gas governor lead time constant	0.6 s
Y_{g}	Gas governor lag time constant	1.0 s
T_{CR}	Combustion reaction time delay	0.01 s
T_{F}	Fuel time constant	0.23 s
T_{CD}	Compressor discharge volume time constant	0.2 s
T_{P}	Power system time constant	10 s
K_{P}	Power system constant	10
α_{TH}	Participation factor of thermal unit	0.543478
α_{HY}	Participation factor of hydro unit	0.326084
α_{G}	Participation factor of gas unit	0.130438

generating system are shown in Table 1, and system simulation model is given in Fig. 1.

3 Controller Design

In this work, PID controller is considered for AGC scheme in single-area multi-source power system as secondary control loop. The PID controller structure is depicted in Fig. 2, and it consists of three controller actions with respect gain values. The gain values are proportional, integral and derivative controller, and its gain values are K_P, K_I and K_D, respectively. The input of values of PID controller is considered as area control error, and it is a linear combination of deviations in system frequency and deviation in tie-line power in between connected power system.

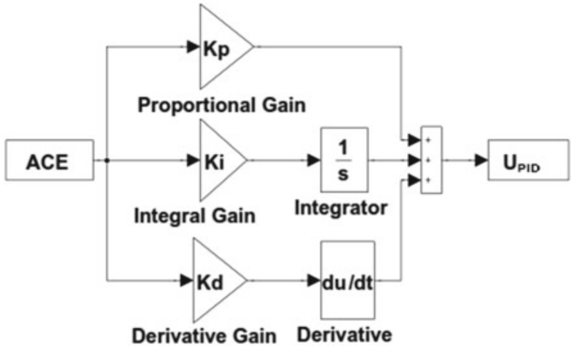


Fig. 2 Structure of PID controller

Objective Function

At the time of controller gain values, optimization selection of appropriate objective function plays vital role. The objective is selected based on the requirement in system output response. In this proposed research work, ITAE cost function is implemented at the time of optimization controller gain value optimization process. The expression for ITAE objective function is given as follows (1):

$$J = \text{ITAE} = \int_{0}^{t_{\text{sim}}} t.|e(t)|\text{d}t \tag{1}$$

For PID controller parameters (gain values) ranges are chosen in between 0 to 1, and it is given (5–7). From the earlier structure controller, it is obviously shown that area controller error (ACE) is considered as an input signal, and control signal is considered as the output of controller (U_{PID}) and given in (4). The generated control signal by the controller depends on the ACE values. The ACE is defined as "the linear arrangement of system frequency error and error in tie-line power." The expression for area control error is depicted in Eq. (3), and PID controller generated control signal is given in Eq. (2).

The PID controller transfer function is given as the following expression:

$$G_{\text{PID}}(s) = K_{\text{P}} + K_{\text{I}}S + \frac{K_{\text{D}}}{S} \quad \text{or} \quad G_{\text{PID}}(s) = K_{\text{P}}\left(1 + \frac{1}{T_i S} + T_{\text{D}}S\right) \tag{2}$$

where
Proportional gain is K_{p},
Integral gain is K_{i},
Derivative gain is K_{d},
Integral action time is T_{i} and
Derivative action time T_{d}.
The expression of ACE is given by:

Table 2 Gain values of controller tuned with different optimization techniques

Optimization	Proportional gain	Integral gain	Derivative gain	Performance index
GA	0.964608	0.996576	0.6472559	0.1562122
PSO	0.9880474	0.9875885	0.5933165	0.1553838
HS	0.9937	0.9994	0.5323	0.1532
PSF-HS	0.9999	0.9999	0.553	0.152858

$$\text{ACE}_i = B_i.\Delta F_i + \Delta P_{\text{tie }i,j} \tag{3}$$

The generated control signal by each controller in each area as:

$$U_{\text{PID}} = u_{\text{i}}(t) = K_{\text{P}}.\text{ACE} + K_{\text{I}} \int_0^t \text{ACE}\,\text{d}t + K_{\text{D}} \frac{\text{d ACE}}{\text{d}t} \tag{4}$$

$$K_{\text{P}}^{\min} \leq K_{\text{P}} \leq K_{\text{P}}^{\max} \tag{5}$$

$$K_{\text{I}}^{\min} \leq K_{\text{I}} \leq K_{\text{I}}^{\max} \tag{6}$$

$$K_{\text{D}}^{\min} \leq K_{\text{D}} \leq K_{\text{D}}^{\max} \tag{7}$$

where K_{P}, K_{I} and K_{D} are the minimum gain and maximum gain values of the proposed controller gain. The minimum and maximum controller gain values are chosen to be 0 and 1, respectively. The detail about the proposed optimization technique is clearly depicted into the following section, and the corresponding gain values are given in Table 2.

4 Proposed Parameter-Setting-Free Harmony Search Algorithm-Tuned PID Controller

In 2010, Geem and Sim [57] proposed parameter-setting-free harmony search algorithm which is the modified version of classical harmony search [58]. Like the other meta-heuristic algorithms, the performance of the classical harmony search is strongly dependent on the values of its parameters which are harmony memory consideration rate (HMCR) and pitch adjusting rate (PAR). Finding the optimal parameter values is a laborious task which requires expertise and knowledge about the algorithm, its parameters and the problem.

In this context, parameter-setting-free harmony search algorithm is good choice in the optimization of PID controller. Optimum value of HMCR and PAR has been selected through three operations which are random tuning, rehearsal and performance as described in the [57]. In this work, parameter-setting-free harmony search has been applied to optimize Eq. (1) for the single-area power generating multi-source model given in Figure 1. Three parameter values (p, i, d) have to be optimized according to Eq. (1). The history of the HMCR and PAR is given in Fig. 3a, b, respectively. After 300 iterations, optimum controller gain values of GA, PSO, HS and PSF-HS have been reported in Table 2.

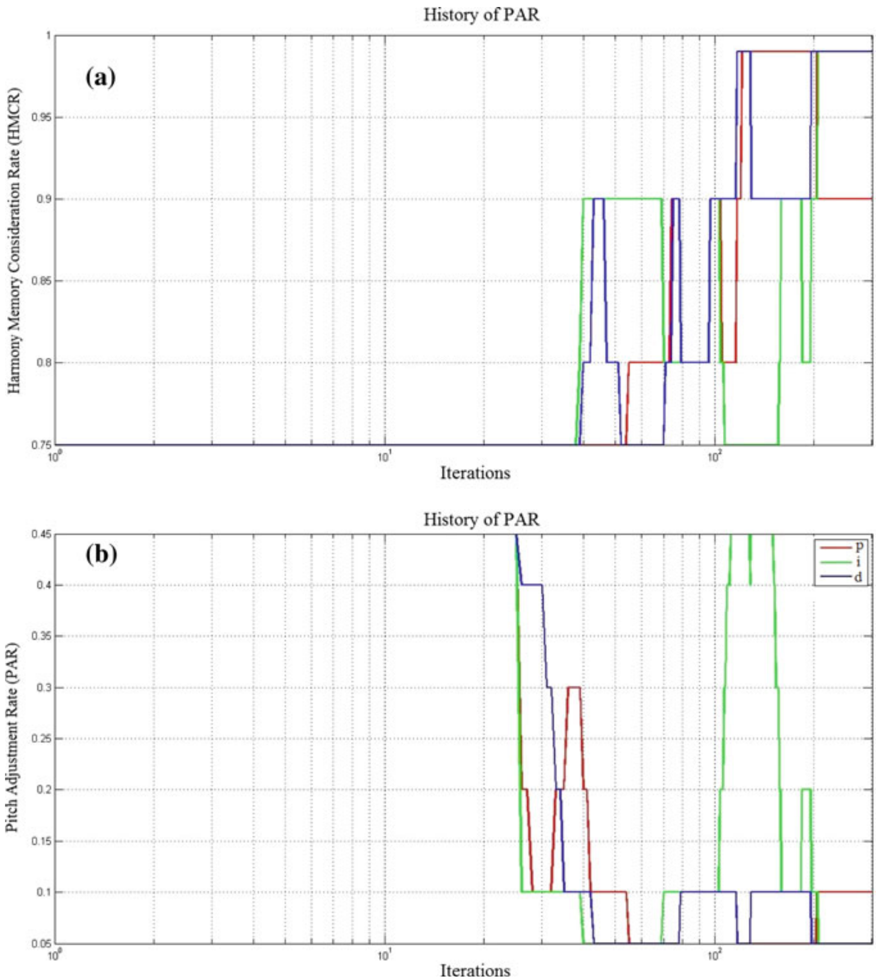


Fig. 3 **a** History of HMCR. **b** History of PAR

5 Simulation Results and Discussions

The proposed PSF-HS algorithm is implemented to tune gain values of PID controller in single-area multi-source power generating system. The model is developed under MATLAB\Simulink environment. The designing details of the proposed system design are given in Sect. 4. The optimal gain values are tuned by implementing optimization technique, and optimized gain values of controller gain values are given in Table 2 for investigated power system.

A 1% SLP is applied into analyzed power generating system for verification of the proposed optimization technique-based controller performance in single-area multi-source power system. The response comparison of the proposed algorithm performance is compared with the open-loop response under sudden load demand situation. The frequency deviation responses of GA-PID controller, PSO-PID controller, HS-PID controller and PSF-HS-PID controller performance verified with response of open-loop system under 1% SLP environment are depicted in Figs. 4, 5, 6 and 7, respectively. The time-domain specification parameters, peak overshoot, settling time and undershoot values, are given in Table 3 for various optimization technique-based controller performances.

From the response comparisons in Figs. 4, 5, 6 and 7 in that dashed line shows the result of PSO technique-tuned controller response, and solid line shows the result of open-loop system performance under critical load disturbance situation.

Based on the response comparisons shown in Figs. 4, 5, 6 and 7, it obviously shows that the performance of system is improved compared to open-loop response of system by utilizing optimization technique for optimization of controller gain values. And also, time-domain specification numerical parameters values are shown in Table 3. In addition, the response of GA, PSO, HS and PSF-HS techniques is compared, and time-domain specification numerical parameters values are given in Table 3.

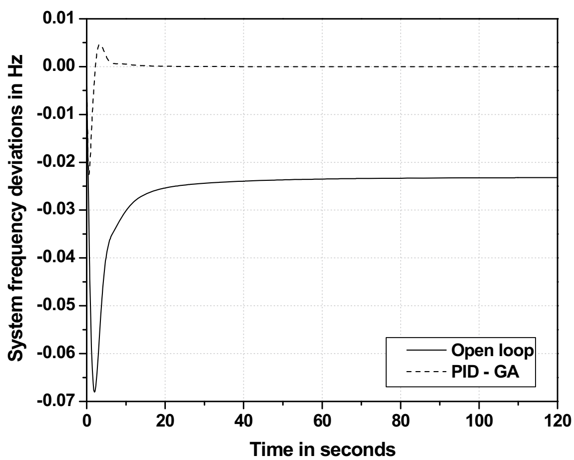


Fig. 4 System frequency changes comparison with open-loop and GA-based PID controller

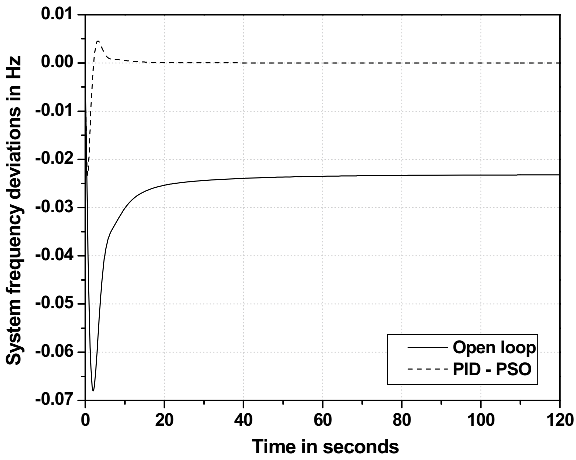


Fig. 5 System frequency changes comparison with open loop and PSO based PID controller

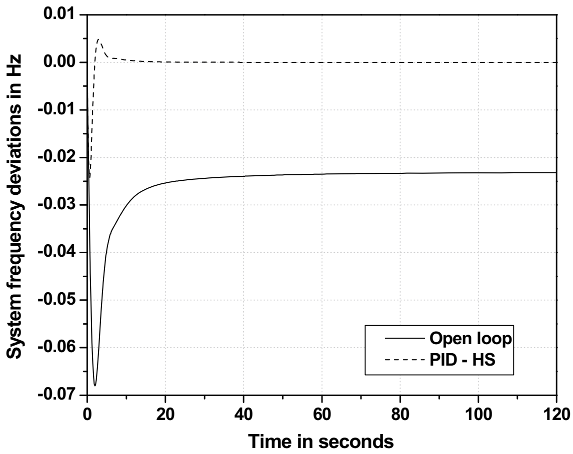


Fig. 6 System frequency changes comparison with open-loop and HS-based PID controller

In that comparison of simulation response in Fig. 8, dotted line, dash line, dash dot line and solid lines show the performance of GA, PSO, HS and PSFHS technique-optimized controller response, respectively. From the simulation response comparison result and Table 3 indicates numerical values and it is effectively evident that proposed technique-tuned controller yield minimal settling time over GA, PSO and HS optimization technique-tuned controller response.

The bar chart comparisons in Figs. 9, 10 and 11 shows the values of peak overshoot and undershoot values, settling time for different optimization technique-tuned controllers performance under emergency load demand situation scenario.

The bar chart comparisons (Figs. 9, 10 and 11) of peak overshoot, settling time and undershoot values clearly show that the proposed PSF-HS technique-tuned controller yields better performance over GA, PSO and HS technique optimized controller

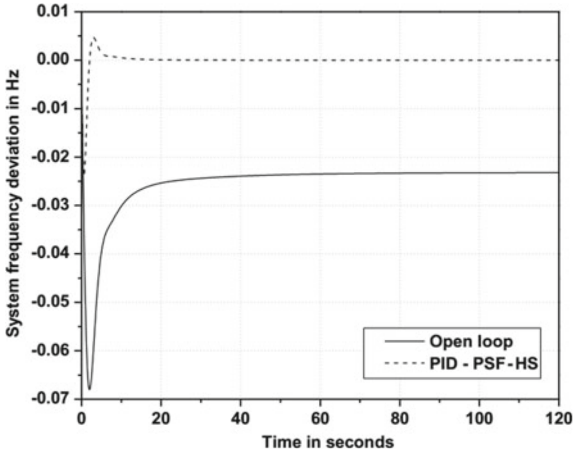


Fig. 7 System frequency changes comparison with open-loop and PSF-HS-based PID controller

Table 3 Settling time (T_s), overshoot (OS), undershoot (US) values for different technique optimized controllers

Parameter/controller	Settling time (s)	Peak overshoot (Hz)	Peak undershoot (Hz)
PID–GA	14.5	0.0049	0.0230
PID–PSO	16	0.0048	0.0235
PID–HS	15.5	0.005	0.0240
PID-PSF-HS	13.5	0.0047	0.022

performance under emergency loading conditions in power system with lesser peak magnitude of system frequency with settling time.

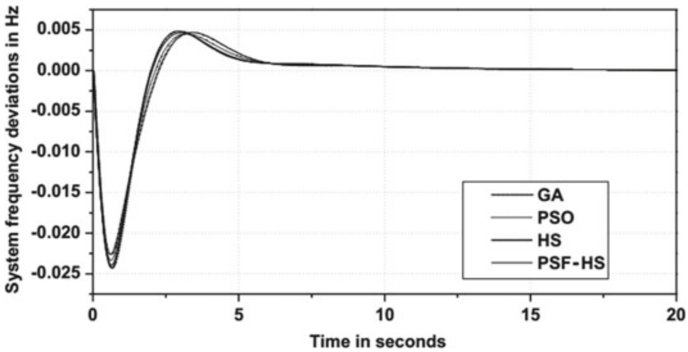


Fig. 8 System frequency changes comparison with GA, PSO, HS and PSF-HS-PID controllers

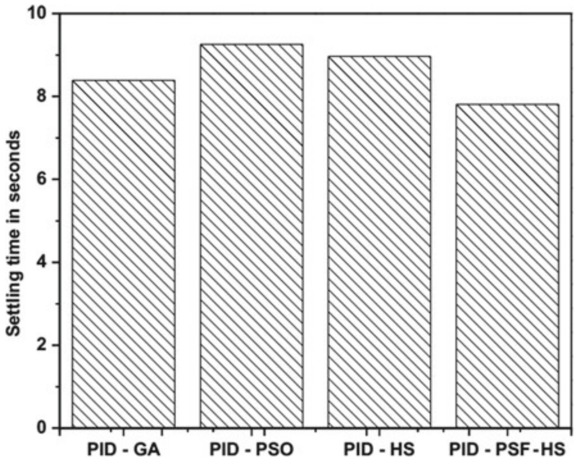


Fig. 9 Bar chart comparisons of settling different technique-tuned PID controllers

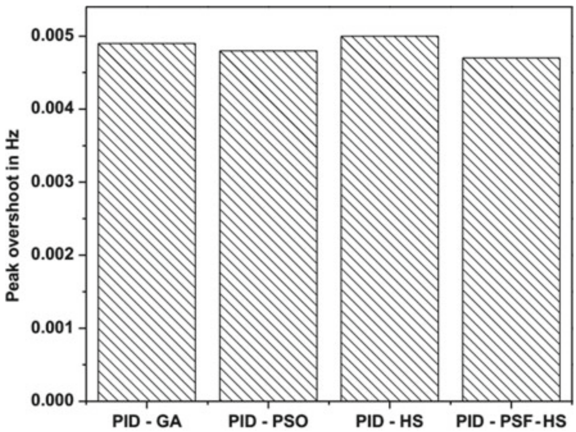


Fig. 10 Comparisons in bar chart of overshoot with different technique-tuned PID controllers

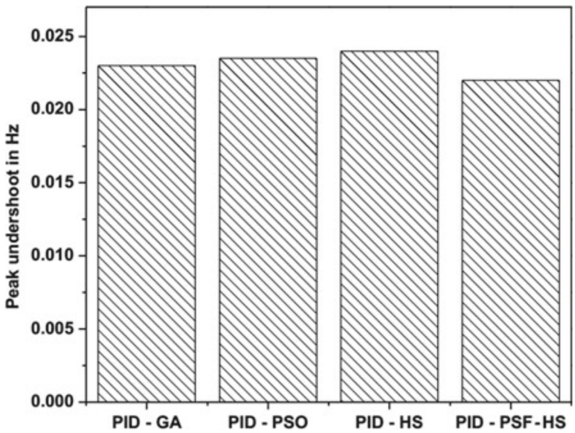


Fig. 11 Comparisons in bar chart of undershoot with different technique-tuned PID controllers

6 Conclusion

In this work, automatic generation control of single-area multi-source power generating system is analyzed by the inclusion of secondary PID controller. The single-area multi-source power system comprises thermal, hydro and gas power plant. The PID controller considered as a secondary controller and implemented in single-area power system to keep the power system parameters within the prescribed limit at the time of emergency situation is effectively investigated. In this work, parameter-setting-free harmony search technique is implemented to tune the PID controller gain values, and the performance of the proposed tuning technique is compared with GA technique, PSO technique and HS technique-tuned controller response. The simulation result comparison clearly shows that PSF-HS technique-based PID controller response yields better controlled response over GA, PSO and HS technique-tuned PID controller response in terms of minimal peak over- and undershoot with settling time in the response.

References

1. Rajesh, K. S., & Dash, S. S. (2019). Load frequency control of autonomous power system using adaptive fuzzy based PID controller optimized on improved sine cosine algorithm. *Journal of Ambient Intelligence and Humanized Computing, 10*(6), 2361–2373.
2. Golshannavaz, S., Khezri, R., Esmaeeli, M., & Siano, P. (2018). A two-stage robust-intelligent controller design for efficient LFC based on Kharitonov theorem and fuzzy logic. *Journal of Ambient Intelligence and Humanized Computing, 9*(5), 1445–1454.
3. Sambariya, D. K., & Fagna, R. (2017, July). A novel elephant herding optimization based PID controller design for Load frequency control in power system. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)* (pp. 595–600). IEEE.
4. Padhy, S., & Panda, S. (2017). A hybrid stochastic fractal search and pattern search technique based cascade PI-PD controller for automatic generation control of multi-source power systems in presence of plug in electric vehicles. *CAAI Transactions on Intelligence Technology, 2*(1), 12–25.
5. Arya, Y. (2017). AGC performance enrichment of multi-source hydrothermal gas power systems using new optimized FOFPID controller and redox flow batteries. *Energy, 127,* 704–715.
6. Sharma, Y., & Saikia, L. C. (2015). Automatic generation control of a multi-area ST–Thermal power system using grey wolf optimizer algorithm based classical controllers. *International Journal of Electrical Power & Energy Systems, 73,* 853–862.
7. Ketabi, A., & Fini, M. H. (2015). An adaptive set-point modulation technique to enhance the performance of load frequency controllers in a multi-area power system. *Journal of Electrical Systems and Information Technology, 2*(3), 391–405.
8. Shivaie, M., Kazemi, M. G., & Ameli, M. T. (2015). A modified harmony search algorithm for solving load-frequency control of non-linear interconnected hydrothermal power systems. *Sustainable Energy Technologies and Assessments, 10,* 53–62.
9. Sathya, M. R., & Ansari, M. M. T. (2015). Load frequency control using bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system. *International Journal of Electrical Power and Energy Systems, 64,* 365–374.

10. Abdelaziz, A. Y., & Ali, E. S. (2015). Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system. *International Journal of Electrical Power and Energy Systems, 73,* 632–643.
11. Chaturvedi, D. K., Umrao, R., & Malik, O. P. (2015). Adaptive polar fuzzy logic based load frequency controller. *International Journal of Electrical Power and Energy Systems, 66,* 154–159.
12. Dash, P., Saikia, L. C., & Sinha, N. (2015). Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller. *International Journal of Electrical Power and Energy Systems, 68,* 364–372.
13. Francis, R., & Chidambaram, I. A. (2015). Optimized PI+ load–frequency controller using BWNN approach for an interconnected reheat power system with RFB and hydrogen electrolyser units. *International Journal of Electrical Power and Energy Systems, 67,* 381–392.
14. Gorripotu, T. S., Sahu, R. K., & Panda, S. (2015). AGC of a multi-area power system under deregulated environment using redox flow batteries and interline power flow controller. *Engineering Science and Technology, an International Journal, J18*(4), 555–557.
15. Mohanty, B., & Hota, P. K. (2015). Comparative performance analysis of fruit fly optimisation algorithm for multi-area multi-source automatic generation control under deregulated environment. *IET Generation, Transmission and Distribution, 9*(14), 1845–1855.
16. Khadanga, R. K., & Satapathy, J. K. (2015). Time delay approach for PSS and SSSC based coordinated controller design using hybrid PSO–GSA algorithm. *International Journal of Electrical Power and Energy Systems, 7,* 262–273.
17. Sahu, R. K., Panda, S., & Padhan, S. (2015). A novel hybrid gravitational search and pattern search algorithm for load frequency control of nonlinear power system. *Applied Soft Computing, 29*, 310–327.
18. Khadanga, R. K., & Satapathy, J. K. (2015). A new hybrid GA–GSA algorithm for tuning damping controller parameters for a unified power flow controller. *International Journal of Electrical Power & Energy Systems, 73*, 1060–1069.
19. Ali, E. S., & Abd-Elazim, S. M. (2013). BFOA based design of PID controller for two area load frequency control with nonlinearities. *International Journal of Electrical Power and Energy Systems*, *51*, 224–231
20. Padhan, S., Sahu, R. K., & Panda, S. (2014). Application of firefly algorithm for load frequency control of multi-area interconnected power system. *Electric Power Components and Systems, 42*(13), 1419–1430.
21. Abd-Elazim, S. M., & Ali, E. S. (2016). Load frequency controller design of a two-area system composing of PV grid and thermal generator via firefly algorithm. *Neural Computing & Applications* 1–10.
22. Abd-Elazim, S. M., & Ali, E. S. (2016). Load frequency controller design via BAT algorithm for nonlinear interconnected power system. *International Journal of Electrical Power & Energy Systems, 77*, 166–177.
23. Chaine, S., & Tripathy, M. (2015). Design of an optimal SMES for automatic generation control of two-area thermal power system using cuckoo search algorithm. *Journal of Electrical Systems and Information Technology, 2*(1), 1–13 (In Press)
24. Gozde, H., Taplamacioglu, M.C., & Kocaarslan, I. (2012). Comparative performance analysis of artificial bee colony algorithm in automatic generation control for interconnected reheat thermal power system. *Electric Power and Energy Systems, 42*, 167–178
25. Farook, S., & Sangameswara Raju, P. (2012). Feasible AGC controllers to optimize LFC regulation in deregulated power system using evolutionary hybrid genetic firefly algorithm. *Journal of Electrical Systems, 8*(4), 459–471.
26. Shivaie, M., Kazemi, M.G., & Ameli, M. T. (2015). A modified harmony search algorithm for solving load-frequency control of non-linear interconnected hydrothermal power systems. *Sustainable Energy Technologies and Assessments, 10*, 53–62.
27. Sahu, R. K., Panda, S., & Padhan, S. (2015). A hybrid firefy algorithm and pattern search technique for automatic generation control of multi area power systems. *Electric Power and Energy Systems, 64*, 9–23.

28. Jagatheesan, K., Anand, B., & Ebrahim, M. A. (2014). Stochastic particle swarm optimization for tuning of PID controller in load frequency control of single area reheat thermal power system. *International Journal of Electrical and Power Engineering*, *8*(2), 33–40 ISSN: 1990-7958.
29. Omar, M., Soliman, M., Abdel Ghany, A. M., & Bendary, F. (2013). Optimal tuning of PID controllers for hydrothermal load frequency control using Ant colony optimization. *International Journal of Electrical Engineering and Informatics, 5*, 348–354.
30. Jagatheesan, K., Anand, B., & Dey, N. (2015). Automatic generation control of thermal-thermal-hydro power systems with PID controller using ant colony optimization. *International Journal of Service Science, Management, Engineering, and Technology, 6*(2), 18–34.
31. Jagatheesan, K., Anand, B., Dey, N., & Ashour, A. S. (2015). Artificial Intelligence in performance analysis of load frequency control in thermal-wind-hydro power systems. *International Journal of Advanced Computer Science and Applications, 6*(7), 203–212.
32. Das, S., Kothari, M. L., Kothari, D. P., & Nanda, J. (1991) Variable structure control strategy to automatic generation control of interconnected reheat thermal system. *IEE Proceedings-D*, **138**(6), 579–585
33. Kothari, M. L., Nandha, J., Kothari, D. P., & Das, D. (1989, May). Discrete-mode automatic generation control of a two-area reheat thermal system with new area control error. *IEEE transactions on Power Systems*, *4*(3), 730–738.
34. Nanda, J., & Saikia, L. C. (2008). Comparisons of performance of several types of classical controller in automatic generation control for an inter connected multi-area thermal system. In *2008 Australasian Universities Power Engineering Conference (AUPEC'08)*, pp. 1–6.
35. Saikia, L. C., Sinha, N., & Nanda, J. (2013). Maiden application of bacterial foraging based fuzzy IDD controller in AGC of a multi-area hydrothermal system. *Electric Power and Energy Systems, 45*, 98–106.
36. Taher, S. A., Fini, M. H., & Aliabadi, S. F. (2014). Fractional order PID controller design for LFC in electric power systems using imperialist competitive algorithm. *Ain Shams Engineering Journal, 5*, 121–135.
37. Sahu, R. K., Panda, S., & Rout, U. K. (2013). DE optimized parallel 2-DOF PID controller for load frequency control of power system with governor dead-band nonlinearity. *Electric Power and Energy Systems, 49,* 19–33.
38. Shabani, H., Vahidi, B., & Ebrahimpour, M. (2013). A robust PID controller based on Imperialist competitive algorithm for load frequency control of power systems. *ISA Transactions, 52,* 88–95.
39. Subha, S. (2014). Load frequency control with fuzzy logic controller considering governor dead band and generation rate constraint non-linearities. *World Applied Science Journal, 29*(8), 1059–1066
40. Nanda, J., & Sakkaram, J. S. (2003). Automatic generation control with fuzzy logic controller considering generation rate constraint. In *Proceedings of the 6th International Conference on Advance in Power System Control, Operation and Management, APSCOM 2003*, (pp. 770–775). Hong Kong.
41. Swain, A. K., & Mohanty, A. K. (1995). Adaptive load frequency control of an interconnected hydrothermal system considering generation rate constraints, *76*, 109–114.
42. Francis, R., & Chidambaram, I. A. (2015). Optimized PI+ load-frequency controller using BWNN approach for an interconnected reheat power system with RFB and hydrogen electrolyser units. *Electric Power and Energy Systems, 67*, 381–392
43. Chaine, S., Tripaathy, M., & Satpathy, S. (2015). NSGA-II based optimal control scheme of wind thermal power system for improvement of frequency regulation characteristics. *Ain Shams Engineering Journal* 6(3), 851–863.
44. Sathya, M.R., & Mohamed Thameem Ansari, M. (2015). Load frequency control using bat inspired algorithm based dual mode gain scheduling of PI controller for interconnected power system. *Electric power and energy systems, 64*, 365–374.
45. Pan, I., & Das, S. (2015). Fractional-order load frequency control of interconnected power systems using chaotic multi-objective optimization. *Applied Soft Computing, 29*, 328–344

46. Dash, P., Saikia, L. C., & Sinha, N. (2015). Comparison of performance of several FACTS devices using Cuckoo search algorithm optimized 2DOF controllers in multi-area AGC. *Electric Power and Energy Systems, 65,* 316–324.
47. Dash, P., Saikia, L. C., & Sinha, N. (2015). Automatic generation control of multi area thermal system using bat algorithm optimized PD-PID cascade controller. *Electric Power and Energy Systems, 68,* 364–372.
48. Jagatheesan, K., & Anand, B. (2012). Dynamic performance of multi-area hydro thermal power systems with integral controller considering various performance indices methods. In *IEEE International Conference on Emerging Trends in Science, Engineering and Technology*, Tiruchirappalli, December 13–14, 2012.
49. Jagatheesan, K., & Anand, B. (2014). Automatic generation control of three area hydro-thermal power systems considering electric and mechanical governor with conventional and ant colony optimization technique. *Advances in Natural and Applied Science*, *8*(20), 25–33. ISSN: 1998-1090
50. Anand, B., & Ebenezer Jeyakumar, A. (2009). Load frequency control with fuzzy logic controller considering non-linearities and boiler dynamics. *ACSE, 8*, 15–20.
51. Dey, N. (Ed.) (2017). *Advancements in applied metaheuristic computing*. IGI Global.
52. Dey, N., Ashour, A., & Bhattacharyya, S. (2020). *Applied nature-inspired computing: algorithms and case studies*. Springer.
53. Das, S. K., Kumar, A., Das, B., & Burnwal, A. P. (2013). On soft computing techniques in various areas. *Computer Science and Information Technology, 3,* 59.
54. Bera, S., Das, S. K., & Karati, A. (2020). Intelligent routing in wireless sensor network based on african buffalo optimization. In *Nature Inspired Computing for Wireless Sensor Networks* (pp. 119–142). Springer, Singapore.
55. De, D., Mukherjee, A., Das, S. K., & Dey, N., *Nature inspired computing for wireless sensor networks*.
56. Kacprzyk, J. (2019). *Lecture notes in networks and systems*.
57. Geem, Z. W., & Sim, K.-B. (2010). Parameter-setting-free harmony search algorithm. *Applied Mathematics and Computation, 217*(8), 3881–3889.
58. Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, *76*(2), 60–68.

Environmental Sound Classification Using Neural Network and Deep Learning

Dharma Rane, Pushkar Shirodkar, Trilochan Panigrahi, and S. Mini

1 Introduction

The rapid development and urbanization in developed countries like India has resulted in an ever increasing number of vehicles on existing roads. This has lead to frequent traffic congestion and the desideratum for efficient traffic management systems. Road noise is moreover considered by the World Health Organization (WHO) as the second most problematic nuisance for people's health and well-being, after air pollution [1, 2]. The growth of urban population in India as an example is so intensive that the environment of many cities fails to satisfy many biological and social human needs. Traffic discipline is not followed and horns are used unnecessarily in the cities. Therefore, it has been admitted that the noise from road traffic constitutes the one of the danger to human health especially the patient in the intensive care unit (ICU), pregnant woman and new born babies, as road transport is the source of noise located directly near the places of human residence and hospitals [3].

In the similar fashion rail and aviation traffic also grown up increase because of our rapid economical growth. Further, with an increasing pace in the development of technologies in developing countries, the unwanted noise from man made machinery present in the environment which is also increasing. For example, in the European Union, about 56 million people (54%) living in areas with more than 250,000 inhab-

D. Rane · P. Shirodkar · T. Panigrahi (✉) · S. Mini
National Institute of Technology Goa, Goa, India
e-mail: tpanigrahi80@gmail.com

D. Rane
e-mail: dharmarane19971@gmail.com

P. Shirodkar
e-mail: shirodkarpushkar.9@gmail.com

S. Mini
e-mail: mini@nitgoa.ac.in

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_3

itants are exposed to road traffic noise of more than average LDEN 55 dB per year. According to Global Burden of Disease 2010-11 nearly 1.3 billion people are affected by hearing loss. Investigators rated hearing loss as the 13th most important contributor (19.9 million years, 2.6% of total number) to the global years lived with disability (YLD) [1]. The study says that, log term exposer to the noise from the road traffic, rail and aviation noise causes many diseases like hyper-station, breast cancer and diabetics [4–7].

In order to reduce the annoyance, it is an important task to reduce the noise level after doing temporal and spatial assessment in real time. A major challenge to appropriately categorized noise is random nature. Recently many attempts have been made to classify sounds such as sounds from cars, ambulances, trains, metros, factories [8]. Different kind of noise are classified after real time assessment using different soft computing techniques [9, 10]. Various data mining approaches may be applied here [11, 12].

Neural networks have been gaining increasing popularity in signal processing and other classification problems. Many researches have used neural network to classify the sounds [13]. Other approaches suggest the utilization of Fourier transform and presence of signal in certain frequency range to decide whether sound from a particular source contributes to it [14]. Extraction of frequency domain features followed by utilization of artificial neural network for classification has also been attempted [15–19]. In fact, deep learning (DL) is playing an increasingly important role in our day to lives. It made a huge impact in areas, such as cancer diagnosis, precision medicine, self-driving cars, predictive forecasting, and speech recognition [20, 21].

In proposed work, the mobile phone can record the sound signal of the environment using it microphone for predefined duration. To make the algorithm expeditious and efficient, the input signal is windowed into blocks. After dividing the signals into windows, the proposed method extracts Mel-frequency cepstral coefficients (MFCCs), spectral centroid, spectral flux, spectral roll-off and zero crossing rate as features from input audio signal [22, 23]. These features are habituated to train a Bayesian regularized artificial neural network (BRANN) [24]. This reduces possible over fitting of the neural network and thus the proposed method can be utilized in the mobile app.

Further, a new model that uses two set of feature vectors at a time for classification is proposed and implemented to improve the performance [25]. The training and testing of the BRANN is done by using standard data set available in the repository. We also tested the real time ambulance noise in Indian traffic. The contribution of ambulance sound in the presence of other noise like horn and engine sound is also estimated.

Once the network is trained it can be preserved as a model file. This can be habituated to relegate without the desideratum for MATLAB environment and hence can be acclimated to implement the same function in an app on a cellphone. Currently preserved models from MATLAB can be utilized in Apple devices. The same network can be implemented in python using the tensor flow library and can be used for developing Android applications.

The aims of the proposed work is to classify different sounds present in the environment in the domain of Indian scenario and with the development of an app on android platform. This app can be used for real time detection and classification. Also the information of sources can be used to tackle them as well as plan buildings such as hospitals that have constrains for level and type of noise produced in the environment.

1.1 Scope

The scope for the proposed work is as follows

- Understanding different kind of acoustic noise present in the urban cities.
- Designing a neural network model to classify the environment sounds.
- To estimate the contribution of a particular acoustic noise in a particular time period.
- Improving the prediction accuracy of the model with the deep learning algorithms and techniques.
- The performance improvement by providing two samples at a time to the neural network.

2 Related Work

In order to minimize the acoustic noise level near residential area in a city, it is important to know where and what kind of the noise source are present. After assessing the noise level, the contribution each class of noise is important to know for further planning to minimize them. Institute of Sound and Vibration Research, University of Southampton, United Kingdom and Department of Applied Physics, University of Granada, Spain working on the classification of noise using different machine learning algorithms. Different kind of noises are classified after real time assessment [9] using most recent developed classifier. There are many bio-inspired techniques to handle big data are provided in [26].

Soft computing approach for vehicular traffic noise prediction is presented in [10]. The methods are generalized linear model, decision trees, random forests and neural networks used to predict the sound pressure level at different locations in the Patiala city in India. The model predict the noise level hourly by taking the input variables as traffic volume per hour, percentage of heavy vehicles and average speed of vehicles. Similar kind of analysis also done in the Tehran's city in [27]. But these methods are not classifying the noise, only predict the noise intensity at a particular time.

The use of machine-learning methods to effectively detect kind of environmental noise sources is given in [28, 29]. Both unsupervised such as clustering methods, principle components analysis and supervised methods such as support vector machines, and neural networks are used to classify the signals based on the obtained features. The generalization performance of all the methods is assessed and the strengths and weaknesses of each approach are discussed.

A machine learning approach for classifying audio signals to distinguish whale sounds from other sources of sound in the underwater sounds cape, such as ships has been analyzed in [30]. This is a challenging problem like environmental noise classification due to wide variation in ambient background noise and whale vocalization patterns within and across species. The authors adapted deep convolutional neural networks (CNN) to analyze spectral patterns of different noise sources. Further, the demonstration on robust performance of open-source databases including whale vocalizations.

The problem of modeling and predicting environmental noise in urban cities is a complex and non-linear problem. A prediction model based on a back-propagation neural network to solve this problem is proposed and examined in [31, 32]. The model is intended to precisely predict a five minute integration period level and temporal-spectral composition of the sound pressure of urban environments.

In recent past, research has been done on the use of machine learning for speech processing applications. Research is now focused on use of deep learning for speech-related applications and shows better results when compared to others. A survey on studies that have been conducted for speech applications is provided in [29, 33]. It has been found that the deep learning algorithm is one of the popular machine learning tool for speech recognition problem. Even though machine learning algorithms are used for impulsive noise classification [34]. But the present problem is different from these applications.

Like air, quality of water also degrade due to domestic and industrial pollution. Many researcher have achieved reasonable success in predicting the water quality using machine learning algorithms. In one of the approach, cuckoo search (CS) algorithm is applied to improve the support in the classification process during the water quality prediction [35]. Neural network also used to predict the structural failure of multistoried buildings [36]. Genetic algorithm based neural network is also used to predict the failure in [37]. A geometrical and linear programming approach are used for routing in ad-hoc network [38, 39]. Similar kind of optimization algorithms may be used to improve the performance of the neural network model. This kind of methods may not be feasible to implement in android platform as the platform has memory and computational constraints.

Neural networks are being applied to the problems in the area of noise identification and recognition. Then, environmental noise source classification using artificial neural networks (ANN) has been reported in literature since 2010 [15, 37] where three non-stationary noise sources are chosen to recognize. These are highway, subway and airport. Only time-domain based feature parameters are used and achieved 54% accuracy. But two-phase ANN classifier achieved 83–89% accuracy rates. Environmental noise classification with convolutional neural networks has been provided in [13]. The model is relying on mel-frequency cepstral coefficients and achieves results comparable to other state-of-the-art approaches, but the complexity is more.

In the field of ANN, the back propagation neural network is a very popular technique which relies on supervised learning. One drawback of standard back propagation neural networks is the potential for over fitting the training data set, which results in reduced accuracy on unknown data set. To overcome that problem, Bayesian regularized artificial neural networks (BRANNs) is developed [24]. It can reduce the need for lengthy cross-validation and successfully applied in many applications [40]. As per the literature, deep learning algorithms are providing better performance, thus in our study, we followed large network with deep learning algorithm based neural network for classification.

3 Problem Formulation

Let the sound signal is represented as $x(t)$ defined in the closed interval $[0, T]$ where t is the time variable and T is the total duration of the recorded audio signal. As we know that the continuous time signal can not be stored as it is. The signal $x(t)$ is sampled with frequency F_s which is typically around 44.1 kHz for audio signals. The Sampling period $T_s = \frac{1}{f_s}$ and the discrete time signal after sampling is [25]

$$x_s[n] = x(nT_s) \tag{1}$$

where $x_s[n]$ is sampled signal and n is the sampling index. Now, the discrete time signal $x_s[n]$ is defined over the interval $[0, N]$, where N is defined as

$$N = \left\lfloor \frac{T}{T_s} \right\rfloor \tag{2}$$

Now the discrete-time signal $x_s[n]$ is used to process in order to know the kind of sound present in it and their contributions. The step by step process is given in Fig. 1.

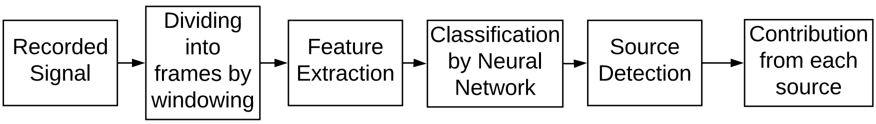


Fig. 1 Block diagram of 1 input-output model for urban noise classification using neural network

3.1 Segmenting an Audio Signal Into Windows

The features of the signal can be extracted in both time and frequency domain from the discrete time signal $x_s[n]$ over the length $N + 1$. Usually the N value is very large. There will be huge computational burden for extracting the feature. Large memory is required to store signal. In order to have a clear understanding about signal $x_s[n]$ we divide the signal $x_s[n]$ into small segments called Windows.

3.1.1 Importance of Windowing

One can process whole recorded signal at a time. But it is always good if we process the signal in block wise. Therefore we followed windowing method to divide the recorded audio signal into windows. The importance of windowing techniques are as follows:

- It reduces the waiting time to record whole signal.
- Memory requirement to store the signal is reduced which is always an advantage in IoT device.
- Segmenting the signal into windows allows for real time classification of signal.
- It reduces the length of signal for feature extraction thereby reducing computational effort.
- It allows for parallel computing.

A discrete time signal $x_s[n]$ of length $N + 1$ can be expressed as a combination of non overlapping individual windows $w_i[n]$, $i \in [1, N_p]$. The number of windows N_p is defined as

$$N_p = \left\lceil \frac{N+1}{L} \right\rceil$$

where L is the length of each window. If the number of input data from the signal is less in last window, insert zeros in the right and make the window length L.

Windowing method is explained here. Discrete time signal $x_s[n]$ which is divided into many segments that are of fixed length L such that $L < N$ and the signal $x_s[n]$ is divided into fixed number of parts N_p. The ith window $w_i[n]$ is given by Eq. (3).

$$w_i[n] = \Big\{ x_s[n], \qquad iL \leq n < (i+1)L \tag{3}$$

In Eq. (3), i can take integer values from $1, 2, 3, \ldots N_p$.

Using windows we can represent the sampled signal as

$$x_s[n] = \{w_1[n], w_2[n], \ldots, w_{N_p}[n]\} \tag{4}$$

After dividing the data into small windows, the feature has been extracted in both temporal and frequency domain. Then the feature vector are used to train neural network. Deep learning algorithms are used to train the neural network. Initially, two classes are classified that is noise and presence of ambulance siren. Then the real situation where multiple classes of sounds are present is considered.

4 Feature Extraction

In machine learning and pattern recognition problems, features play a key role in extracting dominant component of input sequence data. Features represents the dominant component that may be least or frequently occurring or can be derived from input sample data [41]. The importance of feature extraction are

- Reduce input data to Neural Network.
- Reduce redundancy present in input data.
- Provides an accurate representation of input data.

On the other hand feature extraction always add extra computational burden to extract them especially in frequency domain feature extraction. We should also care the feature which are distinct for different classes of signals. The features used in our problem of urban sound classification are given in the next section.

4.1 Features Used for Training the Neural Network

Definition 1 Zero Crossing Rate

It is rate at which the signal changes its phase i.e. the rate at which the signal changes its sign from positive to negative and vice versa. It is defined as.

$$Z_{ri} = \frac{1}{L} \sum_{n=0}^{L-1} I_{R<0}(|w_i[n]|^2 - 1) \tag{5}$$

where $w_i[n]$ is a ith windowed signal of length L and $I_{R<0}$ is the indicator function [41].

Definition 2 Energy of the Signal

It is the total **average Energy** present in an audio frame of window size L.

$$E_i = \frac{1}{L} \sum_{n=0}^{L-1} |w_i[n]|^2 \tag{6}$$

Definition 3 Spectral Centroid.

The **spectral centroid** is defined as the center of gravity of the magnitude spectrum of the Fourier transform [41].

$$C_i = \frac{\sum_{n=0}^{L-1} f_i[n] w_i[n]}{\sum_{n=0}^{L-1} w_i[n]} \tag{7}$$

where $x(n)$ is the input signal and $f_i[n]$ is the discrete Fourier transform of $w_i[n]$ given by

$$f_i[n] = \mathcal{F}\{w_i[n]\} \tag{8}$$

having same length as that of input windowed signal $w_i[n]$.

Definition 4 Spectral Rolloff

The **spectral rolloff** is defined as the frequency f_t below which 85% of the magnitude distribution is concentrated. The mean and the variance of the rolloff across time frames in the window are used as features [41].

Let E_n be the total energy present in an audio frame of size L in it magnitude spectrum of discrete Fourier Transform $X(k)$ given by Eq. (9).

$$W_i(k) = \sum_{n=0}^{L-1} w_i[n] e^{-j\frac{2\pi nk}{L}} \tag{9}$$

The total energy present in an audio frame is given by Eq. (10)

$$E_{w_i} = \sum_{k=0}^{L-1} |W_i(k)|^2 \tag{10}$$

for calculating the roll off frequency f_t, we consider frequency in the discrete Fourier transform domain k_t such that we have

$$|W_i(k_t)| \leq cE_{w_i} \tag{11}$$

where $c \in [0, 1]$, in order to calculate roll off frequency we generally keep $c = 0.85$. The rolloff frequency is then calculated as per Eq. (12).

$$f_t = \frac{k_t}{L} \tag{12}$$

Definition 5 Spectral Flux

The **Spectral flux** is the measure of how quickly the power spectrum of a signal is changing, calculated by comparing the power spectrum for one frame against the power spectrum from the previous frame.

Consider $W_i(k)$ as the discrete Fourier transform of the audio frame $w_i[n]$ and $W_{i-1}(k)$ as the DFT of the previous audio frame $w_{i-1}[n]$ as described in Eq. (9). We normalize $W_i(k)$ and $W_{i-1}(k)$ as

$$\widetilde{W}_i(k) = \frac{W_i(k)}{\sum_k W_i(k)} \tag{13}$$

$$\widetilde{W}_{i-1}(k) = \frac{W_{i-1}(k)}{\sum_k W_{i-1}(k)} \tag{14}$$

then we calculate the spectral flux as shown in Eq. (15).

$$\psi_{ni} = \sum_k |W_i(k) - W_{i-1}(k)|^2 \tag{15}$$

Definition 6 Mel-Frequency Cepstrum

In sound processing, the **Mel-Frequency Cepstrum (MFC)** is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC [41].

Let us consider $W_i(k)$ as the Discrete Fourier transform of the audio frame $w_i[n]$ described in (9). Then map the power spectrum to a Non-linear Mel scale as shown in (16)

$$\zeta_{ni}[n] = 10\log_{10}(|W_i(k)|^2 + \epsilon) \tag{16}$$

where generally we take $\epsilon > 1$ and $n = k$ since the length of DFT and $w_i[n]$ is the same. MFCCs are then calculated by taking DCT of log power spectrum $\zeta_{ni}[n]$ as if it were a signal. The resulting MFCC is expressed in Eq. (17).

$$\text{MFCC}[k] = \sum_{n=0}^{L-1} \zeta_{ni}[n] \cos\left(\frac{\pi k}{2L}(2n+1)\right) \tag{17}$$

Now, two examples are provided to understand how features are extracted form the audio signal. The first example is chosen to extract the feature from a single frequency sinusoidal signal using Hamming window. Where as an audio signal is considered in Example 2.

Example 1 Extracting the features of a single Toned input signal (Fig. 2).

Given:

$$x(t) = \cos(2\pi f_o t) \qquad , 0 \le t \le 1 \text{ s} \tag{18}$$

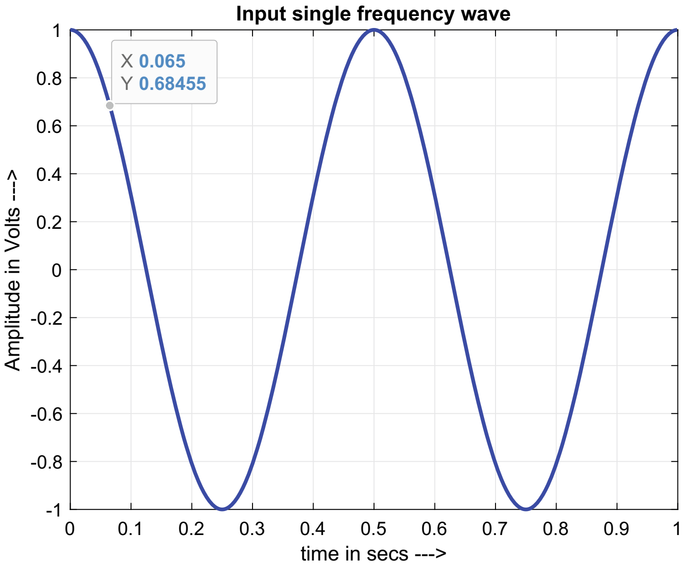


Fig. 2 Input signal with single frequency

The Signal $x(t)$ has a single frequency $f_o = 2$ Hz is sampled with sampling frequency f_s=1 KHz. Extract the features of the given signal using a Hamming window with parameters $L = \Delta = 0.5$ s.
Solution: We apply a hamming window duration $L = 0.5$ s, the hamming window is shown in Fig. 3. Number of frames that will be generated with the application of hamming windows number of frames that will generated will be given as

$$\begin{aligned} \text{number of frames} &= \lfloor \frac{T - L}{\Delta} \rfloor + 1 \\ &= \lfloor \frac{1 - 0.5}{0.5} \rfloor + 1 \\ &= 2 \text{ frames} \end{aligned}$$

The 2 Frames of the hamming window are shown in Fig. 4. The first frame of hamming window is placed at 0 to 0.5 s and the next frame of hamming window is placed at Δ to $\Delta + 0.5$ s.

After multiplying the corresponding Window frame with signal frame we obtain the output as shown in Fig. 5. The features are then calculated for every frame. The first feature Zero Crossing rate is calculated as for frame 1 output as shown in Fig. 5 using Eq. (32) as $Z_{r1} = 0.0050$

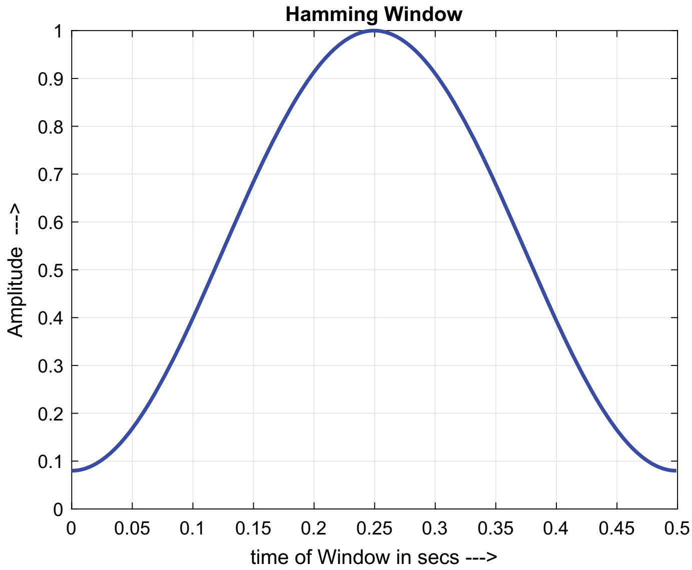


Fig. 3 Hamming window of duration $L = 0.5$ s

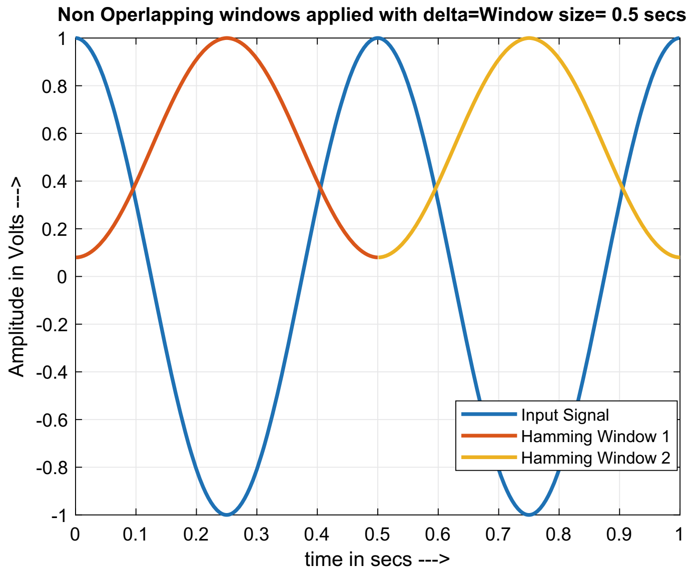


Fig. 4 Positing of hamming window of duration $L = 0.5$ s

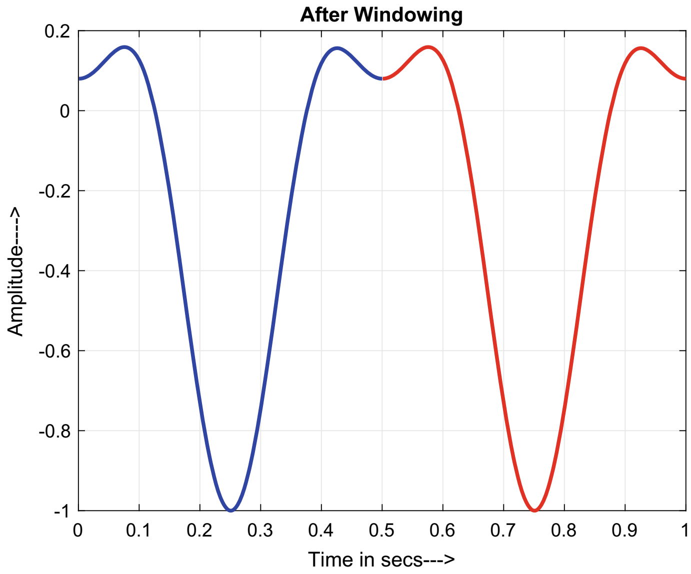


Fig. 5 output after multiplication Hamming Window with corresponding signal frames, each window having a duration of $L = 0.5$ s. Blue and red curve indicate the output of Hamming Window 1 and 2 respectively

Similarly, for the second frame output shown in Fig. 5 we calculate Zero Crossing rate as $Z_{r2} = 0.0050$.

as we guess the two values shall be equal since the output is same. in array format we can write the feature vector for Zero Crossing rate as

$$\begin{aligned} \boldsymbol{Z_r} &= [Z_{r1} \quad Z_{r2}] \\ \boldsymbol{Z_r} &= [0.0050 \quad 0.0050] \end{aligned} \tag{19}$$

Similarly, we can calculate other features as

$$\begin{aligned} \boldsymbol{E_n} &= [0.2249 \quad 0.2249] \\ \boldsymbol{C_n} &= [0.0073 \quad 0.0073] \\ \textbf{Spectral Roll off} &= [0.2986 \quad 0.2941] \times 10^{-26} \\ \boldsymbol{\psi_n} &= [0.0050 \quad 0.0050] \end{aligned} \tag{20}$$

$$
\mathbf{MFCC} = \begin{bmatrix} -83.5699 & -83.5699 \\ 18.5237 & 18.5237 \\ 10.4654 & 10.4654 \\ 1.9459 & 1.9459 \\ -3.0794 & -3.0794 \\ -3.3539 & -3.3539 \\ -0.6482 & -0.6482 \\ 1.9628 & 1.9628 \\ 2.4191 & 2.4191 \\ 0.8621 & 0.8620 \\ -0.9950 & -0.9950 \\ -1.5487 & -1.5487 \\ -0.5803 & -0.5803 \end{bmatrix}
$$

Example 2 Extracting the features of a sample

Given: Window Size - 1 s; step size $\Delta = 0.2$ s; sampling frequency $F_s = 44.1$ kHz

Solution:
$E_n = \frac{1}{L}\sum_{t=0}^{L-1} |s(t)|^2 = 0.0138$
$Z_r = \frac{1}{T}\sum_{t=0}^{T-1} I_{R<0}(|s(t)|^2 - 1) = 0.0598$
Spectral Centroid = 1.6213×10^4
Spectral Rolloff = 1.6285×10^4
Spectral Flux = 0.5813×10^{-4}
The Mel-Frequency Cepstral Coefficients for the audio signal shown in Fig. 6 is listed in Table 1.

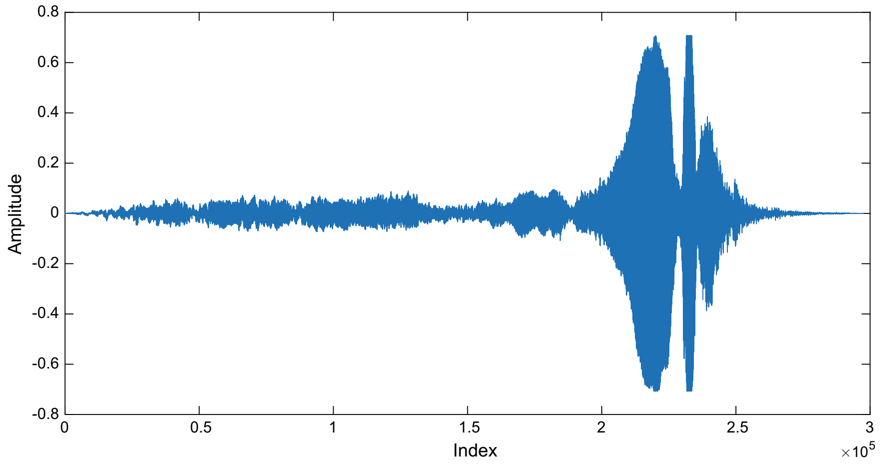


Fig. 6 Input sound signal in Example 1

Table 1 MFCC feature matrix 11 × 12 of signal given in Fig. 6

−9.817	−27.237	2.840	−0.932	−0.399	0.709	0.987	0.649	−0.352	0.033	0.531	0.574
−9.116	−26.090	3.146	−1.369	−0.578	0.342	1.071	0.645	−0.092	−0.097	0.126	0.587
−8.220	−24.255	3.215	−1.374	−0.065	0.339	0.463	0.763	−0.245	−0.231	−0.244	0.265
−7.270	−23.404	3.253	−1.420	−0.277	0.221	0.493	0.794	−0.112	−0.048	−0.262	0.020
−6.028	−23.558	3.844	−1.195	−0.041	0.310	0.886	1.034	0.521	0.122	−0.284	−0.501
−5.916	−23.585	3.955	−1.149	0.006	0.190	0.617	1.119	0.713	0.414	-0.024	−0.732
−5.856	−23.610	3.669	−1.412	−0.151	−0.154	0.590	0.999	0.535	0.650	0.160	−0.677
−6.387	−23.227	3.384	−1.160	−0.251	−0.282	0.495	0.594	0.712	0.884	0.391	−0.802
−5.745	−22.811	3.215	−0.978	−0.063	−0.447	0.234	0.751	0.934	0.847	0.694	−0.681
−5.448	−23.041	3.210	−0.939	0.420	−0.498	0.119	0.830	0.832	1.030	1.159	−0.172
−5.302	−22.213	3.927	−0.844	0.263	−0.703	−0.077	0.604	0.961	0.901	0.850	0.066

5 Designing the Cost Function for Bayesian Regularised Neural Network

In this section, how to design a neural network by defining its input layer, hidden layer and output layer is discussed. Of all the neural network tested for binary classification, it is found that the Bayesian regularized neural networks outperformed over all other neural networks which shall be discussed further.

5.1 Over-Fitting and Regularization of Neural Networks

Over-fitting is one of the most common problems that is faced when solving a classification problem. It occurs when the classification algorithm performs exceptionally well for the training data and its performance does not reflect in the test data. Different regularization methods seek to reduce over-fitting by making certain changes in the learning algorithm that allow the neural network model to generalize better. In deep learning this is done by penalizing the weight matrices such that no particular node gets more weight then required.

5.2 Bayesian Regularization

Bayesian regularization is a regularization technique for artificial neural networks that relies on Bayes theorem as shown in Eq. (21) in order to provide an alternate cost function for training the neural network.

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{P(A_1)P(B|A_1) + P(A_2)P(B|A_2) + \cdots + P(A_n)P(B|A_n)}$$

Here A_i refers to any one of the event $A_1, A_2, \ldots, A_n$ and B refers to a test event.

Such artificial neural networks which are trained using the Bayesian rules are called **Bayesian Regularized Artificial Neural Networks** or **BRANNs** [24, 40]. These are considered to be more robust compared to standard back-propagation networks and also difficult to over-train because they give a Bayesian criterion to stop training.

5.2.1 Cost Function for Bayesian Regularized Neural Network

The Cost function for BRANNs is calculated based on maximum likelihood estimation (MLE) which states that minimizing the MSE (mean squared error) is same as maximizing the log probability density of the correct output.

Consider $y_i = f(\mathbf{W}, x_i)$, where y_i is an output trained on the input x_i and weight vector $\mathbf{W}$. Now if the output is affected by Gaussian noise, the probability of getting target value t_i is given by Eq. (21).

$$P(t_i|y_i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[\frac{-1}{2}\left(\frac{t_i - y_i}{\sigma_n}\right)^2\right] \tag{21}$$

We are simply stating that the probability density of the target value given the output after applying Gaussian Noise is the Gaussian distribution centered around the output.

Applying $-\log$ on both sides of Eq. (21), we get

$$-\log(P(t_i|y_i)) = -\log\left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right) + \frac{1}{2}\left(\frac{t_i - y_i}{\sigma_n}\right)^2 \tag{22}$$

let $-\log\left(\frac{1}{\sqrt{2\pi\sigma_n^2}}\right) = K$, where K is a constant. Therefore, Eq. (22) now becomes

$$-\log(P(t_i|y_i)) = K + \frac{1}{2}\left(\frac{t_i - y_i}{\sigma_n}\right)^2 \tag{23}$$

When we are working on multiple training pattern (x_i, y_i) in the training set D, it is intend to maximize the product of probabilities as shown in (24).

$$P(D|\mathbf{W}) = \prod_i P(t_i|y_i) \tag{24}$$

In other words, it can be said that the probability of a training set D given weight vector $\mathbf{W}$ is product of individual probability of i_{th} target t_i given i_{th} output y_i.

Equation (24) can be written in log domain as the sum of individual probabilities. Now, applying log on both sides, we have

$$-\log(P(D|\mathbf{W})) = \sum_i \log(P\ (t_i|y_i)) \tag{25}$$

$$= \sum_i \frac{1}{2}\left(\frac{t_i - y_i}{\sigma_n}\right)^2 \tag{26}$$

Equation (26) is the log probability of observed output y_i, given a weight vector **W** that helps in maximizing the log probability density of the output to be closer to the target value considering the output distribution is Gaussian.

Now, with given weight vector **W** and training set D, we can write using Bayes theorem as

$$P(\mathbf{W}|D) = \frac{P(\mathbf{W})p(D|\mathbf{W})}{\int P(\mathbf{W})P(D|\mathbf{W})} \tag{27}$$

where,

- $P(\mathbf{W}|D)$ is the posterior probability of the weight vector **W** given the training set D.
- $P(\mathbf{W})$ is the prior probability of the weight vector.
- $P(D|\mathbf{W})$ is the probability of the observed data given weight vector **W**.
- And, the denominator is the integral of all possible weight vectors.

The cost function is then calculated using Eq. (28)

$$\begin{aligned} Cost &= -\log(P(\mathbf{W}|D)) \\ &= -\log(P(\mathbf{W}) - \log(P(D|\mathbf{W}))) + \log(P(D)) \end{aligned} \tag{28}$$

where, $P(\mathbf{W})$

$$P(\mathbf{W}) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{\frac{-1}{2}\left(\frac{\mathbf{w}}{\sigma_w}\right)^2} \tag{29}$$

Therefore, the Bayesian Inference for Maximum Aposteriori rule is as follows:

$$\begin{aligned} \text{Cost} &= -\log(P(\mathbf{W}|D)) \\ &= -\log(P(\mathbf{W}) - \log(P(D|\mathbf{W}))) + \log(P(D)) \\ &= \frac{1}{2\sigma_w^2}\sum_k w_k + \frac{1}{2\sigma_D^2}\sum_i (t_i - y_i)^2 + K \\ &= \frac{1}{2\sigma_w^2}\sum_k w_k + \frac{1}{2\sigma_D^2}\sum_i (t_i - y_i)^2 \\ &= \frac{1}{\sigma_D^2}\left[\frac{\sigma_D^2}{2\sigma_w^2}\sum_k w_k^2 + E\right] \end{aligned} \tag{30}$$

This training algorithm is used in different neural networks to classify urban noise. And surprisingly it is found that the Bayesian training algorithm provides best performance over others.

6 Bayesian Regularised Neural Networks for Urban Sound Noise Classification: A Deep Learning Approach

In this section, we illustrate how can design Bayesian regularized neural network in MATLAB. Then the use of neural network (NN) for the classification of urban noise is explained. Various neural networks available in MATLAB, their design, performance verification and comparison are discussed first for binary classification. Only the presence of ambulance sound in urban cities is detected. Then, we discuss a novel approach where two inputs are provided together to the NN model, which improves the structure and prediction accuracy of classifying ambulance sound. Detection of more than two sources present at a time, that is multi-label detection from a recorded signal in real time in urban traffic is illustrated. The simulation has been done using MATLAB and the data set used which is available in repository [25].

6.1 Deep Neural Network Design

The basic neural networks which are available in MATLAB such as the feed forward net (FFNet), cascade forward Net (CFNet), pattern net (Pnet) and fit net (FitNet) is discussed here. Each of the network can be trained by choosing one of the training algorithms. Neural networks provided by MATLAB and their architectures are shown in Figs. 7, 8, 9 and 10. All the networks are designed with 1 input layer, 10 hidden layers and 1 output layer.

The deep neural network model can be visualized as shown by the block diagram in Fig. 1 as the 1 input-output model. We feed an input consisting a column vector of 35 features and produces output 1 if ambulance sound is present and 0 otherwise for binary classification. Then the number of occurrence of ones to determine the percentage contribution of ambulance from other sounds.

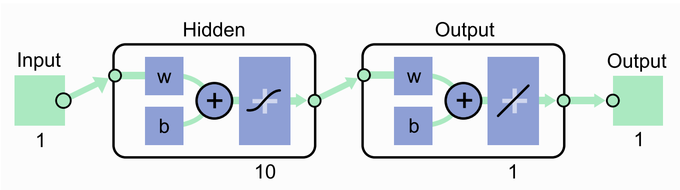


Fig. 7 A feed forward network designed in MATLAB

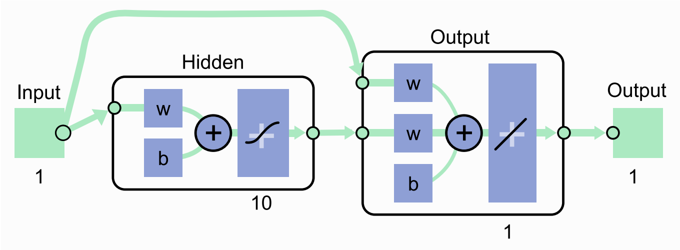


Fig. 8 A Cascade forward network designed in MATLAB

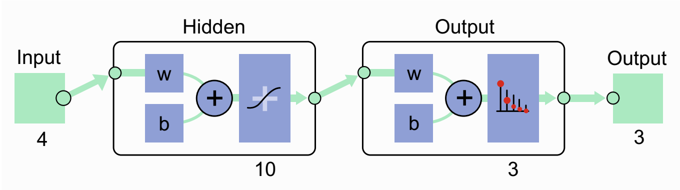


Fig. 9 A Pattern network designed in MATLAB

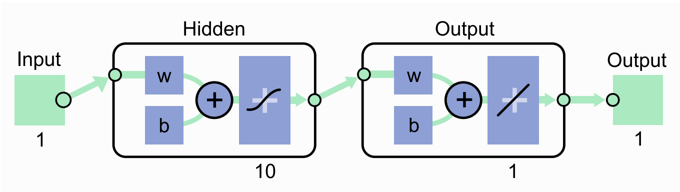


Fig. 10 A Fit network similar to feed forward network designed in MATLAB

6.1.1 Choosing the Training Methods

Different training algorithms are available for training the previously mentioned network. The training methods provided by MATLAB are listed in the Table 2. All these training algorithms are used to train the 4 different kinds of neural network for binary classification to know their corresponding mean square error performance. The available data set is used by employing the methods provided.

6.1.2 Performance Analysis for Different Training Methods

The training methods as listed in Table 2 are used in different neural networks such as the FFNet, CFNet, PNet and FitNet Networks. The input is 35 features of the merged combination of ambulance and other noise mixed randomly which is common to all the networks. The performance of neural network is calculated as a mean squared

Table 2 Different training algorithms for multilayer shallow neural networks

Acronym	Function	Algorithm
LM	Trainlm	Levenberg-marquardt
BR	Trainbr	Bayesian regularization
BFG	Trainbfg	BFGS quasi-newton
RP	Trainrp	Resilient backpropagation
SCG	Trainscg	Scaled conjugate gradient
CGB	Traincgb	Conjugate gradient with powell/beale restarts
CGF	Traincgf	Fletcher-powell conjugate gradient
CGP	Traincgp	Polak-ribire conjugate gradient
OSS	Trainoss	One step secant
GDX	Traingdx	Variable learning rate gradient descent
GDM	Traingdm	Gradient descent with momentum
GD	Traingd	Gradient descent

Table 3 MSE performance comparison of neural networks with different training methods

Training algorithm	Feed forward network	Cascade forward network	Pattern net	Fit net
LM	10^{-10}	10^{-4}	10^{-8}	10^{-10}
BR	10^{-14}	10^{-11}	10^{-9}	10^{-15}
BFG	10^{-4}	10^{-3}	10^{-9}	10^{-3}
RP	10^{-3}	10^{-3}	10^{-6}	10^{-3}
SCG	10^{-5}	10^{-3}	10^{-7}	10^{-4}
CGB	10^{-2}	10^{-3}	10^{-11}	10^{-3}
CGF	10^{-3}	10^{-4}	10^{-12}	10^{-4}
CGP	10^{-5}	10^{-4}	10^{-11}	10^{-2}
OSS	10^{-4}	10^{-3}	10^{-5}	10^{-4}
GDX	10^{-3}	10^{-3}	10^{-6}	10^{-3}
GDM	10^{-2}	10^{-2}	10^{-2}	10^{-3}
GD	10^{-2}	10^{-2}	10^{-2}	10^{-2}

error (MSE) of the difference between the training and validation set. The MSE performance of different neural networks is listed in Table 3 [25].

It is found while comparing all the methods in Table 3 that the Fit neural network provides the best performance with the Bayesian regularization as the training method. The second best is the feed forward network with the same Bayesian regularization training algorithm. Therefore, classification of noise sound is to be done using these two afore said methods in the simulation section.

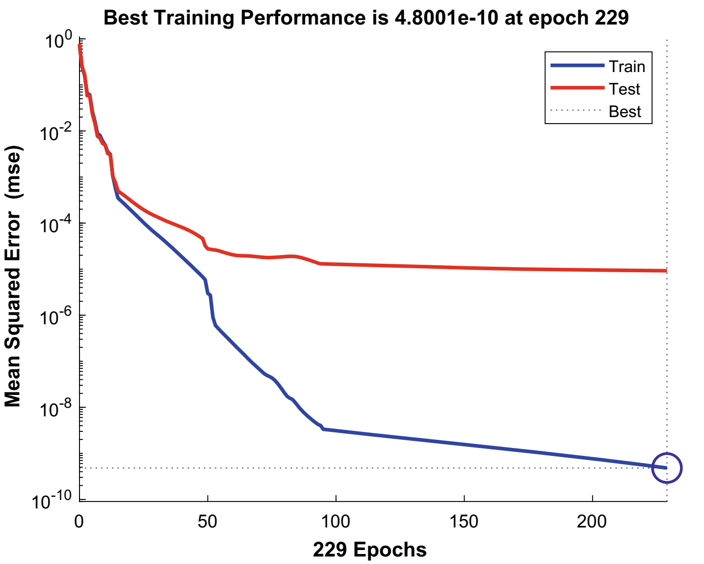


Fig. 11 MSE performance of feed forwardNet

6.1.3 Training Performance of FFNet and FitNet Neural Network

The networks are trained by using the available data. The feature are extracted from each of the sound signal and then used for training. The training performance of FFNet and FitNet neural network is shown in Figs. 11 and 12 for two classes, where the MSE is plotted as a function of epochs. Here 70% of total available data is used for training and remaining 30% used for testing. It is as expected that the training gives better performance over the testing.

It has been seen from Figs. 11 and 12 that the MSE performance of FitNet is better the FFNet in both training and testing. The MSE is much below 10^{-4} in case of FitNet during testing, whereas the FFNet provides nearly 10^{-4}.

6.1.4 Classification Output

An input of 35 features from merged combination of ambulance and other sounds treated as noise is provided to the model. The model suppose to produce the output 1 if ambulance is detected and 0 otherwise. But it may not produces always exact 0 or 1. Therefore, we then pass the model output through a hardlimiter with threshold $\lambda = 0.5$. The output of the hardlimiter is depicted in Figs. 13 and 14 for Feed ForwardNet and FitNet.

Comparison between the contribution of ambulance sound detected by the neural network is shown as a pie chart. For an input data which consisted of mixed com-

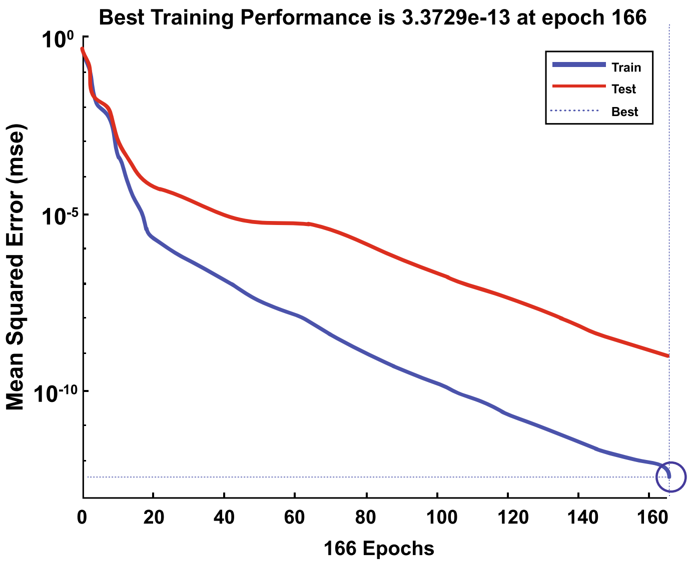


Fig. 12 MSE performance FitNet

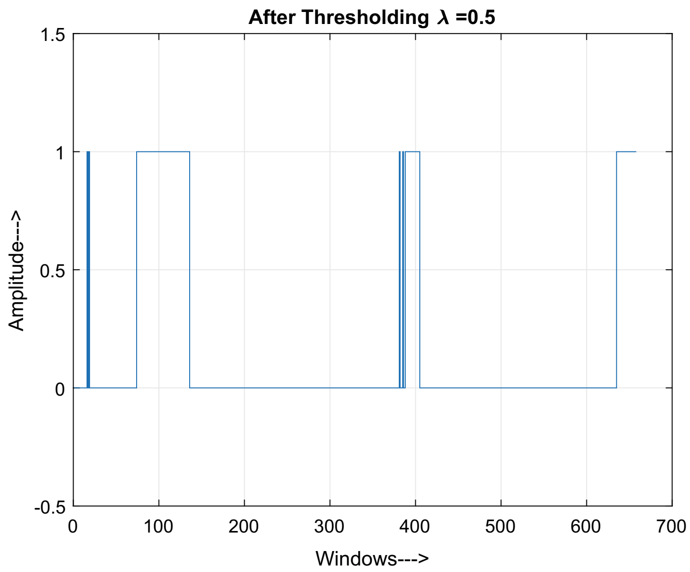


Fig. 13 Output of feed forward net with threshold when the input is a mixture ambulance and other sounds

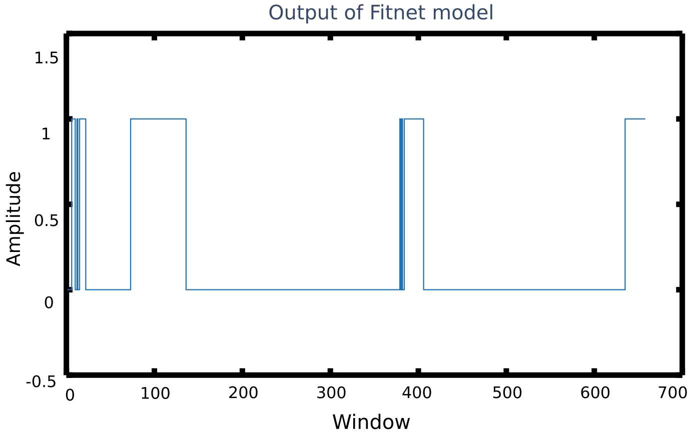
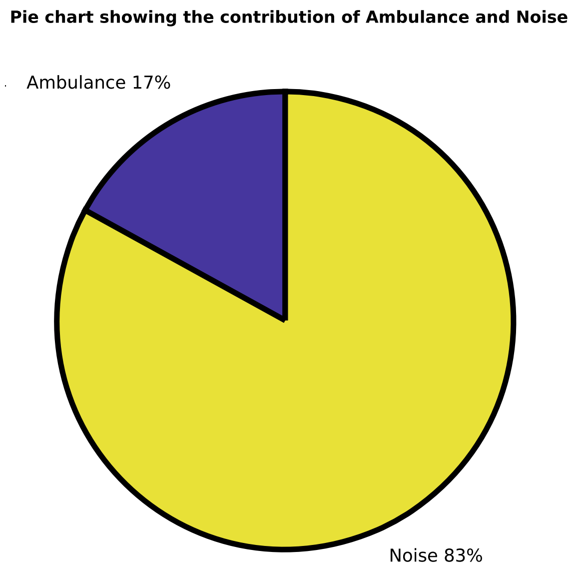


Fig. 14 Output of fitNet with threshold when the input is a mixture ambulance and other sounds

Fig. 15 Contribution of each sound by FFNet by using Pi chart



bination of ambulance sound and other sound treated as noise. It is found that the FitNet NN classified 19% of the sound contributed by ambulance and rest by noise as shown in the Pie chart Fig. 15. Whereas, the FFNet NN classified that 17% of the sound is contributed by ambulance and rest by noise which is shown in Fig. 16. So there is a 2% error in categorizing by FFNet model with respect to FitNet.

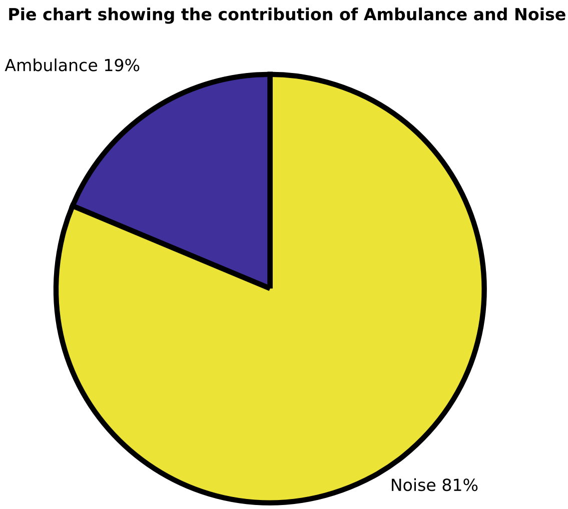


Fig. 16 Contribution of each sound by FitNet by using Pi chart

6.2 Novel Design: Two Input-Output Neural Network Model

Previously, only one input array of 35 features for each of the signals is given to the neural network and corresponding only one output is obtained. Now, the model is extended by giving two input sets, each of 35 features at a time and then two corresponding output consisting of ones if ambulance has been detected and zero otherwise. The process is indicated in the block diagram as shown in Fig. 17.

The input and output are specified as column vector as given below

$$\text{output} = \begin{cases} [0\ 0]^T & \text{if input} = [\text{Noise Noise}]^T \\ [0\ 1]^T & \text{if input} = [\text{Noise Ambulance}]^T \\ [1\ 0]^T & \text{if input} = [\text{Ambulance Noise}]^T \\ [1\ 1]^T & \text{if input} = [\text{Ambulance Ambulance}]^T \end{cases} \quad (31)$$

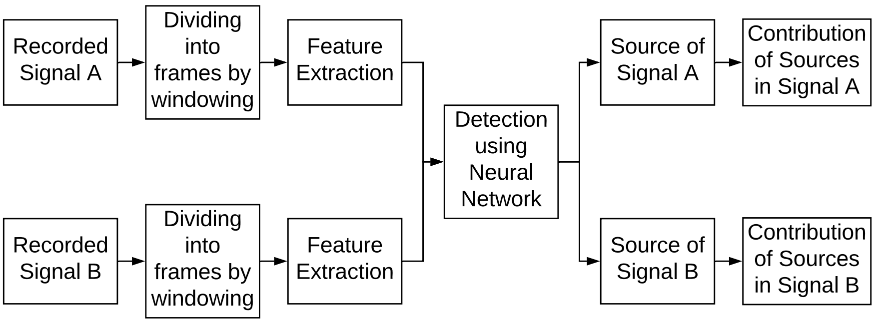


Fig. 17 Block diagram of two input-output NN model for urban noise classification

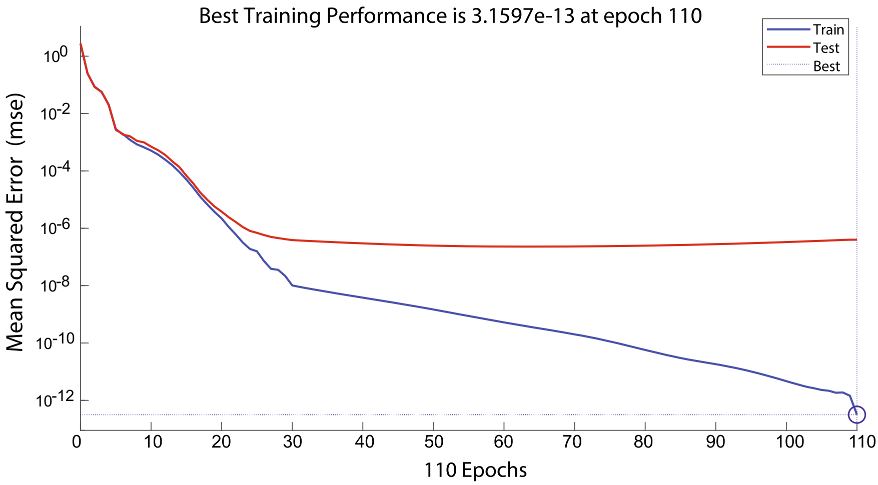


Fig. 18 Performance of two input-output NN model

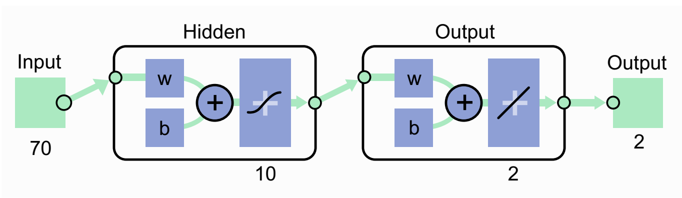


Fig. 19 Two Input-output fit neural network model

6.2.1 Performance Analysis

It has been found from the figures that the performance of the two input-output model was better as compared to the existing 1 input-output model which is depicted in Fig. 18. Thus it can be concluded that, shuffling of data increased accuracy of model to be able to classify sounds efficiently.

6.2.2 Output of the Two Input-Output Model

The input to neural network is a column vector of 2 input feature sets of permutation of ambulance and Noise. Each input has a 35 feature sets so when we combine them in column form the model visualizes it as 70 feature sets and produces two outputs 0 and 1. The output can be recognized by the form represented by Eq. 31.

Figure 20 shows the all possibles states that the output can have for all possible combinations of input data ambulance and noise.

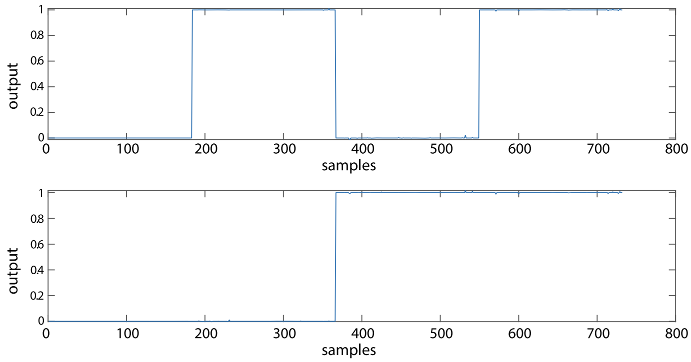


Fig. 20 Output of two input-output neural network model

6.3 Classification of Multiple Sound Source

In real-time scenario, noise is generated by various sources and hence it is imperative that the number of sources detected is to be increased. In this subsection, a more complex network, having greater number of layers is used in order to tackle this relatively challenging problem. This results in increase in size of output vector to K number of classes detected.

6.3.1 Single Label Detection of K Classes

The single-label detection of N classes suggests that in a particular window only one class will be detected out of the fixed classes i.e. only one output out of the 10 output layer nodes will stand high and others low. This is typically known as the one hot encode method. Further elaboration can be noticed as per Eq. (32).

$$y = [c[0]\ c[1]\ c[2]\ ...\ c[K-1]] \tag{32}$$

where N is number of total classes. Consider $c[i] \in [c[0]\ c[1]\ c[2] \ldots c[K]]$ then we have

$$c[i] = \begin{cases} 1 & \text{if } i\text{thclass is detected} \\ 0 & \text{if } i\text{thclass is not detected} \end{cases} \tag{33}$$

The model is trained using Urban S8 K dataset over 100 epochs and it provides a test accuracy of 95.2%. The predicted contribution of 10 classes for a test data samples is shown in the bar graph depicted in Fig. 21.

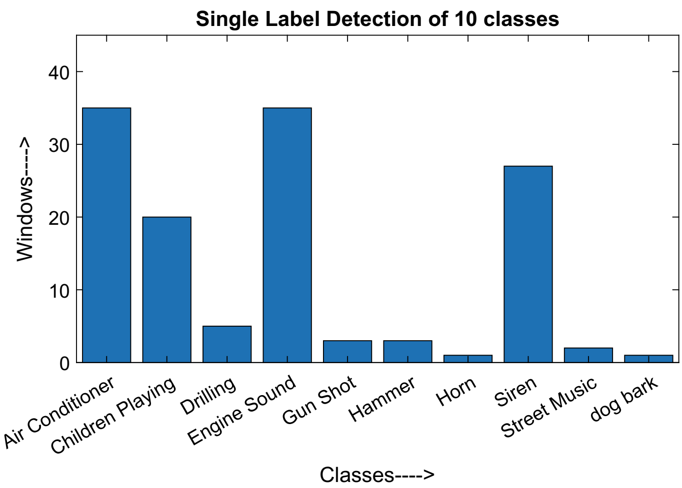


Fig. 21 Bar graph estimated from the model for a test signal

6.3.2 Multi-label Detection of *N* Classes

The term Multi-label detection refers to the detection of more than one class in a particular window. Multi-label classification could also be achieved provided the window size is kept as small as possible but in order to reduce the computational complexity keeping window size small is not advisable. The Multi-label classification is to be trained over all mixed combination of audio signals so if we have a dataset of K classes, the model needs to be trained over new dataset of $2^K - 1$ classes consisting of all possible combination. The Multi-label classification is further elaborated as per Eq. (34)

$$y = [c[0]\ c[1]\ c[2] \ldots c[K-1]] \tag{34}$$

where N is the number of classes. Consider Eq. (35)

$$c[i], c[i+1], c[i+2] \ldots, c[i+k] \in [c[0]\ c[1]\ c[2] \ldots c[K-1]] \tag{35}$$

where k is an arbitrary constant. So according Multi-label classification we have

$$c[i], c[i+1], c[i+2] \ldots, c[i+k] = \begin{cases} 1 & \text{if } C > 1 \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

where $C > 1$ represents more than one class detected i.e. $i, i+1, \ldots, (i+k)$th class is detected.

Table 4 Making a mixed dataset for multi-label classification

S. No.	a1	a2	a3	a4	Class
1	0	0	0	1	a4
2	0	0	1	0	a3
3	0	0	1	1	a3, a4
4	0	1	0	0	a2
5	0	1	0	1	a2, a4
6	0	1	1	0	a2, a3
7	0	1	1	1	a2, a3, a4
8	1	0	0	0	a1
9	1	0	0	1	a1, a4
10	1	0	1	0	a1, a3
11	1	0	1	1	a1, a3, a4
12	1	1	0	0	a1, a2
13	1	1	0	1	a1, a2, a4
14	1	1	1	0	a1, a2, a3
15	1	1	1	1	a1, a2, a3, a4

Now consider, we have 4 classes to be classified which are say a1, a2, a3, a4. So in order to train the model we shall need $2^4 - 1$ i.e 15 classes which should contain all possible combinations of the 4 classes a1, a2, a3, a4. This method of assigning labels and designed mixed classes is well illustrated in Table 4, where 1 corresponds to particular class present and 0 represents that the particular class is absent.

6.3.3 Multi-label Detection of *K* Classes by Keras Model

The model is trained for 15 classes which is a combination of 4 classes, which are Ambulance Siren, Engine, Children playing and Horn over 1000 epochs. The model is able to classify with a test accuracy of 72.5%. The actual contribution and the predicted contribution is depicted in Figs. 22 and 23.

6.4 Novel Approach: Multi-label Classification Using Bayesian Regularized Fitnet Model and Deep Lerning Approach

The latter model was only able to achieve an accuracy of 72.5% which is inadequate for practical application in the real-life scenario. In this section we return to our older model Bayesian Regularized Fitnet model. This model is able to achieve accuracy

Fig. 22 Pie chart showing actual contribution

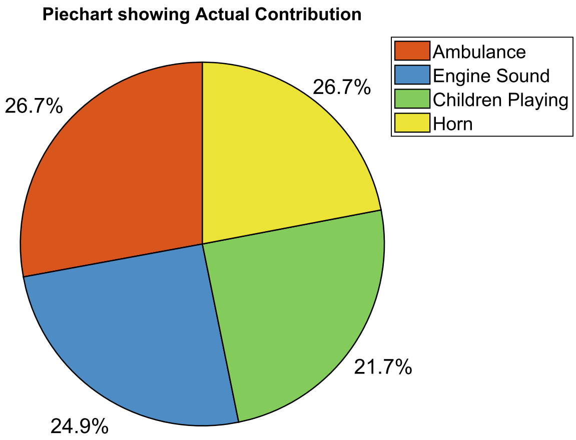
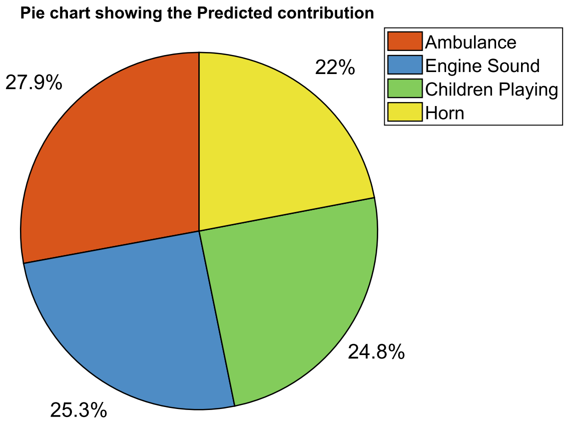


Fig. 23 Pie chart showing predicted contribution



of 95.14% which is considerable improvement over the latter model. It should also be noted that this performance is achieved with a smaller feature vector of size 35. The network was tested for two cases i.e. when the input had three classes and when the output had four classes. The performance of Multi-Label Bayesian regularised FitNet Model (**MLBRANN**) is shown in Fig. 24.

The **MLBRANN** model is tested for an audio signal consisting of mixed combination of labels from a set of ambulance siren, jack hammer, children playing and horn sampled at 44.1KHz. The actual output, and predicted output from the **MLBRANN** model is shown in Figs. 25 and 26.

The composition of the four classes present in the test signal is visulised using a Bar graph as shown in Fig. 27.

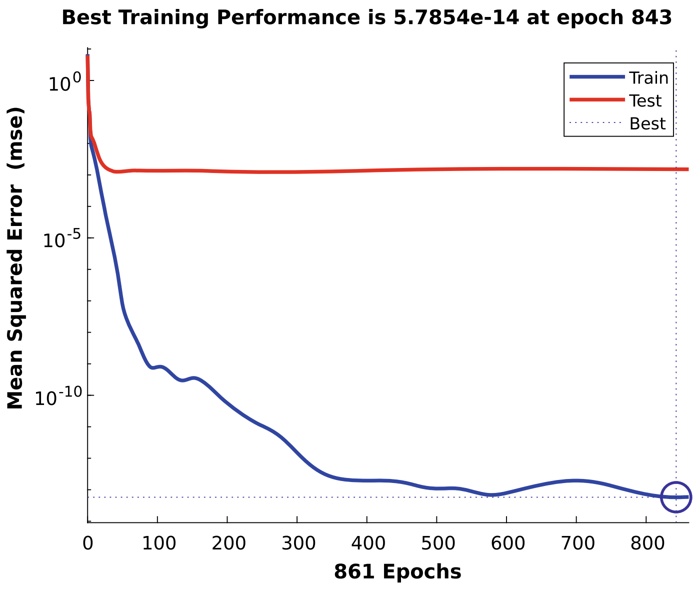


Fig. 24 MSE performance of multi-label Bayesian regularised FitNet model

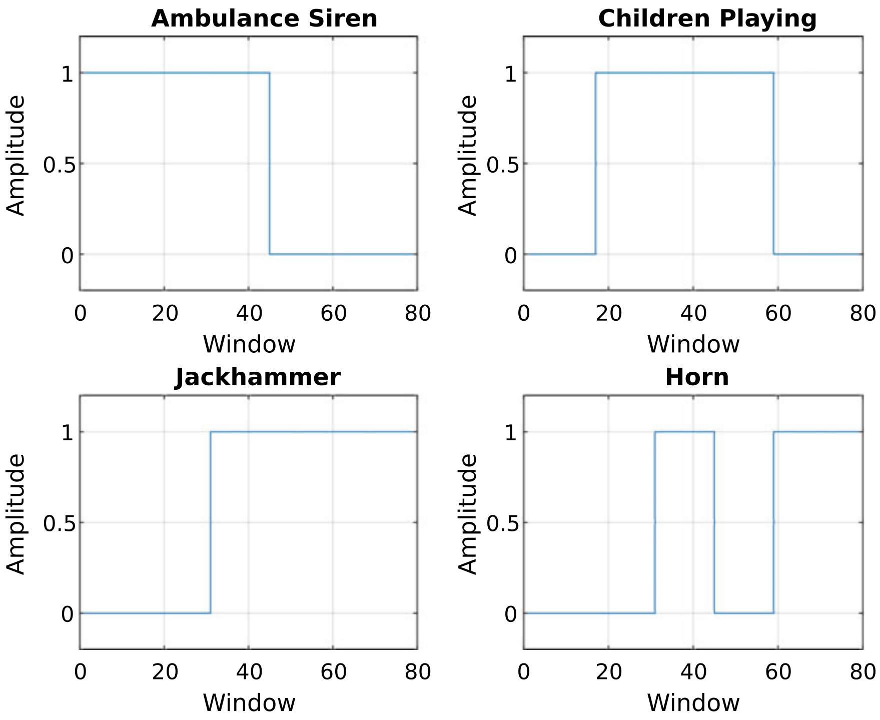


Fig. 25 Actual output of the test signal

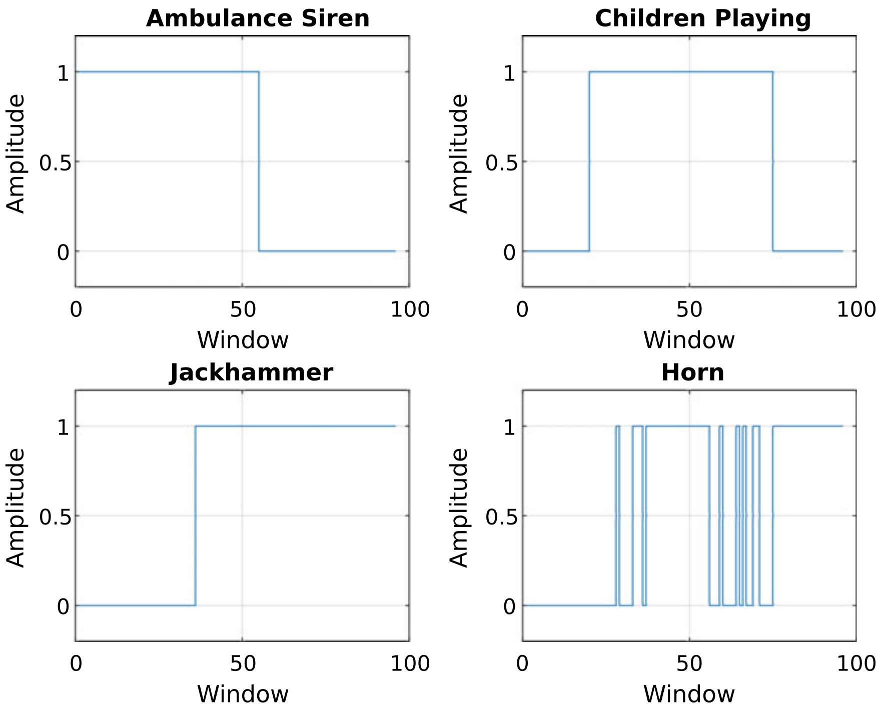


Fig. 26 Predicted output using **MLBRANN** model

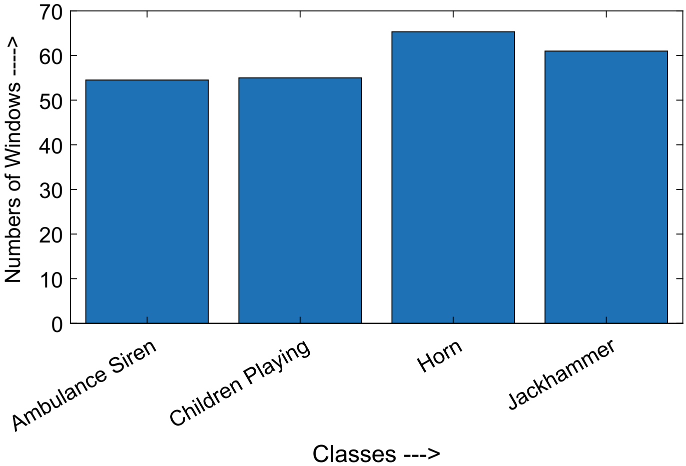


Fig. 27 Predicted contribution of classes using MLBRANN model

7 Conclusion and Future Work

The aim of this book chapter is to classify different urban sounds using neural network and deep learning algorithm. The sound recorded in real time in a urban city constitute of different noises. It always keen to know what are the different classes of noise present temporally at a particular place and their contributions. We have used different neural networks and training algorithms to do this, but the simulation results found that the Bayesian training algorithm in FitNet NN gives best performance. Initially, the NN is used to classify two classes that is presence of ambulance siren or not. The 35 feature vector is extracted from each window of the recorded signal (after dividing into windows) and given as input to the NN.

Then the idea has been extended to ten individual classes. The classification accuracy achieved is 95%. Further, we have also classified when different signals are present at a particular time (that is more than one class of signal present in a window) which is more challenging. It is because the feature vectors are not appropriate when multiple signals are present at a particular time. In fact, a new model is designed that could classify more two windows at a time. This helps in reducing delay for authentic time applications with increase in precision. In future the neural network can be trained for more number of sources recorded in Indian traffic and their contributions can be measured.

8 Problems

Note: MATLAB software may be used for solving the following problems.

8.1 Feature Extraction

Problem 1 Define the following features and give the necessary formulae of the same

(1) Zero crossing rate
(2) Energy of signal
(3) Spectral Centroid

Problem 2 Consider the discrete time signal $x[n] = [1, -1, 2, -3, -1, 3, -4, 6]$. Calculate the following features:

(1) Zero crossing rate
(2) Energy of signal

Are these features are unique for the signal? If not, give more examples of discrete time signals with these feature values.

Problem 3 Consider a discrete time signal $x[n]$, such that it's discrete Fourier transform (DFT) is given by $x[n]$? If yes, calculate the energy.

Problem 4 In the textual Example 1, the features of cosine wave having a duration $T = 1$ second is extracted. Now extract the features of sine wave with frequency $f_o = 3$ Hz sampled with sampling frequency $f_s = 2$ KHz having same duration given as

$$x(t) = \sin(2\pi f_o t), \qquad 0 \leq t \leq 1 \text{ s} \tag{37}$$

use a rectangular window instead of a Hamming window with parameters of the window given as

(1) $\Delta = 0.2$ s and $W_n = 0.2$ s
(2) $\Delta = 0.2$ s and $W_n = 0.5$ s
(3) $\Delta = 0.5$ s and $W_n = 0.2$ s

Problem 5 Repeat Problem 4 for an under-damped signal with same frequency f_o and sampling frequency f_s. The input signal is given as

$$x(t) = e^{-t} \sin(2\pi f_o t), \qquad 0 \leq t \leq 1 \text{ s} \tag{38}$$

Problem 6 Repeat Problem 4 for different windows given below and observe the difference.

(1) Triangular window
(2) Blackman window
(3) Hanning window.

Problem 7 Repeat Problem 4 for multi Toned signal with frequencies $f_o = 2$ Hz, $f_1 = 4$ Hz, $f_2 = 8$ Hz and $f_3 = 16$ Hz, sampled with sampling frequency $f_s = 3$ KHz. The input signal is given as

$$x(t) = \sin(2\pi f_o t) + \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t), \qquad 0 \leq t \leq 5 \text{ s}$$

Problem 8 Repeat Problems 4, 5, and 6, but this time use any recorded audio signal of duration T s and sampling frequency $f_s = 44.1$ KHz.

8.2 Designing of Optimal Neural Network

Problem 9 Different shallow neural networks are available in MATLAB such as

(1) Feed Forward Neural Network
(2) Cascade Forward Neural Network

(3) Pattern Neural Network
(4) Fit Neural Network.

Make a list of the performance of these networks in terms of mean squared error (MSE) for any two classes as per your choice. Train the neural network using a labeled data set for various training algorithms as illustrated in Table 3. Observe for which neural network and training algorithm you get the least MSE.

Problem 10 Repeat Problem 9 using the 2-input-output model whose architecture is illustrated in Fig. 19 and observe how the MSE is changed with time in terms of epochs taken for training the neural network.

8.3 Single Label Detection

Problem 11 Design a single label classifier to classify environmental sounds with maximum accuracy as discussed in Sect. 6.3. You may choose any five different classes from the Urban Sounds 8 k data set for your reference and make a labeled data set for the same. Make detection graphs and contribution in terms of bar graphs as discussed in that section.

8.4 Multi Label Detection

Problem 12 Design a Multi label Classifier to classify environmental sounds with maximum accuracy as discussed in sect. 6.4. You may choose any 3 different classes for your reference and make a labelled dataset all possible combinations i.e. 7 classes for the same. You may use the sound data from the Urban Sounds 8k dataset as discussed in that section. Make detection graphs and contribution in terms of bar graphs as discussed in that section.

Acknowledgements The authors would like to thank Science and Engineering Research Board (SERB), Department of Science and Technology, Government of India, for providing the financial support to carry out this work.

References

1. Theakston, F. (2011). Burden of disease from environmental noise-quantification of healthy life years lost in Europe. In *The WHO European Centre for Environment and Health*. Bonn Office: WHO Regional Office for Europe coordinated the development of this publication.
2. Brown, A. L. (2015). Effects of road traffic noise on health: From burden of disease to effectiveness of interventions. *Procedia Environmental Sciences*, *30*, 3–9.

3. Mavrin, V., Makarova, I., & Prikhodko, A. (2018). Assessment of the influence of the noise level of road transport on the state of the environment. *Transportation Research Procedia*, *36*, 514–519. System and digital technologies for ensuring traffic safety.
4. Zeeb, H., Hegewald, J., Schubert, M., Wagner, M., Dröge, P., Swart, E., Seidler, A. (2017). Traffic noise and hypertension–results from a large case-control study. *Environmental Research*, *157*, 110–117.
5. Seidler, A., Wagner, M., Schubert, M., Dröge, P., Römer, K., Pons-Kühnemann, J., & Hegewald, J. (2016). Aircraft, road and railway traffic noise as risk factors for heart failure and hypertensive heart disease-a case-control study based on secondary data. *International Journal of Hygiene and Environmental Health*, *219*(8), 749–758.
6. Hegewald, J., Schubert, M., Wagner, M., Dröge, P., Prote, U., Swart, E., et al. (2017). Breast cancer and exposure to aircraft, road, and railway-noise: A case-control study based on health insurance records. *Scandinavian Journal of Work, Environment & Health*, *6*, 509–518.
7. Sørensen, M., Andersen, Z. J., Nordsborg, R. B., Becker, T., Tjønneland, A., Overvad, K., & Raaschou-Nielsen, O. (2012). Long-term exposure to road traffic noise and incident diabetes: A cohort study. *Environmental health perspectives*, *121*(2), 217–222.
8. Han, B., & Hwang, E. (2009). Environmental sound classification based on feature collaboration. In *IEEE International Conference on Multimedia and Expo, New York*, pp. 542–545.
9. Torija, A. J., & Ruiz, D. P. (2016). Automated classification of urban locations for environmental noise impact assessment on the basis of road-traffic content. *Expert Systems with Applications*, *53*(1), 1–13.
10. Singh, D., Nigam, S. P., Agrawal, V. P., & Kumar, M. (2016). Vehicular traffic noise prediction using soft computing approach. *Journal of Environmental Management*, *183*, 59–66.
11. Das, H., Naik, B., & Behera, H. S. (2018). Classification of diabetes mellitus disease (dmd): A data mining (dm) approach. In P. K. Pattnaik, S. S. Rautaray, H. Das, & J. Nayak, (Eds.), *Progress in Computing, Analytics and Networking* (pp. 539–549). Springer Singapore.
12. Pradhan, C., Das, H., Naik, B., & Dey, N. (Eds.). (2018). *Handbook of research on information security in biomedical signal processing*. Hershey, PA: IGI Global.
13. Piczak,K. J. (2015). Environmental sound classification with convolutional neural networks. In *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6.
14. Andronicus, F., & Maheswaran. (2015). Intelligent ambulance detection system. *International Journal of Science, Engineering and Technology Research (IJSETR)*, *4* (5).
15. Barkana, B. D., & Saricicek, I. (2010). Environmental noise source classification using neural networks. In *IEEE Seventh International Conference on Information Technology: New Generations* (pp. 259–263). NV: Las Vegas.
16. Otálora, A. S., Osorio, D. E. C., & Moreno, N. C. (2017). Methods for extraction of features and discrimination of emergency sirens. *ARPN Journal of Engineering and Applied Sciences*, *12*(5)
17. Tran, V.-T., Yan, Y.-C., & Tsai, W.-H. (2017, March). Detection of ambulance and fire truck siren sounds using neural networks. *ARPN Journal of Engineering and Applied Sciences*, *12*(5).
18. Carmel, D., Yeshurun, A., & Moshe, Y. (2017). Detection of alarm sounds in noisy environments. In *25th European Signal Processing Conference (EUSIPCO)*, pp. 1839–1843.
19. Das, S. K., Samanta, S., Dey, N., & Kumar, R. (Eds.) (2020). *Design frameworks for wireless networks*. Springer, Lecture Notes in Networks and Systems.
20. Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040–53065.
21. Begum, A., Fatima, F., & Sabahath, A. (2019, April) Implementation of deep learning algorithm with perceptron using tenzorflow library. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0172–01752019.
22. Park, S., & Trevino, J. (2017, March). Automatic detection of emergency vehicles for hearing impaired drivers. *ARPN Journal of Engineering and Applied Sciences*, *12*(5).
23. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

24. Burden, F., & Winkler, D. (2008). Bayesian regularization of neural networks. *Methods in Molecular Biology*, *458*, 25–44.
25. Rane, D., Shirodkar, P., Panigrahi, T., & Mini, S. (2019, March). Detection of ambulance siren in traffic. In *IEEE International Conference on Wireless Communications Signal Processing and Networking (WiSPNET2019)*, Kalavakkam, Tamilinadu, India: IEEE.
26. Dey, N., Das, H., Naik, B., & Behera, H. S. (Eds.). (2019). *Design frameworks for wireless networks*. Academic Press.
27. Givargis, S., & Karimi, H. (2010). A basic neural traffic noise prediction model for Tehran's roads. *Journal of Environmental Management*, *91*(12), 2529–2534.
28. Nykaza, E. T., Boedihardjo, A. P., Blevins, M. G., Hulva, A. M., & Valente, D. (2015). Classification of environmental noise sources using machine-learning methods. *The Journal of the Acoustical Society of America*, *138*(3), 1731–1731.
29. Sparke, C. (2018). Environmental noise classification through machine learning. *Proceedings of ACOUSTICS*, *2018*, 1–9.
30. Chen, L. J., Nguyen, S., Trader, J. M., Moore, A., & Summers, J. E. (2019). Deep learning for underwater noise classification. *The Journal of the Acoustical Society of America*, *145*(3), 1920–1920.
31. Torija, A. J., Ruiz, D. P., & Ramos-Ridao, A. F. (2012). Use of back-propagation neural networks to predict both level and temporal-spectral composition of sound pressure in urban sound environments. *Building and Environment*, *52*, 45–56.
32. Torija, A. J., & Ruiz, D. P. (2015). A general procedure to generate models for urban environmental-noise pollution using feature selection and machine learning methods. *Science of The Total Environment*, *505*, 680–693.
33. Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access*, *7*, 19143–19165.
34. Kunaraj, K., Wenisch, S. M., Balaji, S., & Bosco. M. D. (2020). Impulse noise classification using machine learning classifier and robust statistical features. In S. Smys, J. M. R. S. Tavares, V. E. Balas, & A. M. Iliyasu (Eds.), *Computational Vision and Bio-Inspired Computing* (pp. 631–644). Springer International Publishing.
35. Chatterjee, S., Sarkar, S., Dey, N., Ashour, A. S., Sen, S., & Hassanien, A. E. (2017a). Application of cuckoo search in water quality prediction using artificial neural network. *International Journal of Computational Intelligence Studies*, *6*(2/3), 229–244.
36. Hore, S., Chatterjee, S., Sarkar, S., Dey, N., Ashour, A. S., Balas-Timar, D., & Balas, V. E. (2016). Neural-based prediction of structural failure of multistoried RC buildings. *Structural Engineering and Mechanics*, *58*(3), 459–473.
37. Chatterjee, S., Sarkar, S., Hore, S., Dey, N., Ashour, A. S., Shi, F., et al. (2017b). Structural failure classification for reinforced concrete buildings using trained neural network based multi-objective genetic algorithm. *Structural Engineering and Mechanics*, *63*(4), 429–438.
38. Das, S. K., & Tripathi, S. (2019). Energy efficient routing formation algorithm for hybrid ad-hoc network: A geometric programming approach. *Peer-to-Peer Networking and Applications*, *12*(4), 102–128.
39. Das, S. K., & Tripathi, S. (2018). Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming. *Applied Intelligence*, *48*(4), 1825–1845.
40. Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506.
41. Giannakopoulos, T., & Pikrakis, A. (2014). *Introduction to audio analysis: A MATLAB approach*. Elsevier Science.

Feature Selection Method Using CFO and Rough Sets for Medical Dataset

Ramesh Kumar Huda and Haider Banka

1 Introduction

In the last decade, the applications of DNA microarray increase rapidly in several areas but context with machine learning. In this dataset, a large number of the gene is operated by using a single experiment. Although it has several applications, but basically, DNA microarray is used for classification purposes. In general or traditional classifications, some problems are raised in the training and testing process due to nominal specimens and large dimensions of the gene pattern [1]. So, deriving an intelligent and efficient classification technique is very difficult. Therefore, feature selection is most crucial for the classification process.

In last few years, several authors proposed microarray-based works which are based on randomness in nature and used some bio-inspired and nature-inspired technique such as non-dominated sorting genetic algorithm (NSGA) [2], ant colony optimization (ACO) [3], particle swarm optimization (PSO) [4] and genetics algorithm (GA) [5], each stated method is evaluated with some classification methods and outperform the results. In these works, it is described that feature selection is used for analyzing a high dimensional dataset that helps to enhance learning methodology by reducing redundant information or attributes. The feature selection-based works are efficiently used two basic paradigms of feature selection such as (i) wrapper and (ii) filter. The first paradigm is used to analyze the accuracy of the algorithm with the help of goodness of the sub-parts of the dataset. The second paradigm is used to identify and analyze the basic characteristic of the dataset for training purposes.

R. K. Huda (✉) · H. Banka
Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand 826004, India
e-mail: ramesh.hudda@gmail.com

H. Banka
e-mail: hbanka2002@yahoo.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_4

RST stands for rough set theory, which is pioneered by Pawlak [6]. It is an extension of the fuzzy set theory, which is also used for reducing uncertainty or imprecise information efficiently and intelligently. The main key point for deriving this set theory is for enhancing the accuracy and reducing the issue in the areas of knowledge discovery and data mining. It used several purposed in terms of outperforming the metrics like decreasing unnecessary description information, analyzing and defining dependency among multiple attributes, evaluating decision rules, etc. In the large dataset, complexity is high, so it is difficult to analyze and predict minimal features which are most useful in term of solution. Hence, there is a need to apply several types of optimization or approximation algorithms for finding the local and global solution [7].

The main purpose of the microarray is to derive only the main feature or characteristic from the original dataset after preprocessing and convergence. Before preprocessing, the population is initialized with the help or random dataset. Then, apply searching-based optimization for collecting the main feature for the solution using a novel optimization technique which named as central force optimization (CFO). It is a multi-objective optimization technique to work as a stochastic in nature. In this paper, CFO is used with the fusion of rough set to collect and explore the optimal features from the large set of the dataset.

2 Preliminaries

This preliminaries section highlights the basic techniques named as CFO and rough set, which are used in this paper as a key point for handling uncertainty and collecting useful feature from the database.

2.1 Central Force Optimization

CFO is an intelligent-based meta-heuristic technique which is rapidly used in many areas [8–10]. Meta-heuristic technique is more powerful than heuristic technique or simple mathematical approach which is divided into two types as simple meta-heuristic and applied meta-heuristics [11, 12]. This technique is also known as nature-inspired technique which consists of several methods such as ant colony optimization [13, 14], particle swarm optimization [15, 16], African buffalo optimization [17], and fuzzy logic [18, 19]. This technique optimizes the parameters local as well as global way. It is also based on an evolutionary system [20]. It is based on a multidimensional technique, so it is also used as a multi-objective system that enhances the maximization system of the problem. It is fully deterministic due to its gravitational kinematics. The necessary steps of the CFO algorithms are as follows.

Step 1 Initialize the population
Step 2 Design main objective function as fitness function
Step 3 Assign acceleration
Step 4 Based on acceleration analyze and compute the new position
Step 5 Validate the new position based on decision space
Step 6 Update the fitness function for next iteration
Step 7 Calculate the next accelerations
Step 8 Check termination criteria, if reached then stop otherwise repeat the same steps.

The algorithm of CFO contains several mathematical modeling in term of optimization. Initially, it contains some data structures as shown in Eqs. (1)–(4), where R indicates position vectors and A indicates acceleration. Range of p, i, and j as $1 \le p \le N_P$, $1 \le i \le N_d$, $0 \le j \le N_t$. In Eq. (3), M is the fitness array. In Eq. (4), M_{best} is the best fitness function and N_{saved} is an array of last saving information. The range of q is $1 \le q \le N_{saved}$. After data structure declaration, initialization is required which shown as uniform probes on each coordinate axis, initial acceleration, and initial fitness function. Equations (5) and (6) show uniform probes on each coordinate axis. In Eq. (5), $n = 1$ to $\frac{N_p}{N_d}$:, and $p = n + \frac{(i-1)N_p}{N_d}$. Initial acceleration and initial fitness are shown in Eqs. (7) and (8), where values of p and i in Eq. (7) are $1 \le p \le N_p$, $1 \le \mathrm{i} \le N_d$ and for Eq. (8) are $1 \le p \le N_p$, $1 \le \mathrm{i} \le N_d$. The best fitness function is shown in Eq. (9). Time loop of the algorithm is varies $1 \le \mathrm{j} \le N_t$. This time loop is divided as new probe position, update fitness matrix, update accelerations, and increment. New probe position shown in Eq. (10), For $p = 1$ to N_p, i =1 to N_d. Update fitness matrix is shown in Eq. (11), for $p = 1$ to N_p, and update best fitness is M_{best}(s.). Update acceleration is shown in Eq. (12), for $p = 1$ to N_p, $i = 1$ to N_d, and j is increment as $j \to j + 1$, and repeat from (3)(A) until $j = N_t$ or other stopping criterion has been met.

$$R(p, i, j) \tag{1}$$

$$A(p, i, j) \tag{2}$$

$$M(p, j) = f(R(p, i, j)) \tag{3}$$

$$\text{Best\_fitness} = M_{best}(q) \tag{4}$$

$$\text{For } i = 1 \text{ } to \text{ } N_d \tag{5}$$

$$R(p, i, 0) = x_i^{min} + \frac{(n-1)(x_i^{max} - x_i^{min})}{\frac{N_p}{N_d} - 1} \tag{6}$$

$$A(p, i, 0) = 0 \tag{7}$$

$$M(p, 0) = f(R(p, i, 0)) \tag{8}$$

$$\text{Best Fitness} = \text{MAX}(M(p, 0),\ 1 \le p \le N_p) \tag{9}$$

$$R(p, i, j) = R(p, i, j-1) + \frac{1}{2}A(p, i, j-1)\Delta t^2 \tag{10}$$

$$M(p, j) = f(R(p, i, j)) \tag{11}$$

$$A(p, i, j) = G \sum_{k=1, k\neq p}^{N_p} U(M(k, j) - M(p, j)).(M(k, j) - M(p, j))^{\alpha}.\frac{R(k, i, j) - R(p, i, j))}{\left|\mathbf{R_j^k} - \mathbf{R_j^p}\right|^{\beta}} \tag{12}$$

2.2 *Rough Set Theory*

The rough set is an extension of the fuzzy set, which is also used to reduce the uncertainty of any information. Basically, it is used for data mining and a knowledge-based system for handling precise information efficiently. In this set theory, granularity is the main key element which is used to design indiscernibility relation. It is used to divide the field values. Finally, it helps to make efficient approximation within imprecise information by using lower and upper approximations [21]. In RST, for $N_1 \subseteq N$ and $P \subseteq E$, $\underline{P}\, N_1$ is the lower approximation which portrayed in Eq. (13), in which P according to N_1 if $[x]_P$ only contains instances in N_1. $\underline{P}\, N_1$ assess the number of instances that have been totally isolated from other classes instances. Equation (14) indicates objective function in PSORS. It shows depiction of the P based on target class in N. In the experiment that an feature subset having value of fitness (P) = 1, it implies that feature subset totally isolate each class from other classes.

$$\underline{P}N_1 = \{x \in N \mid [x]_P \subseteq N_1\} \tag{13}$$

$$\text{Fitness}(P) = \frac{\sum_{i=1}^{n} |\underline{P}N_i|}{|N|} \tag{14}$$

3 Proposed Algorithm

This is the main section of the paper which illustrates the main work in step by step that includes discussion about gene expression data preprocessing, design and analyze the fitness function based on constraints and parameters of the objective function.

3.1 Preprocessing of Gene Expression Data

The proposed method consists of two classes problem as (i) normal sample and (ii) diseased sample. The combining of both based on gene expression is initialization. It has more features with less sample. Among more features, most of the features is unnecessary redundant or not useful. This is a two-class problem because in this problem, one attribute, i.e., d contains two members whose value is defined as V_{d}. The proposed method contains a heuristic phase based on the thresholding system. The purpose of this phase is to reduce unnecessary and redundant information or attributes for collecting useful and required output to achieve the purpose of the goal. This phase uses CFO for determining sufficient and correct classification and make the goal decision accurately.

This preprocessing phase also leads to normalization that helps to scaling dataset and enabling the gene of the microarray in the experiment. The purpose of this normalization is to reduce ambiguous gene moderately neither high reduction nor low reduction. In this system, decision-maker first selects a minimal set of gene through the classes to achieve the most reliable classification. This system is based on Eq. (15) which is used for normalization of the attributes. In this equation, $\min_j$ and $\max_j$ indicate lowest and highest values of the gene for a_j attribute in all samples and value range is between 0 and 1. Equation (16) is used for position calculation of the kth partition where k varies 1 to 4. The thresholds values Th_i and Th_f are selected based on the idea given in [22]. Each pattern here is arranged in ascending order based on jth axis. The partition here is divided into some small classes that is also known as interval based on width δ and then counted the frequencies fr_c. In Eq. (16), cfr_{c-1} is a cumulative frequency of the preceding class interval like $cfr_{c-1} \leq R_k \leq cfr_c$, $R_k = \frac{N*k}{4}$ is a rank of the kth partition value, and l_c is a lower limit for the cth class interval. The used attributes are in the form of table which is convert as binary (i.e., 0 and 1) as: If $a^{'}(x) \leq Th_i$ then put "0" ElseIf $a^{'}(x) \geq Th_f$, then put "1" Else put "*" (i.e., do not care condition). After that, evaluate average of all occurrences of "*" for entry value in the table which is works as threshold Th_a. Finally, delete all attributes those having more "*" than the threshold value Th_a. It is known as updated or reduces attribute of the table A_r.

$$a_j^{'}(x_i) = \frac{a_j(x_i) - \min_j}{\max_j - \min_j}, \qquad \forall i \tag{15}$$

$$Th_k = l_c + \frac{R_k - cfr_{c-1}}{fr_c} * \delta \tag{16}$$

3.2 Fitness Function

This phase is used to evaluate the fitness function where the set of genes is represented as a string, and the nature of the string is binary. The length of the string is N that represents the total number of attributes, i.e., known as conditional attributes. The basic elements 0 and 1 indicate the absence and presence of the attributes. Basically, in this proposal, the fitness function is used for feature selection, which is the main objective of the proposal that is done by using the CFO technique. The main objective contains two sub-objectives like F_1 and F_2 where the first part indicates a number of 1's, i.e., number of features, and the second part indicates a description of v for the target in N for each iteration.

The gene expressions are represented by binary strings of length N, where N is the number of conditional attributes. In the bit representation, a "1" implies that the corresponding attribute is present while "0" means that it is not. The feature selection can be done by CFO algorithm using the following objective function is shown in Eq. 17. We have to use fitness function for feature selection, which contain two sub-functions (F_1, F_2). Where F_1 finds number of features (i.e., number of 1's), F_2 decides how well v describe each target in N. In this equation, values of $F_1(\mathbf{v})$ and $F_2(\mathbf{v})$ are $\frac{N-L_{\mathbf{v}}}{N}$ and $\frac{\sum_{i=1}^{n} |\underline{\mathrm{v}}N_i|}{|N|}$ where w_1 and w_2 are weighted factors lies between 0 and 1 and sum of two is always 1. The parameters v, $L_{\mathbf{v}}$, and $\underline{\mathrm{v}}\, N_1$ indicate subset of selected feature as shown in Eq. 18, number of 1's in element v and number of isolated elements. This process helps to decide the nature of v for N class.

$$\text{Fitness function}(F(\mathbf{v})) = w_1 F_1(\mathbf{v}) + w_2 F_2(\mathbf{v}) \tag{17}$$

$$\underline{\mathrm{v}}N_1 = \{x \in N \mid [x]_v \subseteq N_1\}. \tag{18}$$

The main aim of the proposed method to maximize the fitness function $f(\mathbf{v})$ with the help of RST and CFO techniques. It maximize the results in each and every step by optimizing its inherent parameters. The basic steps of the proposed method are shown in Fig. 1.

4 Experimental Result

4.1 Parameter Setting and Datasets

Dataset (colon cancer) is chosen from the UCI machine learning repository, which had 2000 genes (features), two classes(c1 and c2), and 62 samples (44 colon cancer and 22 normal)given in Table 1. The proposed method is used the KNN algorithm,

Step 1: Start
Step 2: Redundancy reduction is made for the high dimensional microarray data as shown in **Eq. (15)**, to get the reduced attribute value table Ar.
Step 3: Compute initial probe positions and corresponding objective fitness function for assigning initial accelerations.
Step 4: Successively compute each probe's new position based on previously computed accelerations.
Step 5: Verify that each probe is located inside the decision space, making corrections as required.
Step 6: Update fitness at each new probe position.
Step 7: Compute accelerations for the next time step using new positions.
Step 8: Repeat Steps (3)-(6) until over time steps or until a termination criterion.
Step 9: Stop

Fig. 1 Basic steps of the proposed method

Table 1 Usage details of the two-class microarray data

Data used	# Attributes	Classes	#Samples
Colon	2000	Colon cancer	40
		Normal	22

which stands for K-nearest neighbor. This is used uses as a classifier in the experiment where values of k varies 1, 3, and 5. The experiment is done in two stages first is training and the second is testing. Each stage is contributed as 50% ratio. In first stage, 31 samples are used as 20 and 11, in the second stage also 31 samples are used as 20 and 11.

4.2 Results and Discussion

Table 2 represents the output of the preprocessing phase in which features reduce to 1102 and then this reduced subset of feature apply in the proposed algorithm (CFORS) to get relevant feature subset.

4.2.1 Result of Existing Method (GA)

According to Table 3, GA selected small feature subset (15 out of 1102) and achieved classification accuracy 71, when $k = 1$, 58.1, when $k = 3$ and 48.1, when $k = 5$. Note: The above-mentioned classification accuracy is average (Avg. of class1 and class2).

Table 2 Details of the two-class microarray data after preprocessing

Dataset	Reduced features#	Classes	Samples		
			Total	Train	Test
Colon	1102	Colon cancer	40	20	20
		Normal	22	11	11

4.2.2 Result of Proposed Method (CFORS)

As per Table 3, the proposed algorithm was selected very small feature subset (5 out of 1102) and achieved better classification accuracy than all features, i.e., classification accuracy 71.2, when $k = 1$, 64.5, when $k = 3$, and 64.5, when $k = 5$.

4.2.3 Comparison of CFORS and GA

According to Table 3, the proposed algorithm (CFORS) selects reduced feature subset and achieved significantly higher classification accuracy than existing algorithm (GA), i.e., in case of CFORS, the size of selected feature subset is 5 and classification accuracy is 71.2, and in case of GA, the size of selected feature subset is 15 and classification accuracy is 71 when $K = 1$.

Table 4 shows that if the value of w_1 in fitness function increases (w_1 =0.6) than the classification accuracy also increases and size of feature subset remain same. Similarly, the value of w_1 = 0.7 than the classification accuracy also increases and size of feature subset remain same.

Table 3 Compare the performance using KNN classifier

Algo.	Reducts	K-nearest neighbors classification (%) on test set								
		$k = 1$			$k = 3$			$k = 5$		
		C_1	C_2	Net	C_1	C_2	Net	C_1	C_2	Net
CFORS	5	85.0	45.5	71.2	80.0	36.4	64.5	80.0	36.4	64.5
GA	15	75.0	63.6	71.0	70	36.4	58.1	75.0	0.0	48.4

Table 4 Results when value of $w_1 = 0.6$ and 0.7 in fitness function

Algo.	Reducts	K-nearest neighbors classification (%) on test set								
		$k = 1$			$k = 3$			$k = 5$		
		C_1	C_2	Net	C_1	C_2	Net	C_1	C_2	Net
CFORS, 0.6	5	85.0	46	71.4	81.1	36.6	64.8	80.2	36.5	64.7
CFORS,0.75	5	85.1	46.1	71.4	81.2	36.7	64.9	80.3	36.6	64.7

5 Conclusion

The proposed method is a hybridization technique which is a fusion of CFO and RST. The combination of both makes the proposed algorithm is more efficient and intelligent in terms of gene selection and classification. It outperforms the existing feature selection algorithm GA based on the cardinality of feature and classification accuracy. It uses KNN classifier to identify potentially good gene subsets. Basically, CFO is deterministic in nature as compared to existing techniques which are stochastic in nature. Finally, it is observed that classification accuracy is increased simultaneously based on the increasing weight of the classification quality of the feature selection in the fitness function.

References

1. Sung-Bae, C. H. O., & Hong-Hee, W. O. N. (2003). Data mining for gene expression profiles from dna microarray. *International Journal of Software Engineering and Knowledge Engineering*, *13*(06), 593–608.
2. Banerjee, M., Mitra, S., & Banka, H. (2007, July). Evolutionary rough feature selection in gene expression data. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *37*(4), 622–632.
3. Xiong, W., & Wang, C. (2009. August). A hybrid improved ant colony optimization and random forests feature selection method for microarray data, pp. 559–563.
4. Xue, B., Zhang, M., & Browne, W. N. (2013, December). Particle swarm optimization for feature selection in classification: A multi-objective approach. *Cybernetics, IEEE Transactions on*, *43*(6), 1656–1671.
5. AlSukker, A., Khushaba, R. N., & Al-Ani, A. (2010, October). Enhancing the diversity of genetic algorithm for improved feature selection, pp. 1325–1331.
6. Pawlak, Z. (2002, March). Rough set theory and its applications. *Journal of Telecommunications And Information Technology*.
7. Wang, X., Xia, W., Jensen, R., Teng, X., & Yang, J. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, *28*, 459–471.
8. Liu, Y., & Tian, P. (2015). A multi-start central force optimization for global optimization. *Applied Soft Computing*, *27*, 92–98.
9. Das, S. K., Samanta, S., Dey, N., & Kumar, R. (2020). *Design frameworks for wireless networks*. Springer.
10. De, D., Mukherjee, A., Das, S. K., & Dey, N. (2020). Nature inspired computing for wireless sensor networks.
11. Dey, N. (2017). *Advancements in applied metaheuristic computing*. IGI Global.
12. Dey, N., Ashour, A., & Bhattacharyya, S. (2020). *Applied nature-inspired computing: algorithms and case studies*. Springer.
13. Bouamama, S., Blum, C., & Fages, J.-G. (2019). An algorithm based on ant colony optimization for the minimum connected dominating set problem. *Applied Soft Computing*, *80*, 672–686.
14. Omran, Mahamed G. H., & Al-Sharhan, S. (2019). Improved continuous ant colony optimization algorithms for real-world engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *85*, 818–829.
15. Chatterjee, S., Hore, S., Dey, N., Chakraborty, S., & Ashour, A. S. (2017). Dengue fever classification using gene expression data: a pso based artificial neural network approach. In: *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*. Springer, pp. 331–341.

16. Jagatheesan, K., Anand, B., Samanta, S., Dey, N., Ashour, A. S., & Balas, V. E. (2017). Particle swarm optimisation-based parameters optimisation of pid controller for load frequency control of multi-area reheat thermal power systems. *International Journal of Advanced Intelligence Paradigms*, *9*(5–6), 464–489.
17. Bera, S., Das, S. K., & Karati, A. (2020). Intelligent routing in wireless sensor network based on african buffalo optimization. In: *Nature Inspired Computing for Wireless Sensor Networks* (pp. 119–142). Springer.
18. Das, S. K., & Tripathi, S. (2018). Intelligent energy-aware efficient routing for manet. *Wireless Networks*, *24*(4), 1139–1159.
19. Das, S. K., & Tripathi, S. (2017). Energy efficient routing formation technique for hybrid ad hoc network using fusion of artificial intelligence techniques. *International Journal of Communication Systems*, *30*(16), e3340.
20. Formato, R. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. *OPSEARCH*, *46*, 25–51.
21. Kumar Huda, R., & Banka, H. (2019). Efficient feature selection and classification algorithm based on pso and rough sets. *Neural Computing and Applications*, *31*(8), 4287–4303.
22. Mitra, S., & Acharya, T. (2003). *Data mining: Multimedia, soft computing and bioinformatics*. New York: Wiley.

Fuzzy-Based Optimal Solution for Minimization of Loss of Company Based on Uncertain Environment

Manoj Kumar Mandal, B. K. Mahatha, Arun Prasad Burnwal, Santosh Kumar Das, and Aditya Sharma

1 Introduction

In modern era, there are several types of products available for use of the customers or users. The main reason of this choice variations of the users or customers. There are different users available in our country for purchasing several products based on variety of strategies such as free item of "X", if you buy item "Y", gift voucher, free delivery, discount. These varieties help to make flexible choice to the customers. It creates several types of uncertainties in purchasing and selling to the customer. So, to make the exact decision is difficult due to imprecise information in the maker scenario of the company. Therefore, day-by-day profit of the company decreases, and losses increase. Hence, in this paper, a method is proposed for minimizing the loss of company using fusion of quadratic programming and fuzzy logic. Quadratic

M. K. Mandal (✉)
Department of Mathematics, Jharkhand Rai University, Ranchi 835222, India
e-mail: mkmandal346@gmail.com

B. K. Mahatha
Amity School of Engineering and Technology, Amity University Jharkhand, Ranchi 834001, India
e-mail: bhupeshmahatha@gmail.com

A. P. Burnwal
Department of Mathematics, GGSESTC, Bokaro, Jharkhand 827013, India
e-mail: apburnwal@yahoo.com

S. K. Das
School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Institute Park, Pallur Hills, Berhampur, Odisha 761008, India
e-mail: sunsantosh2014@gmail.com

A. Sharma
Institute of Nanoengineering and Microsystems, National Tsing Hua University, No. 101, Sec. 2, Guang Fu Road, Hsinchu 30013, Taiwan, ROC
e-mail: saditya56399@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_5

programming is used to model the main objective and its related constraints in the form of nonlinear. In this model, decision variables are in the form of square. Fuzzy logic is used to reduce the imprecise information efficiently. The combination of both quadratic programming and fuzzy logic helps to model the main goal of the paper.

The roadmap of the paper is as follows. Unit 2 described some information about the existing works. Unit 3 describes the details of the proposed method. Unit 4 describes the simulation analysis, and Unit 5 concludes the paper.

2 Related Works

In the last few decades, several works are proposed based on mathematical optimization for linear or quadratic or fuzzy logic with the context of soft computing and metaheuristic optimization or traditional optimization. Some of the works are described in this section [1–4]. Rodias [5] designed a combined technique for fertilizer application using linear programming. The proposed method used linear programming for optimizing several strategies of fertilizer such as harvesting handling operation, organic fertilization, mineral, and other several usages. These all strategies are mapped into linear programming for solving different issues of the system. Ji et al. [6] designed a multi-objective optimization technique for linear programming using a game method for supply chain management. The proposed method is the fusion of two methods; the first is duality method for mathematical approach and the second is Karush–Kuhn–Tucker method. In this proposed method, the author tried Pareto equilibrium problem to achieve the purpose of the method. Finally, it achieves the purpose of the method by using multi-linear technique. Laso et al. [7] proposed an application for life cycle management for food waste. The proposed method is based on linear programming technique. The proposed method is based on some strategies such as climate, food, energy, and nexus of water. In this model, nexus of water is defined with the help of linear programming technique for assessment of life cycle. Finally, it helps to minimize the consumption of natural resources. Gharanjik et al. [8] designed max–min technique for application of information and signal processing. The proposed method is modelled by quadratic optimization technique. The proposed method uses Gram–Schmidt technique for handling approximation method to achieve the discrete requirement. Finally, it achieves the goal of the proposed method and reduces the computational cost of the method with respect to real-life scenarios. Hempel et al. [9] proposed a method for hybrid control system using quadratic optimization technique. In this method, the used quadratic programming is mixed integer programming. In this method, the main issue is to manage piecewise affine system. This issue is to efficiently control and manage with the help of mixed integer quadratic optimization technique. In [10], SB has considered for the student academic performance prediction. In this paper, student academic performance prediction is evaluated with the help of different machine learning classification models. Further accuracy has been improved using ensembling methods. Outliers are points that do not follow the patterns from the rest of the data. The clustering-based outlier detection method has been proposed in the [11]; in this method, three nearest $K - 1$, K, $K + 1$ clusters computed using the K-means clustering algorithm. Farthest point from all the clusters is considered as an outlier point. Sales forecasting is an essential facet for the industries associated with sales, wholesale, manufacturing, etc., all around the globe. It is important with respect to resource allocation, revenue estimation, and market strategy planning. A two-level approach shown- [12] performs better in comparison with other single-level model. Jat et al. [13] designed an intelligent technique for QoS in WLAN. This proposal is based on video delivery system. This is based on multimedia application for video data processing and analysing. The data is analysed here based on real-time data that is generated by the Internet. It also helps in video data transmission, storage, evaluating, and broadcasting. In WSN, data is gathered from multiple homogeneous or heterogeneous sources because real-life data is connected with different IoT, IoV, or cloud environment. So, it is difficult to keep the natures of the data in same structure. Information retrieval [14] is a very important part in modern research areas which indicates to collect information that are stored in unstructured form based on multiple local languages and process it in particular pattern after observing. Hao et al. [15] designed an evaluation system for big data analysis. This data is based on IoV where it means Internet of vehicle. This proposal is based on K-means algorithm that is used here as a clustering. In this work, different behaviours of the driving are involved for controlling vehicle. Finally, it helps in reducing fuel consumption and helps in transportation globally. In the last few years, several optimization and fuzzy logic-based works are proposed [16, 17]; some of them are as Tripathi and Das [18] proposed five input parameters based on intelligence routing using multiple criteria of ad hoc network. This is based on soft set method which is mixed by extended fuzzy set, i.e. intuitionistic fuzzy set and two techniques of the multi-criteria decision system. Each input parameter is mapped into the soft set in terms of three elements such true membership value, false membership value, and between both which is known as hesitation membership value. Finally, it helps to resolve the uncertainty of the network efficient and derive optimal route of the network. Das et al. [19] proposed a detailed survey for soft computing technique based on its different inherent paradigms such as fuzzy logic, genetic algorithm, and neural network apart of these also contain some other method based on soft computing results. It guides the user or reader as well as new researchers about basic concepts of soft computing and different elements and their usages. Tripathi and Das [20] proposed a vague set-based routing technique for ad hoc network. Vague set is one of the extended versions of the fuzzy set where fuzzy set deals with the degree of membership value and vague set deals with degree of membership and degree of non-membership value. Later, this work is extended in [21] for evaluating more network metrics. The combination of both helps to recognize the imprecise network parameters efficiently; especially, energy and distance both are the crucial parameters of the network. Finally, it helps to enhance the network metrics and network lifetime. Das et al. [22] designed a routing method for multiple destination ad hoc network where source is one, but destination node is situated in various forms based on different energy systems. In this work, fuzzy logic is used in the form of linguistic variable that divides into some membership function based on randomization which helps to categorize the feasible as well as optimal route and reduce the uncertainty of the network. Mishra et al. [23] proposed a model for grinding process based on fuzzy logic. The proposed method is based on an intelligent operation which is named as compensatory operator. It is based on weighted factoring technique. The proposed method basically optimized several processes such as speed, density, and sectional area which is cross; in this model, all these parameters are imprecise which are compact and model by membership function of the fuzzy logic. Murmu et al. [24] designed a system for predicting and analysing surface roughness. The proposed method is based on hard face component. In this model, fuzzy logic is used to optimize several factors of the machines efficiently and manage various parts of the machine. It uses hard surfacing technique that uses fuzzy logic, and combine system enhances the service mechanism of the machine. Kumari and Burnwal [25] designed an interactive model for inventory control system. The proposed method is based on various mathematical operators for analysing different scenarios of the model. It uses fuzzy logic system for enhancing the model by reducing imprecise parameters of the network. Finally, it solves several objectives of the inventory by combining multi-objective optimization and fuzzy logic of the system.

3 Proposed Method

In the modern era, there are several customers involved in different companies with the context of some uncertainties that relate in selling and purchasing. The reason behind this is because of different choices of the customers that varies from one customer to another customer. It reflects based on the mentality of the customers. The proposed method is based on minimization of loss of a company based on variation of its strategy. The strategy indicates different policies that company is applying to purchase or selling the products. It may be any policy such as discount, get one free one, and coupon system. It varies from company to company. So, today's market is too difficult for any marketing system. The proposed method is based on the fusion of fuzzy logic and quadratic programming. Fuzzy logic is a part of soft computing which works between partial truths and partial false, and its main element is degree of membership value that indicates that degree of truth value depends on the choices of the customers. The quadratic programming is the extended form of linear programming. It is more powerful than linear programming which works based on degree of two. Degree indicates power value of the decision variable that helps to estimate the objective value with its constraints to meet the objective value of the goal. Table 1 shows membership functions of different strategies of the company. In this proposal, three strategies are considered as S_L, S_M, and S_H where "Low", "Medium", and "High" indicate different linguistic variables of the fuzzy logic for reducing the uncertainty of the company. Equations 1–5 indicate mathematical models of the proposed method.

Table 1 Membership functions of strategy

Linguistic variable	Notation	Range
Low	S_L	(0–40%)
Medium	S_M	(25–70%)
High	S_H	(60–100%)

$$\begin{aligned} &\text{Minimize :} && \text{Obj1} = (o_1)^2 + (o_2)^2 \\ &\text{Subject to constraints :} && s_1 o_1 + s_1 o_2 \geq 100,000 \\ &&& s_2 o_1 + s_2 o_2 \geq 100,000 \\ &&& s_3 o_1 + s_3 o_2 \geq 100,000 \end{aligned} \tag{1}$$

where s_1 is the (0–40%) of 100,000, s_2 is the (25–70%) of 100,000, s_3 is the (60–100%) of 100,000.

$$\begin{aligned} &\text{Minimize :} && \text{Obj2} = (o_1)^2 + (o_2)^2 \\ &\text{Subject to constraints :} && s_1 o_1 + s_1 o_2 \geq 200,000 \\ &&& s_2 o_1 + s_2 o_2 \geq 200,000 \\ &&& s_3 o_1 + s_3 o_2 \geq 200,000 \end{aligned} \tag{2}$$

where s_1 is the (0–40%) of 200,000, s_2 is the (25–70%) of 200,000, s_3 is the (60–100%) of 200,000.

$$\begin{aligned} &\text{Minimize :} && \text{Obj3} = (o_1)^2 + (o_2)^2 \\ &\text{Subject to constraints :} && s_1 o_1 + s_1 o_2 \geq 300,000 \\ &&& s_2 o_1 + s_2 o_2 \geq 300,000 \\ &&& s_3 o_1 + s_3 o_2 \geq 300,000 \end{aligned} \tag{3}$$

where s_1 is the (0–40%) of 300,000, s_2 is the (25–70%) of 300,000, s_3 is the (60–100%) of 300,000.

$$\begin{aligned} &\text{Minimize :} && \text{Obj4} = (o_1)^2 + (o_2)^2 \\ &\text{Subject to constraints :} && s_1 o_1 + s_1 o_2 \geq 400,000 \\ &&& s_2 o_1 + s_2 o_2 \geq 400,000 \\ &&& s_3 o_1 + s_3 o_2 \geq 400,000 \end{aligned} \tag{4}$$

where s_1 is the (0–40%) of 400,000, s_2 is the (25–70%) of 400,000, s_3 is the (60–100%) of 400,000.

$$\begin{aligned} &\text{Minimize :} && \text{Obj5} = (o_1)^2 + (o_2)^2 \\ &\text{Subject to constraints :} && s_1 o_1 + s_1 o_2 \geq 500,000 \\ &&& s_2 o_1 + s_2 o_2 \geq 500,000 \\ &&& s_3 o_1 + s_3 o_2 \geq 500,000 \end{aligned} \tag{5}$$

where s_1 is the (0–40%) of 500,000, s_2 is the (25–70%) of 500,000, s_3 is the (60–100%) of 500,000.

In the proposed method, there are two type of productions which are considered as type-1 and type-2. So, loss of the company is also two types based on the types. Hence, loss of type-1 is considered as o_1 and loss of type-2 is considered as o_2. The combination of both is $o_1 + o_2$ which is in the form of quadratic mathematical modelling is $(o_1)^2 + (o_2)^2$. The purpose of this model is to minimize based on variant strategies with context of five iterations or rounds as investments of the company are 100,000, 200,000, 300,000, 400,000, and 500,000. Finally, it is observed that as investment increases, then minimization value also increases.

4 Simulation and Analysis

The proposed method is simulated and verified in LINGO optimization software which is used to optimize nonlinear objectives and constraints of the proposed method. In this paper, total nonlinear objective functions are five with linear constraints. Each objective function contains three linear constraints. So, here, total constraints are 5×3, i.e. 15. Each mathematical model consists of three fuzzy linguistic variables for reducing uncertainty of the system such as “Low”, “Medium”, and “High”. The initial investment of the model is 100,000 and maximum investment of the model is 500,000. The investment of the system increases by 100,000 for each iteration. The total iteration used in this proposed method is 3. Hence, simulation parameters and its descriptions are shown in Table 2.

Figures 1, 2, 3, 4, and 5 show results of the proposed method with the context of five mathematical models based on fusion of quadratic programming and fuzzy logic. Simulation and analysis show that when investment of the company increases, then minimization value also increases for each iteration. In iteration 1, the minimization value is 20.00000 for loss of category 1 which is 2.00015 and loss of category 2 which is 3.999993 with selecting strategies for s_1 is (10, 20%), s_2 is (30, 50%), and

Table 2 Simulation parameters

Parameter	Description
Software	LINGO
Nonlinear objective function	5
Strategy of the company	3
Constraints	5 × 3
Number of iterations	5
Number of linguistic variable	3
Name of the linguistic variables	Low, Medium, High
Initial investment	100,000
Maximum investment	500,000

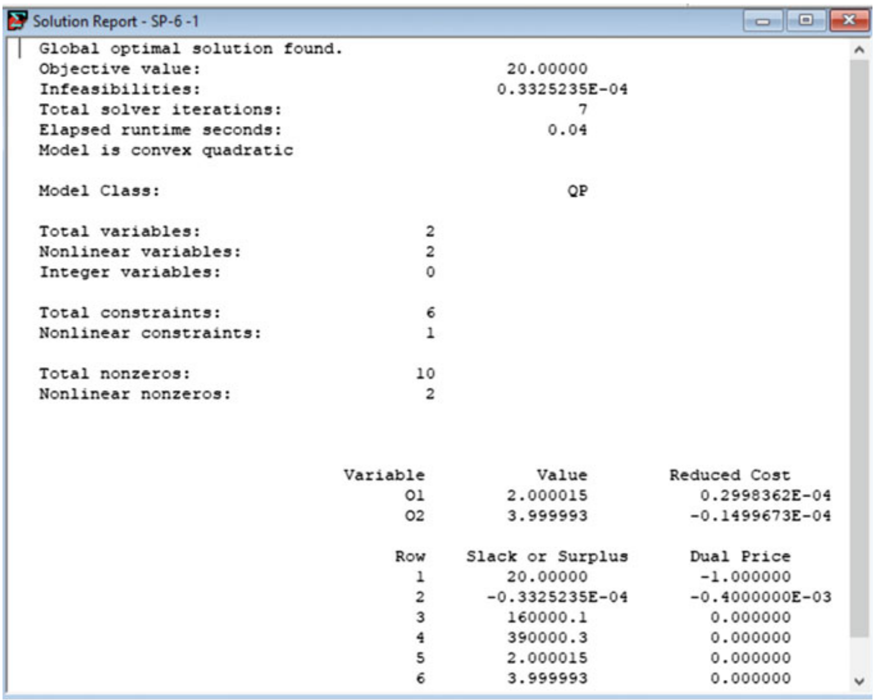
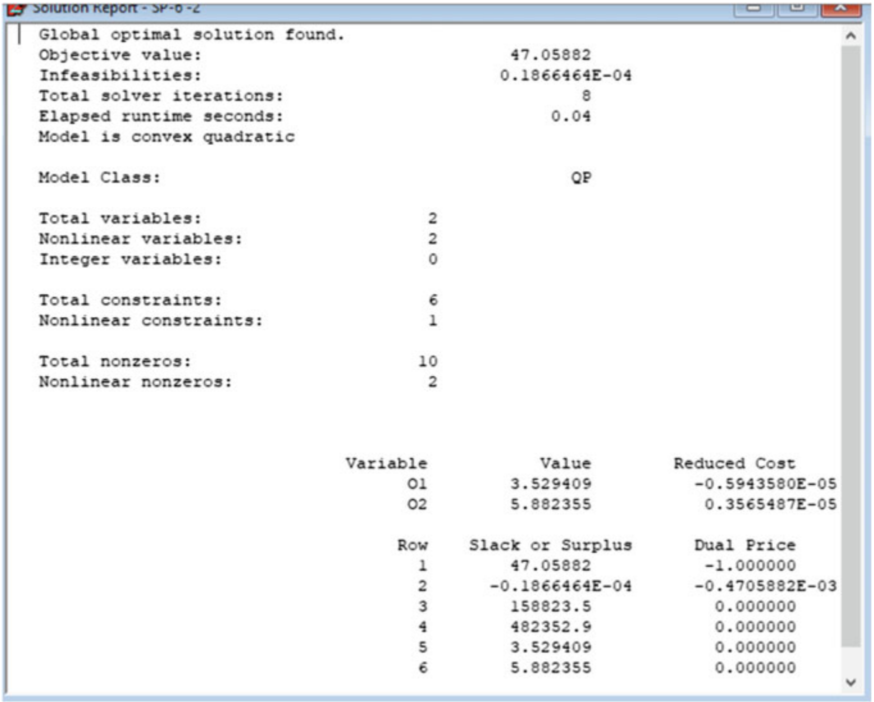

```
Solution Report - SP-6 -1
Global optimal solution found.
Objective value:                              20.00000
Infeasibilities:                            0.3325235E-04
Total solver iterations:                             7
Elapsed runtime seconds:                          0.04
Model is convex quadratic

Model Class:                                        QP

Total variables:                    2
Nonlinear variables:                2
Integer variables:                  0

Total constraints:                  6
Nonlinear constraints:              1

Total nonzeros:                    10
Nonlinear nonzeros:                 2

                       Variable           Value        Reduced Cost
                             O1        2.000015        0.2998362E-04
                             O2        3.999993       -0.1499673E-04

                            Row    Slack or Surplus      Dual Price
                              1        20.00000         -1.000000
                              2     -0.3325235E-04     -0.4000000E-03
                              3        160000.1          0.000000
                              4        390000.3          0.000000
                              5        2.000015          0.000000
                              6        3.999993          0.000000
```

Fig. 1 Minimization of company loss in iteration 1 for investment 100,000

s_3 is (65, 90%). Each strategy is divided into two parts for loss of category 1, i.e. o_1, and loss of category 2 i.e. o_2. In iteration 2, the minimization value is 47.05882 for loss of category 1 which is 3.529409 and loss of category 2 which is 5.882355 with selecting strategies for s_1 is (15, 25%), s_2 is (35, 40%), and s_3 is (60, 80%). In iteration 3, the minimization value is 105.8824 for loss of category 1 which is 8.823512 and loss of category 2 which is 5.294147 with selecting strategies for s_1 is



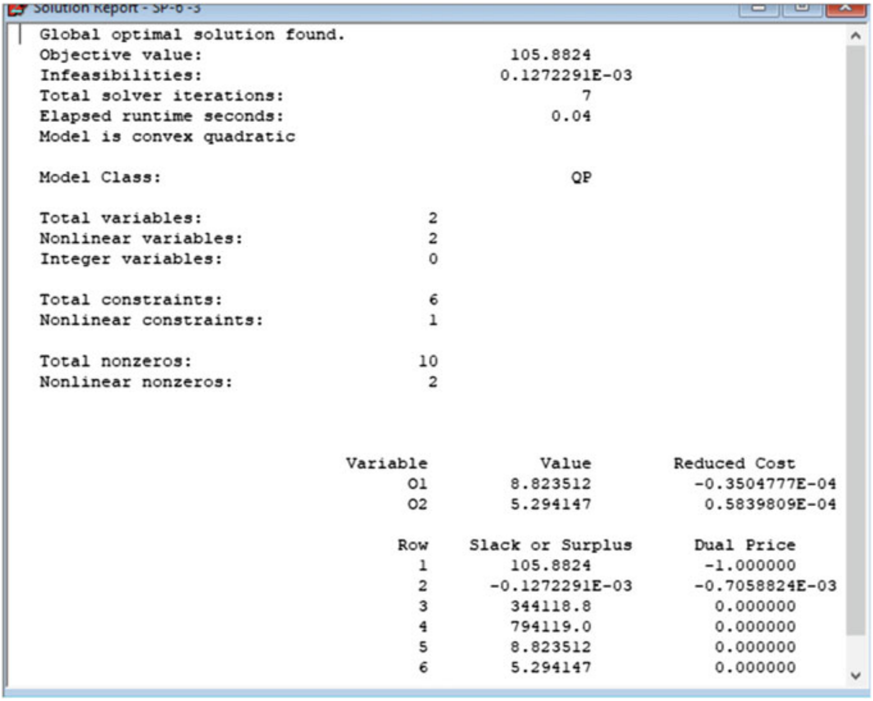
```
Solution Report - SP-6 -2
Global optimal solution found.
Objective value:                              47.05882
Infeasibilities:                              0.1866464E-04
Total solver iterations:                             8
Elapsed runtime seconds:                          0.04
Model is convex quadratic

Model Class:                                        QP

Total variables:                      2
Nonlinear variables:                  2
Integer variables:                    0

Total constraints:                    6
Nonlinear constraints:                1

Total nonzeros:                      10
Nonlinear nonzeros:                   2


                          Variable          Value        Reduced Cost
                                O1       3.529409       -0.5943580E-05
                                O2       5.882355        0.3565487E-05

                               Row    Slack or Surplus      Dual Price
                                 1       47.05882          -1.000000
                                 2      -0.1866464E-04     -0.4705882E-03
                                 3       158823.5           0.000000
                                 4       482352.9           0.000000
                                 5       3.529409           0.000000
                                 6       5.882355           0.000000
```

Fig. 2 Minimization of company loss in iteration 2 for investment 200,000

(25, 15%), s_2 is (37, 60%), and s_3 is (70, 90%). In iteration 4, the minimization value is 160.0000 for loss of category 1 which is 12.00015 and loss of category 2 which is 3.9999548 with selecting strategies for s_1 is (30, 10%), s_2 is (35, 50%), and s_3 is (95, 65%). In iteration 5, the minimization value is 243.9024 for loss of category 1 which is 9.756091 and loss of category 2 which is 12.19513 with selecting strategies for s_1 is (20, 25%), s_2 is (35, 65%), and s_3 is (65, 95%). And the summarized details are shown in Table 3.

5 Conclusions

In this paper, an efficient method is proposed for uncertain environment of the company for minimizing loss. Company strategy varies based on choices of the customer where strategy of the company is fuzzy instead of rigid, because goal of



Solution Report - SP-6 -3

```
Global optimal solution found.
Objective value:                              105.8824
Infeasibilities:                            0.1272291E-03
Total solver iterations:                             7
Elapsed runtime seconds:                          0.04
Model is convex quadratic

Model Class:                                        QP

Total variables:                    2
Nonlinear variables:                2
Integer variables:                  0

Total constraints:                  6
Nonlinear constraints:              1

Total nonzeros:                    10
Nonlinear nonzeros:                 2



                         Variable           Value        Reduced Cost
                               O1        8.823512       -0.3504777E-04
                               O2        5.294147        0.5839809E-04

                              Row    Slack or Surplus      Dual Price
                                1        105.8824          -1.000000
                                2     -0.1272291E-03     -0.7058824E-03
                                3        344118.8           0.000000
                                4        794119.0           0.000000
                                5        8.823512           0.000000
                                6        5.294147           0.000000
```

Fig. 3 Minimization of company loss in iteration 3 for investment 300,000

the company is to reduce maximum loss of the company by estimating imprecise choices. So, in this paper based of fusion of quadratic programming and fuzzy logic, loss of the company is efficiently minimized in each iteration where investment of the company varies by 100,000. It is observed that when investment increases, then loss value also increases with variation of fuzzy linguistic variable and losses of the company.

```
Solution Report - SP-6 -4

Global optimal solution found.
Objective value:                              160.0000
Infeasibilities:                            0.9461895E-03
Total solver iterations:                             7
Elapsed runtime seconds:                          0.04
Model is convex quadratic

Model Class:                                        QP

Total variables:                    2
Nonlinear variables:                2
Integer variables:                  0

Total constraints:                  6
Nonlinear constraints:              1

Total nonzeros:                    10
Nonlinear nonzeros:                 2


                           Variable           Value        Reduced Cost
                                 O1        12.00015        0.3010264E-03
                                 O2        3.999548       -0.9032509E-03

                                Row    Slack or Surplus      Dual Price
                                  1        160.0000         -1.000000
                                  2     -0.9461895E-03      -0.8000000E-03
                                  3        219982.7          0.000000
                                  4        999984.9          0.000000
                                  5        12.00015          0.000000
                                  6        3.999548          0.000000
```

Fig. 4 Minimization of company loss in iteration 4 for investment 400,000

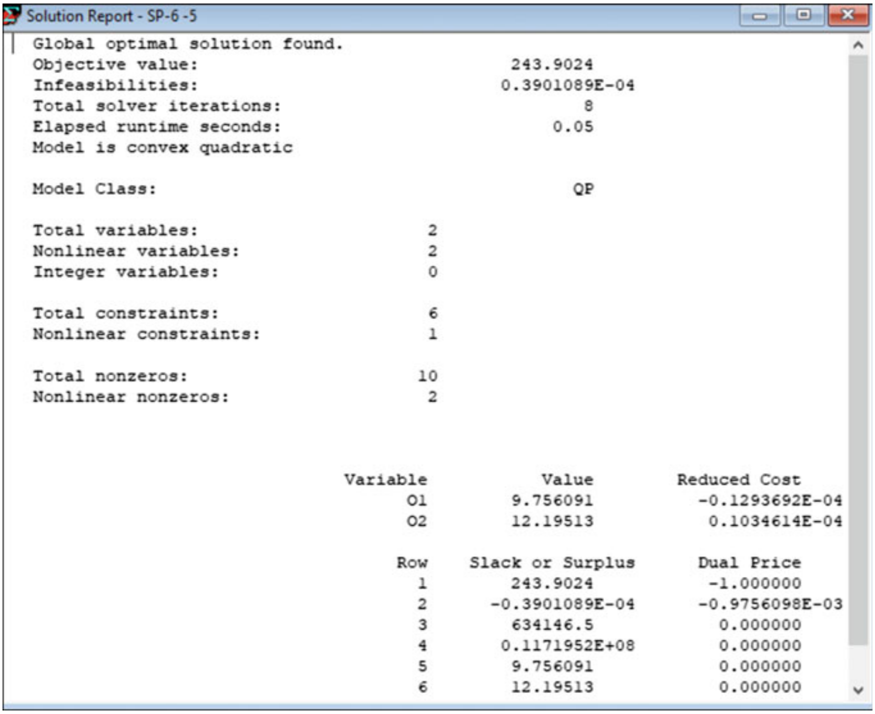


Fig. 5 Minimization of company loss in iteration 5 for investment 500,000

Table 3 Summarized details of the proposed mathematical model

Objective value	Loss of category 1	Loss of category 2	Strategy 1 (%)	Strategy 2 (%)	Strategy 3 (%)
Under investment 100,000					
20.00000	2.00015	3.999993	10, 20	30, 50	65, 90
Under investment 200,000					
47.05882	3.529409	5.882355	15, 25	35, 40	60, 80
Under investment 300,000					
105.8824	8.823512	5.294147	25, 15	37, 60	70, 90
Under investment 400,000					
160.0000	12.00015	3.9999548	30, 10	35, 50	95, 65
Under investment 500,000					
243.9024	9.756091	12.19513	20, 25	35, 65	65, 95

References

1. So-In, C., Permpol, S., & Rujirakul, K. (2016). Soft computing-based localizations in wireless sensor networks. *Pervasive and Mobile Computing, 29,* 17–37.
2. Binh, H. T. T., & Nam, N. H. (2018). Introduction to coverage optimization in wireless sensor networks. In *Soft Computing in Wireless Sensor Networks* (pp. 115–136). Chapman and Hall/CRC.
3. Yun, S., Lee, J., Chung, W., Kim, E., & Kim, S. (2009). A soft computing approach to localization in wireless sensor networks. *Expert Systems with Applications, 36*(4), 7552–7561.
4. Kaur, S., & Mahajan, R. (2018). Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egyptian Informatics Journal, 19*(3), 145–150.
5. Rodias, E. C., Sopegno, A., Berruto, R., Bochtis, D. D., Cavallo, E., & Busato, P. (2019). A combined simulation and linear programming method for scheduling organic fertiliser application. *Biosystems Engineering, 178,* 233–243.
6. Ji, Y., Li, M., & Qu, S. (2018). Multi-objective linear programming games and applications in supply chain competition. *Future Generation Computer Systems, 86,* 591–597.
7. Laso, J., Margallo, M., García-Herrero, I., Fullana, P., Bala, A., Gazulla, C., & Aldaco, R. (2018). Combined application of life cycle assessment and linear programming to evaluate food waste-to-food strategies: Seeking for answers in the nexus approach. *Waste Management, 80,* 186–197.
8. Gharanjik, A., Soltanalian, M., Shankar, M. B., & Ottersten, B. (2019). Grab-n-Pull: A max-min fractional quadratic programming framework with applications in signal and information processing. *Signal Processing, 160,* 1–12.
9. Hempel, A. B., Goulart, P., & Lygeros, J. (2012). Inverse parametric quadratic programming and an application to hybrid control. *IFAC Proceedings Volumes, 45*(17), 68–73.
10. Kumari, P., Jain, P. K., & Pamula, R. (2018, March). An efficient use of ensemble methods to predict students academic performance. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)* (pp. 1–6) IEEE.
11. Mishra, G., Agarwal, S., Jain, P. K., & Pamula, R. (2019). Outlier detection using subset formation of clustering based method. *International Conference on Advanced Computing Networking and Informatics* (pp. 521–528). Singapore: Springer.
12. Punam, K., Pamula, R., & Jain, P. K. (2018, September). A two-level statistical model for big mart sales prediction. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 617–620). IEEE.
13. Jat, D. S., Bishnoi, L. C., & Nambahu, S. (2018). An Intelligent wireless QoS technology for big data video delivery in WLAN. *International Journal of Ambient Computing and Intelligence (IJACI), 9*(4), 1–14.
14. Rasheed, I., & Banka, H. (2018, March). Query expansion in information retrieval for urdu language. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)* (pp. 1–6). IEEE.
15. Hao, R., Yang, H., & Zhou, Z. (2019). Driving behavior evaluation model base on big data from internet of vehicles. *International Journal of Ambient Computing and Intelligence (IJACI), 10*(4), 78–95.
16. Das, S. K., Samanta, S., Dey, N., & Kumar, R. (2020). *Design frameworks for wireless networks.* Springer.
17. De, D., Mukherjee, A., Das, S. K., & Dey, N. (2020). *Nature inspired computing for wireless sensor networks.* Springer
18. Das, S. K., & Tripathi, S. (2018). Intelligent energy-aware efficient routing for MANET. *Wireless Networks, 24*(4), 1139–1159.
19. Das, S. K., Kumar, A., Das, B., & Burnwal, A. P. (2013). On soft computing techniques in various areas. *Computer Science and Information Technology, 3,* 59.
20. Das, S. K., & Tripathi, S. (2015). Energy efficient routing protocol for manet based on vague set measurement technique. *Procedia Computer Science, 58,* 348–355.

21. Das, S. K., & Tripathi, S. (2016). Energy efficient routing protocol for MANET using vague set. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving* (pp. 235–245). Singapore: Springer.
22. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015, February). Fuzzy based energy efficient multicast routing for ad-hoc network. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (pp. 1–5). IEEE.
23. Mishra, B. K., Yadav, B., Jha, S. K., & Burnwal, A. P. (2015). Fuzzy set theory approach to model super abrasive grinding process using weighted compensatory operator. *International Journal of Research in Computer Applications and Robotics (IJRCAR), 3,* 62–68.
24. Murmu, S., Jha, S. K., Burnwal, A. P., & Kumar, V. (2015). A proposed fuzzy logic based system for predicting surface roughness when turning hard faced components. *International Journal of Computer Applications, 125*(4).
25. Kumari, N., & Burnwal, A. P. (2017). Interactive fuzzy programming model in multi-objective inventory control problem using various operators. *International Journal of Students' Research in Technology & Management, 5*(4), 18–26.

Intelligent Networking

Impacts of Computational Techniques for Wireless Sensor Networks

Chandan Kumar Shiva, Basetti Vedik, Ritesh Kumar, Sheila Mahapatra, and Saurav Raj

1 Introduction

1.1 WSN: Basic Introduction

The utility of the WSNs is to provide solutions to monitor and sense data in dangerous or isolated areas.

Wireless ad--hoc networks are occasionally described by one single performance metric [1]. Wireless sensor network (WSN) is a collective assembly of devices or may be said a combination of multiple sensors nodes of different characteristics. Sensor nodes are generally powered by batteries, but stationary nodes may also have a more reliable power source. It is composed of large quantity of low-cost, low-powered devices, multi-functional and resource-constrained sensor nodes [2]. The function of the nodes is to buffer data from sensors and perform computations using

C. K. Shiva (✉) · B. Vedik · R. Kumar
Department of Electrical and Electronics Engineering, SR Engineering College, Ananthsagar, Hasanparthy, Warangal, Telangana, India
e-mail: chandankumarshiva@gmail.com

B. Vedik
e-mail: b.vedik@gmail.com

R. Kumar
e-mail: riteshraj3336@gmail.com

S. Mahapatra
Department of Electrical and Electronics Engineering, Alliance University, Bangalore, India
e-mail: mahapatrasheila@gmail.com

S. Raj
Department of Electrical Engineering, Institute of Chemical Technology Mumbai, Marathwada Campus, Jalna, India
e-mail: sauravsonusahu@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_6

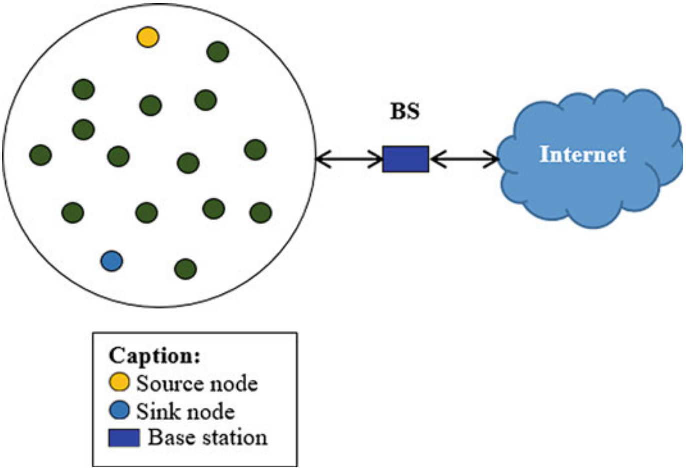


Fig. 1 General configuration of WSN

that data. Gateway devices generally have a more reliable power source and greater processing capability and can interact with outside networks. It performs measurements of some physical phenomena, collect and process data, communicate with other peers or a central information processing unit, the sink. These nodes contains a small processor sensing devices that are capable of sensing various phenomena, such as pressure, temperature, humidity, position, velocity, acceleration, force, vibration, proximity, sound motion, biochemical agents. They are also capable of processing textual, voice and video data making them useful [3]. In spite of its huge applications in wide engineering field, sensor networks are designed with flexibility to withstand harsh environmental conditions. These are networks of wireless interconnected smart devices which can be designed and deployed to retrieve sensor data of interest from their host environments [2]. The general configuration of WSN is shown in Fig. 1.

1.2 WSN: Basic Application Area

With the recent development of the technologies (like improvement in processor, communication and consumption of low-powered implanted electronic and computing devices) and more effective solution methods in wireless network domain, this network is going to be more popular and provides better results in both industrial and commercial applications [4, 5]. After the proficient design of WSN, its functioning depends upon the optimal choice of locations and power assignments of the sensors that need to be densely deployed in a neighborhood of interest. The study of the optimal choice of locations and power assignments of the sensors has been studied in [6]. The wide area of application of WSN has been shown in Fig. 2. Here,

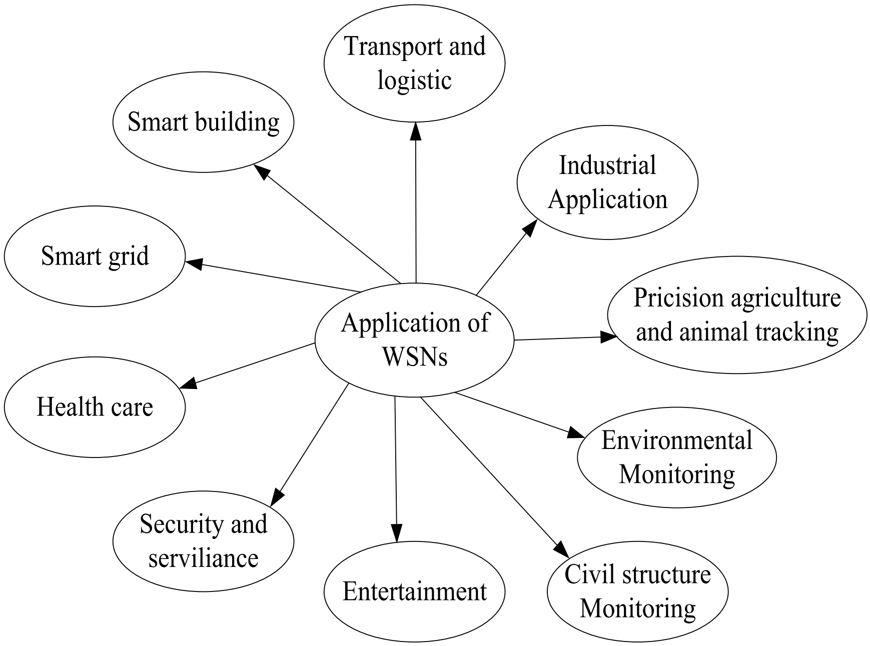


Fig. 2 General application of WSN [13]

some of the application of WSN is shown below. The detail study of wireless network such as WSN, wireless body area network, mobile ad-hoc network and wireless ad-hoc network has been studied in [7]. The application of bio-inspired computational techniques for efficient and optimal solutions for WSN problems in the contexts of fault analysis and diagnosis and traffic management has been studied in [8]. The applicability of African buffalo optimization (ABO) on the routing in the WSN has been stated in [9]. The study of a home-based wireless ECG monitoring system using Zigbee technology has been studied in [10]. Such studies are quite useful is for monitoring people from their own home as well as for periodic monitoring by physicians for appropriate health care. One of the most critical problems of WSNs that has its huge impacts on WSNs performance such as area coverage has been shown in [11]. In this work, improved cuckoo search and chaotic flower pollination algorithm are used for the study. To solve the problem of WANET nodes, i.*e*., powered by battery with limited capacity, a routing protocol named as intelligent energy competency multipath routing protocol for WANET has been studied in [12].

This chapter outlines the significance of optimization methods in WSN. It introduces the various optimization methods in this chapter and its applicability in WSN applications. It gives the details of all the used optimization method in details. In the continuation of the study, this chapter is divided into sections as follows. The primary importance of optimization techniques is shown in Sect. 2. The description of applied optimization techniques is shown in Sect. 3. The details of optimization techniques applied to WSN are shown in Sect. 4. The performance evaluation of all the methods is shown in Sect. 5. The importance of the work done is shown in Sect. 6.

2 Need of Optimization

Whenever, there is more number of variables than equations and it is necessary to determine maximum or minimum value of equation, then optimization technique is adopted to determine optimal value of variables. Optimization can be used for both maximizing and minimizing the value of equations. Therefore, optimization may be defined as "doing the most with the least" [14] or "the process of finding the most effective or favorable value or condition" [15]. This is all about the fundamentals of optimization method. The evolutionary multi-objective optimization may be a well-known and valuable field of research and creating calculations to comprehend some genuine multi-target issues related to real-life problem [16]. Optimization problems can be found in all the fields of science. Researchers or engineers are usually asked to identify best possible optimal settings for particular problem, it may be minimization or maximization based problems. The main objective of optimization is to identify "best" setting with respect to set of prioritized criteria or constraints. The constraints are divided to two categories: equality and inequality [17–19].

Computational intelligence is a powerful optimization approach that relates to human intelligence. It shows optimal solution for the problem even for the multi-objective optimization problem. In the past work done, different classical approaches have been adopted to solve the problems arising in WSNs. In the past work done, a large number of the current sensor network design method were given attention on single-objective optimization problem [20]. However, different approaches have their own advantages, disadvantages and limitations. The topics of this chapter are to study the feature comparison of the studied method with some already existing methods.

3 Description of Applied Algorithm

Multi-objective optimization methods are proposed for the quality design of WSNs which is a difficult task. The difficulty arises because the performance of WSN depends on various performance parameters. In the following section, the multi-objective optimization techniques are studied subjected to WSN related to various design problems. The description of the applied optimization techniques is shown in Fig. 3.

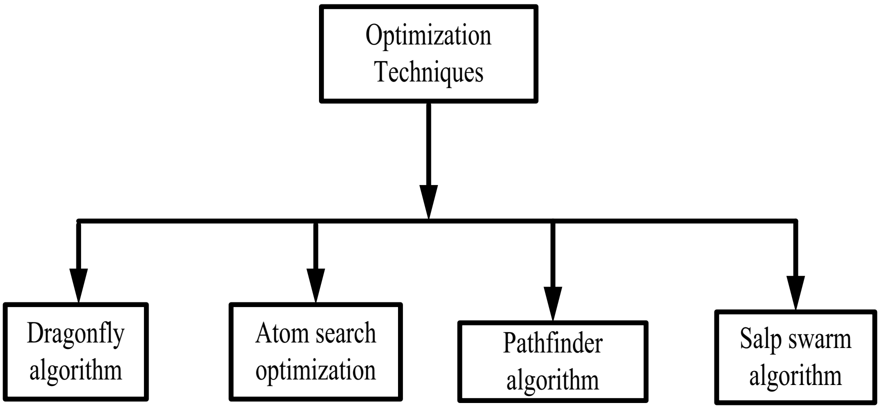


Fig. 3 Applied optimization techniques

3.1 Dragonfly Algorithm

The DA technique is proposed in the year 2015 by Mirjalili [21], which mimics the swarm behavior. In this algorithm, the dragonflies are considered as small predator that hunts the small preys. In the process of searching for preys, these dragonflies undergo two stages. These two stages of dragonflies hunting behavior is depicted in Fig. 4. Therefore, the position of dragonfly during optimization process represents the solution in search space [21–23]. Further, during optimization, five principles, namely separation, alignment, cohesion, attraction and diversion, are considered in order to avoid collision among dragonflies, to travel with same velocity to different areas with neighboring dragonflies, to move toward the center of the group, to get attraction toward to prey and to escape from being hunted, respectively.

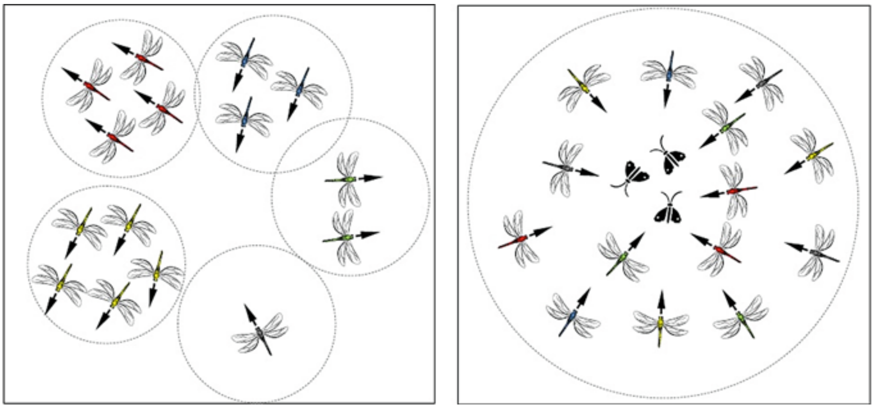


Fig. 4 Static and dynamic behavior of dragonflies [21]

Therefore, the process of hunting the preys (finding the best solution) by dragonflies is shown in the following flow chart given in Fig. 5 by considering the above five principles and using the quasi-oppositional-based learning concept.

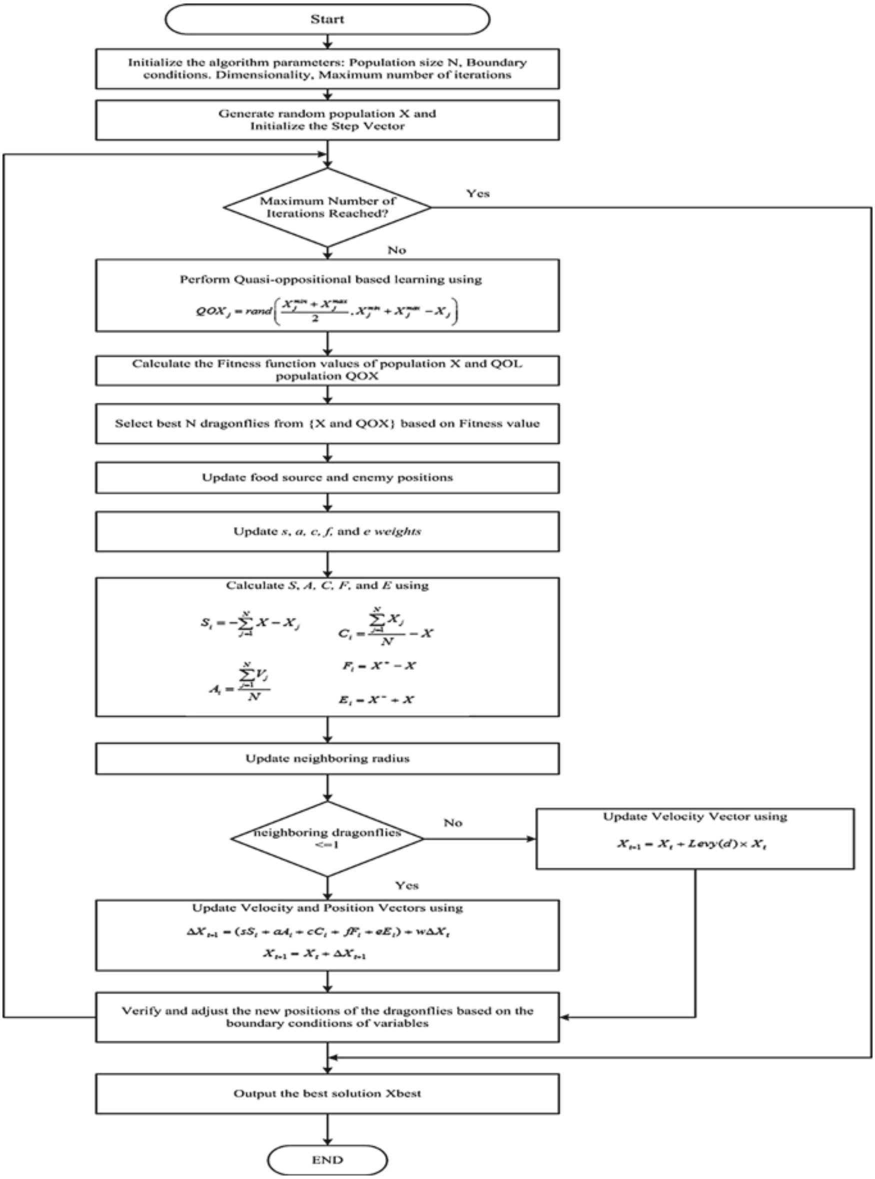


Fig. 5 Flowchart of QODA method

The detailed description of dragonfly algorithm can be found in [21–23], and quasi-oppositional-based learning can be found in [24–29].

3.2 Quasi-opposition Atom Search Optimization (QOASO) Algorithm

In 2018, Zhao et al. [30] proposed a novel physics-inspired optimization called as atom search optimization (ASO) algorithm. This technique is based on the concept of motion of atoms that tracks the traditional molecular mechanics [31]. Therefore, in ASO algorithm, the position of each atom denotes the solution for the given problem. Further, mass of the atom denotes the quality of the solution found, i.e., an atom with heavier mass depicts the best solution. Furthermore, the interactive forces between the atoms denote the exploration and exploitation of the solution for the given problem; i.e., atoms with lighter mass will be attracted toward heavier atoms and hence have high acceleration. In this process, the atoms with lighter mass explore the search space and identify the promising regions. Alternatively, the atoms with heavier mass exploit the promising regions search area due to their less acceleration [31, 32]. This process of finding the optimal solution is shown in the following flow chart shown in Fig. 6.

The detailed description of ASO can be found in [31, 32], and quasi-oppositional-based learning can be found in [24–29].

3.3 Pathfinder Algorithm (PA)

Pathfinder algorithm (PA) is a meta-heuristic technique proposed by Yapici [33] in the year 2019. This technique is based on the concept of cooperative movement of animals in group to find the food source. In search of feeding area or water or pasture, the animal swarms use their three abilities, namely searching, exploiting and hunting. The authors in [33] suggest the movement of individual herd of a group not only depends on the movement of leader but also the neighboring individual. Based on the behavior of animals in finding target, the mathematical modeling of the algorithm is explained as follows.

In order to find the food source, the leader of the group needs to be identified, for which the fitness of each herd in the group is evaluated and the best fitness herd of the group is selected as the pathfinder [33]. This pathfinder moves in the search space according to Eq. (1).

$$\left.\begin{aligned} x_p^{\text{iter}+1} &= x_p^{\text{iter}} + 2 \times r_1 \times \left(x_p^{\text{iter}} - x_p^{\text{iter}-1}\right) + A \\ A &= u_1 \times e^{\frac{-2\times \text{iter}}{\max_{\text{iter}}}} \end{aligned}\right\} \quad (1)$$

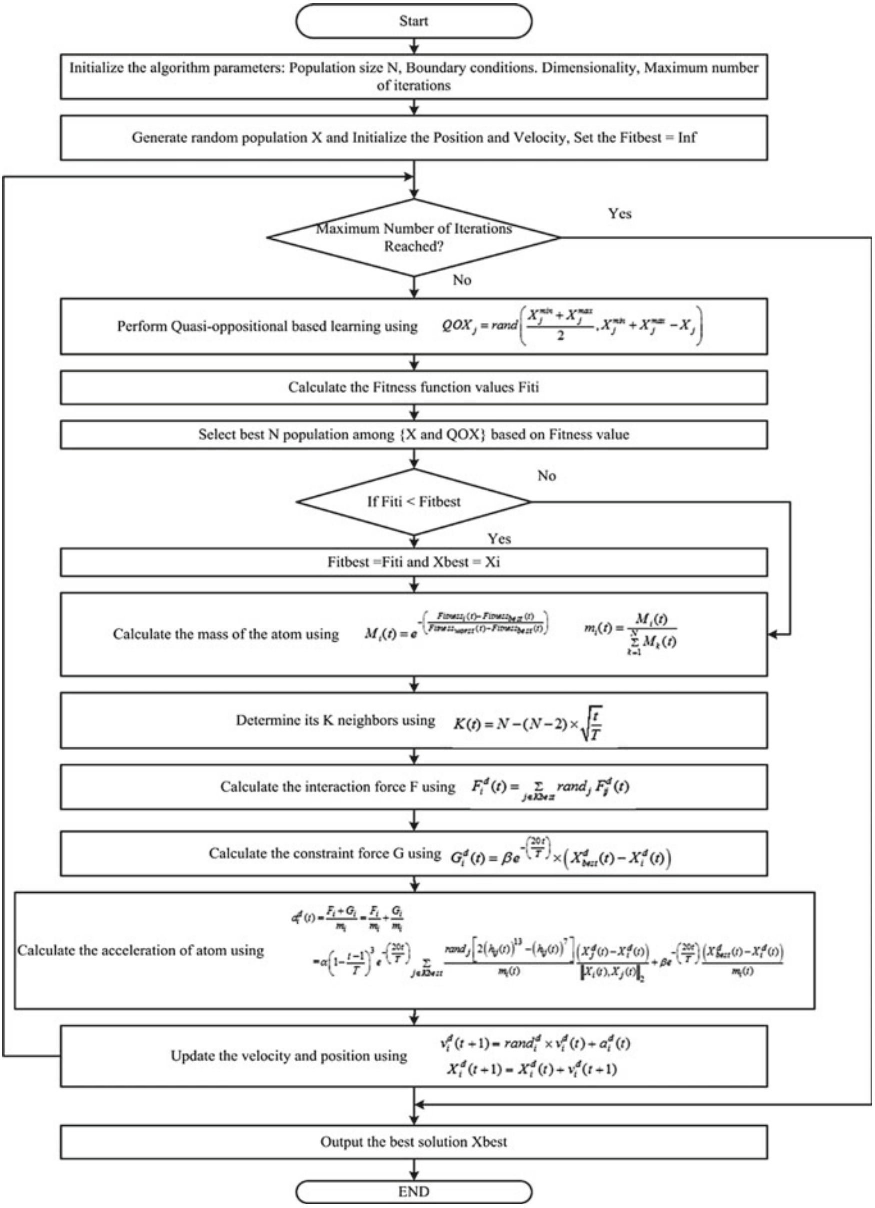


Fig. 6 Flowchart of QOASO method

Here, the pathfinder attempts to locate the food source area. This food source area is assumed to be global optimum solution. Further, for any iteration, the position of the leader/pathfinder is considered as the current optimum solution and remaining herd move toward this current optimum solution [33]. Therefore, the movement of remaining herd movements with respect to pathfinder and neighboring herd is modeled as follows:

$$\left.\begin{aligned} x_i^{k+1} &= x_i^k + L \times \left(x_j^k - x_i^k\right) + M \times \left(x_p^k - x_i^k\right) + \varepsilon, \qquad i \geq 2 \\ \varepsilon &= \left(1 - \frac{\text{iter}}{\text{max iter}}\right) \times u_2 \times D_{ij} \end{aligned}\right\} \tag{2}$$

where x_i represents the position vector of ith individual herd, L and M is equal to αr_2 and βr_3 respectively, r_2 and r_3 denotes the random variable generated uniformly in the range [0, 1], α represents the coefficient of interaction between the individual herd with its neighbor herd, β represents the coefficient of attraction that keeps the individual herd of the group at a random distance with its pathfinder, ε denotes the vibration which ensures random movement of the herds with in specified range, u_2 represents the random vector generated in the range of [−1,1], and D_{ij} signifies the distance between two neighbor herds, i.e., $D_{ij} = \left\|x_i - x_j\right\|$. The values of α and β are selected randomly in the range of 1–2 [33].

3.3.1 Quasi-opposition Pathfinder Algorithm (QOPA)

After updating the position of each herd in the group, the concept of quasi-opposition-based learning (QOBL) is applied to enhance the convergence speed of the pathfinder algorithm. To perform QOBL, current candidate solution along with its quasi-opposite candidate solution is considered to attain improved candidate solution. The quasi-opposite candidate solution is created for each solution present in the current population.

3.4 Salp Swarm Algorithm

Salp swarm algorithm (SSA) is a contemporary algorithm developed by Mirjalili established primarily on the swarming behavior of jelly-like sea creatures called salps [34]. Salps belong to the Salpidae group, and swarm of salps forms a riveting salp chain which can be modeled mathematically as an evolved algorithm. The emergence of salp chain facilitates better coordination and versatility for probing the food [35]. The basic updating process in SSA involves the transition of leader salp position with respect to food source, and the follower salp reconditions their position with respect to each other. The updated position of the leader salp is given by Eq. (3).

$$x_k^1 = \begin{cases} F_k + c_1((ub_k - lb_k)c_2 + lb_k) & c_3 \geq 0 \\ F_k - c_1((ub_k - lb_k)c_2 + lb_k) & c_3 \prec 0 \end{cases} \tag{3}$$

The coefficient c_1 is the primary governing parameter which justifies the tradeoff between exploitation and exploration and is given by Eq. (4):

$$c_1 = 2e^{-\left(\frac{4m}{M}\right)^2} \tag{4}$$

where m is the current iteration and M is the total number of iterations. The significance of parameter c_1 is its ability to be adaptive such that it gradually decreases in search space making SSA first explore and then exploit the search space. The salp followers are updated based on Newton's law of motion and is given in Eq. (5).

$$x_k^i = \frac{1}{2}\left(x_k^i + x_k^{i-1}\right) \tag{5}$$

where x_k^i is the position of ith salp follower in kth dimension and $i \geq 2$. Equations (1) and (3) simulate the salp chain. The SSA generates a gradual salp movement, thereby avert getting trapped in local optima problem. The best solution of the salps is graded in accordance with their fitness function, and the optimal outcomes are saved.

4 Optimization Techniques Applied in WSN

The modeling of the WSN may be classified into four parts, i.e., consumption of energy, path loss model, system lifetime model and consumption of energy [3].

4.1 Modeling of Consumption of Energy

The total energy consumption of each node of the satellite can be calculated as a summation of both transmission energy and circuit energy consumption [3].

Let, E_i is the energy consumption of ith node of satellite during transmission time T_i is given as follows.

$$E_i = T_i(P_{ti} + P_{ci}) \tag{6}$$

where P_{ci} and P_{ti} denote circuit power consumption and transmission power consumption, respectively, for ith node of the satellite.

The circuit power consumption having a co-relation with transmission power consumption is given as follows.

$$P_{ci} = \alpha_i + \left(\frac{1}{\eta} - 1\right) P_{ti} = \alpha_i + P_{ai} \tag{7}$$

where α_i represents component of transmit power independent which approximately used to consider for the consumed power by circuit unit for ith node of satellite. P_{ai} represents the power consumed by power amplifier used in analog to digital converter which is associated with the circuit unit and η represents the efficiency of power amplifier which can be calculated as follows.

$$\eta = \left(\frac{P_{ti}}{P_{ti} + P_{ai}}\right) \tag{8}$$

The entire energy consumption of ith satellite node is defined as follows.

$$E_i = \frac{1}{\eta} P_{ti} T_i + \alpha_i T_i = \frac{1}{\eta}(P_{ti} + \alpha_{ci}) \tag{9}$$

where α_{ci} represents the power consumption for an equivalent circuit for N number of satellite node; the entire energy consumption can be calculated as follows.

$$E_i = \sum_{i=1}^{N} E_i = \frac{1}{\eta}(P_{ti} + \alpha_{ci}) T_i \tag{10}$$

4.2 Path Loss Model

Path loss defined as the ratio of transmitted power P_{ti} of satellite node *i,* and the receiving power P_{rc} received by head of the system is stated as follows.

$$P_{Li} = 10 \log\left(\frac{P_{ti}}{P_{rc}}\right) \tag{11}$$

The receiving power P_{rc} can be calculated as follows.

$$P_{rc} = \left(\frac{P_{ti} G_{ti} G_{rc} \lambda^2}{16\pi^2 d_i^2 \beta}\right) \tag{12}$$

where G_{ti} denotes gain of ith satellite node, G_{rc} gain of receiving system head, d_i^2 is Euclidian distance between the ith satellite node and receiving cluster head, λ is the wavelength, and it can be calculated using $v = n\lambda$ where v is velocity of light, and n indicates the frequency, β is the system loss factor, associated with polarization mismatch between the antenna.

4.3 System Lifetime Model

In wireless communication networks, there are different definitions of system lifetime. Here, for system-based satellite network considered three definition, i.e., orbital lifetime (τ_{orb}), node lifetime (τ_{nod}) and system lifetime $\left(\tau_{sys}\right)$. Considering all three, it is presumed as follows.

$$\tau_{\text{orb}} \geq \tau_{\text{nod}} \geq \tau_{\text{sys}} \tag{13}$$

The nodal lifetime of ith satellite can be defined as follows.

$$\tau_{\text{nod},i} = \frac{\eta E_{b,i}}{P_{ti} + \alpha_{ci}} \tag{14}$$

where $E_{b,i}$ is the total energy of battery of ith satellite.

And, the system lifetime can be formulated as follows.

$$\tau_{\text{sys}} = \min\left\{\tau_{\text{nod},1,}\tau_{\text{nod},2,} \ldots, \tau_{\text{nod},N}\right\} \tag{15}$$

4.4 Coverage Model

The wireless sensors network in the space can be designed and monitor within a critical area for security purpose. The coverage ϕ_{c} is defined as follows.

$$\phi_{\text{c}} = \frac{A_{\text{c}}}{A_{\text{t}}} \tag{16}$$

where A_{c} and A_{t} are the coverage area and total of interest, respectively.

4.5 Multi-objective Optimization Problem

The design of WSNs is a relatively complex integrated task because of its performance depends on multi-objective system performance parameters such as energy consumption, system lifetime, coverage, and number of satellites. One of the design goals is to maximize the lifetime of the sensor network [3, 36]. For the optimization, the following mentioned scenarios have been discussed as followed.

Scenario (a): Optimization of energy consumption
Scenario (b): Optimization of system lifetime
Scenario (c): Coverage optimization problem

Scenario (d): Optimization of the participating number of satellites.

Scenario (a) Optimization of energy consumption

The optimization of energy consumption can be expressed as follows.

$$\begin{aligned} \min f_1(P_t, T, N) = \frac{1}{\eta}(P_{ti} + \alpha_{ci})T_i \\ \text{where, } P_t = \left[P_{t,1,} P_{t,2,} \ldots, P_{t,N}\right]^T \end{aligned} \tag{17}$$

T represents transmission duration of total satellite and is defined as follows.

$$T = \left[T_{1,} T_{2,} \ldots, T_{,N}\right]^T \tag{18}$$

Scenario (b) Optimization of system lifetime

The optimization of system lifetime can be defined as the minimization problem as follow.

$$\min f_2(P_t, T, N) = \left[\tau^* - \tau_{\text{sys}}\right] \tag{19}$$

$$\text{It has been assumed that } \tau^* \geq \tau_{\text{orb},i} \tag{20}$$

Scenario (c) Coverage optimization problem

The main objective of space-based WSN is to maximize the coverage of each system with minimum number of nodes, and the optimization of coverage can be expressed as follows.

$$\begin{aligned} \min f_3(N) = (1 - \phi_c) \\ \text{Subjected to } 0 \leq N \leq N_{\max} \end{aligned} \tag{21}$$

where $N_{\max}$ is the maximum number of satellites available to form cluster-based sensor network.

Scenario (d) Optimization of the participating number of satellites

The number of participating satellites in the system needs to be minimized, and it can be expressed as follows.

$$\begin{aligned} \min f_4(N) = N \\ \text{Subjected to } 0 \leq N \leq N_{\max} \end{aligned} \tag{22}$$

Table 1 Different features of the optimization techniques

DA [37]	ASO [30]	SSA [38]	QOPA [33]
Few parameters needed to tune	Algorithm needs fewer parameters	Good convergence acceleration	Fast convergence characteristics
Convergence time is reasonable	Good global exploration ability	Adaptability, robustness, and scalability	Few algorithmic parameters
No internal memory that can lead to premature convergence to the local optimum		Reasonable execution time and few parameter tuning	

5 Performance Evaluation

Here, the objective is to show the comparative analysis of the studied swarm computation techniques with some of the stated features. In this chapter, the recent computational techniques QODA, QOASO, QOPA and SSA are compared with ANO, PSO, ABC and GSA. Some of the features of the recent optimization techniques are stated in Table 1. The features comparisons of the studied methods are shown in Table 2.

6 Conclusion

In this study, the features comparison of WSN have been studied with the recently developed algorithms like QODA, QOASO algorithm, SSA and QOPA as compared to previous studied optimization techniques like ANO, PSO, ABC and GSA. The problems being discussed are energy consumption, system lifetime, coverage optimization problem and participation of number of satellites in this work. The study showed that the recent optimization techniques provide good results for the stated features. The results showed that the studied optimization techniques are powerful to solve the problems of WSN. In the future work, some more completed problems of WSN can be studied with the hybridization of two or more powerful optimization techniques.

Table 2 Feature comparison of the proposed method with some existing methods

Features	ANO	PSO	ABC	GSA	QODA	QOASO	QOPA	SSA
1	√	√	√	√	√	√	√	√
2	√	√	√	√	√	√	√	√
3	×	×	×	×	×	×	×	×
4	3	2	2	2	4	4	4	4
5	3	2	2	2	4	4	4	4
6	3	2	2	2	4	4	4	4
7	2	2	2	2	1	1	1	1
8	3	2	2	2	4	4	4	4
9	2	2	2	2	1	1	1	1
10	3	2	2	2	4	4	4	4
11	2	2	2	2	1	1	1	1
12	2	2	3	3	4	4	4	4
13	2	2	3	3	4	4	4	4
14	2	2	3	3	4	4	4	4
15	2	2	3	3	4	4	4	4
16	2	2	3	3	4	4	4	4
17	2	2	3	3	4	4	4	4
18	2	2	3	3	4	4	4	4

Caption

1: Routing loop avoidance	2: Source initiated	3: Receiver initiated
4: QoS support	5: Residual energy	6: Network lifetime
7: Delay	8: Packet delivery ratio	9: Communication overhead
10: Throughput	11: Packet loss	12: Scalability
13: Bandwidth	14: Robustness	15: Connectivity status
16: Handling high mobility	17: Handling traffic load	18: Handling mutual interference
Yes: √	No: ×	Very High:4
High: 3	Medium: 2	Low: 1

References

1. Jaffres-Runser, K., Comaniciu, C., & Gorce, J. M. (2013). A multiobjective optimization framework for routing in wireless ad hoc networks. *Ad Hoc Networks, 11*, 2147–2171. arXiv preprint arXiv:0902.0782.
2. Pinto, A. R., Poehls, L. B., Montez, C., & Vargas, F. (2012). Power optimization for wireless sensor networks. IntechOpen. In *Wireless Sensor Networks-Technology and Applications*.
3. Iqbal, M., Naeem, M., Anpalagan, A., Ahmed, A., & Azam, M. (2015). Wireless sensor network optimization: Multi-objective paradigm. *Sensors, 15*(7), 17572–17620.
4. Aggarwal, R., Mittal, A., & Kaur, R. (2016). Various optimization techniques used in wireless sensor networks. *International Research Journal of Engineering and Technology (IRJET),*

3(06), 2085–2090.
5. Shiva, C. K., & Kumar, R. (2020). Quasi-oppositional harmony search algorithm approach for Ad Hoc and sensor networks. In *Nature Inspired Computing for Wireless Sensor Networks* (pp. 175–194). Springer, Singapore.
6. Konstantinidis, A., & Yang, K. (2011). Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing, 11*(6), 4117–4134.
7. Das, S. K., Samanta, S., Dey, N., & Kumar, R. (Eds.). (2020). *Design frameworks for wireless networks*. Singapore: Springer.
8. De, D., Mukherjee, A., Das, S. K., & Dey, N. (2020). *Nature inspired computing for wireless sensor networks*. Springer.
9. Bera, S., Das, S. K., & Karati, A. (2020). Intelligent routing in wireless sensor network based on african buffalo optimization. In *Nature Inspired Computing for Wireless Sensor Networks* (pp. 119–142). Singapore: Springer.
10. Dey, N., Ashour, A. S., Shi, F., Fong, S. J., & Sherratt, R. S. (2017). Developing residential wireless sensor networks for ECG healthcare monitoring. *IEEE Transactions on Consumer Electronics, 63*(4), 442–449.
11. Binh, H. T. T., Hanh, N. T., & Dey, N. (2018). Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. *Neural Computing and Applications, 30*(7), 2305–2317.
12. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015). Intelligent energy competency multipath routing in wanet. In *Information Systems Design and Intelligent Applications* (pp. 535–543). New Delhi: Springer.
13. Umashankar, M. L., & Ramakrishna, M. V. (2017). Optimization techniques in wireless sensor networks: A survey. *6*, 704–708.
14. Gomez, A. G., & Oakes, W. C. (2004). *Engineering your future: A project-based introduction to engineering*. Great Lakes Press.
15. Lockhart, S. D., & Johnson, C. M. (2000). *Engineering design communication: Conveying design through graphics*. Addison-Wesley.
16. Guliashki, V., Toshev, H., & Korsemov, C. (2009). Survey of evolutionary algorithms used in multiobjective optimization. *Problems of Engineering Cybernetics and Robotics, 60*(1), 42–54.
17. Nandi, M., Shiva, C. K., & Mukherjee, V. (2019). Moth-flame algorithm for TCSC-and SMES-based controller design in automatic generation control of a two-area multi-unit hydro-power system. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 1–24.
18. Nandi, M., Shiva, C. K., & Mukherjee, V. (2019). A moth–flame optimization for UPFC–RFB-based load frequency stabilization of a realistic power system with various nonlinearities. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *43*(1), 581–606.
19. Verma, S., & Shiva, C. K. (2020). A novel salp swarm algorithm for expansion planning with security constraints. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 1–10.
20. Özdemir, S., Bara'a, A. A., & Khalil, Ö. A. (2013). Multi-objective evolutionary algorithm based on decomposition for energy efficient coverage in wireless sensor networks. *Wireless Personal Communications, 71*(1), 195–215.
21. Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications, 27*(4), 1053–1073. https://doi.org/10.1007/s00521-015-1920-1. (Springer, London).
22. Rahman, C. M., & Rashid, T. A. (2019). Dragonfly algorithm and its applications in applied science survey. *Computational Intelligence and Neuroscience, 2019,* 1–21. https://doi.org/10.1155/2019/9293617.
23. Bao, X., Jia, H., & Lang, C. (2019). Dragonfly algorithm with opposition-based learning for multilevel thresholding color image segmentation. *Symmetry, 11*(5), 716–740. https://doi.org/10.3390/sym11050716.

24. Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Comput Gr, 21,* 25–34.
25. Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. In *Proceeding of International Conference on Computational Intelligence, Modelling Control and Automation* (Vol. 1, pp. 695–701).
26. Guha, D., Roy, P., & Banerjee, S. (2017). Quasi-oppositional symbiotic organism search algorithm applied to load frequency control. *Swarm and Evolutionary Computation, 33,* 46–67.
27. Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2008). Opposition versus randomness in soft computing techniques. *Applied Soft Computing, 8*(2), 906–918.
28. Guha, D., Roy, P. K., & Banerjee, S. (2016). Load frequency control of large scale power system using quasi-oppositional grey wolf optimization algorithm. *Engineering Science and Technology, an International Journal, 19*(4), 1693–1713.
29. Shaw, B., Mukherjee, V., & Ghoshal, S. P. (2014). Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm. *International Journal of Electrical Power & Energy Systems, 55,* 29–40.
30. Zhao, W., Wang, L., & Zhang, Z. (2019). Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems, 163,* 283–304.
31. Fu, Y., Li, Z., Qu, C., & Chen, H., (2020). Modified atom search optimization based on immunologic mechanism and reinforcement learning. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2020/4568906.
32. Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems, 91,* 601–610.
33. Yapici, H., & Cetinkaya, N. (2019). new meta-heuristic optimizer: Pathfinder algorithm. *Applied Soft Computing Journal, 78,* 545–568.
34. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191.
35. Mahapatra, S., Raj, S., Krishna, S. M. (2020). Optimal TCSC location for reactive power optimization using oppositional salp swarm algorithm. In *Innovation in Electrical Power Engineering, Communication, and Computing Technology* (pp. 413–424). Singapore: Springer.
36. Yang, E., Erdogan, A. T., Arslan, T., & Barton, N. H. (2011). Multi-objective evolutionary optimizations of a space-based reconfigurable sensor network under hard constraints. *Soft Computing, 15,* 25–36.
37. Rahman, C. M., & Rashid, T. A. (2019). Dragonfly Algorithm and Its Applications in Applied Science Survey. *Computational Intelligence and Neuroscience.*
38. Abualigah, L., Shehab, M., Alshinwan, M., & Alabool, H. (2019). Salp swarm algorithm: A comprehensive survey. *Neural Computing and Applications*, 1–21.

Efficient Node Deployment Based on Immune-Inspired Computing Algorithm for Wireless Sensor Networks

Nabil Sabor and Mohammed Abo-Zahhad

1 Introduction

One of the key points in the design of wireless sensor networks (WSNs) is the coverage of the sensor field. Many factors affect the network coverage, namely the sensing model, and the deployment strategy. The popular deployment methods for the sensor nodes in WSNs are random deployment and deterministic deployment. The random deployment is easy and less expensive, and it is preferred for large hostile environments such as battlefield or forest environment. However, random deployment can cause coverage holes in the sensor field, while the deterministic deployment is used in applications where the deployment is physically reachable and has a small size. Although the deterministic deployment satisfies the network connectivity, it is complex in large networks and harsh environments [1–3].

In order to combine the advantages of the random and deterministic methods, the randomly deployed nodes should be automatically rearranged in the field to satisfy the coverage requirements. This can be done by attaching the sensor nodes with vehicles. However, the mobility systems of the nodes expend some energy and this adds a new challenge in designing WSN [4, 5]. Therefore, the problem is how to redeploy the sensor nodes subject to balance the trade-off between the network coverage and the consumption energy. This is a non-deterministic polynomial-time hard (NP-hard) problem. Many deployment algorithms were developed for solving the coverage

N. Sabor (✉) · M. Abo-Zahhad
Electrical and Electronics Engineering Department, Faculty of Engineering, Assiut University, Assiut 71516, Egypt
e-mail: nabil_sabor@aun.edu.eg

M. Abo-Zahhad
e-mail: mohammed.zahhad@ejust.edu.eg

Department of Electronics and Communications Engineering, E-JUST-Alexandria, New Borg El Arab, Egypt

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*,
Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_7

problem as illustrated in Table 1. These algorithms are classified according to control manner (i.e., running method) into centralized (C) and distributed (D) algorithms. The sensing unit of sensor nodes is simulated as a binary model or a probabilistic model. The objectives, features, and drawbacks of the developed algorithms are explained in Table 1.

In this chapter, two energy-efficient deployment algorithms are proposed depending on the multi-objective immune algorithm (MOIA) to maximize the coverage of the network with a minimum mobility cost. The first deployment algorithm is called an immune-based node deployment algorithm (INDA). The INDA utilizes the MOIA to relocate the deployed sensor nodes in the network based on maximizing the network coverage and minimizing the moving cost, while the second deployment algorithm is called a centralized Voronoi-based immune deployment algorithm (CVIDA). The idea of the CVIDA is based on combining the MOIA and the Voronoi diagram to find the locations of the sensor nodes and the optimal working nodes based on reducing the mobility cost, adjusting the sensing range, and controlling the communication radio (i.e., active/sleep) of each node. CVIDA takes into its consideration the dissipated energy in the sensing, the mobility, and the redundant coverage besides the network coverage.

2 Multi-objective Immune Algorithm

Multi-objective immune algorithm (MOIA) [6–8] is one of the nature-inspired computing algorithms that mimics the antigen–antibody reaction in the mammal's immune system. The antibody and the antigen in the MOIA represent the feasible solution and the objective function for a traditional optimization method. MOIA is a global search algorithm and produces solution sets with a different diversity, convergence, and distribution. Also, it has much less computational cost because it adapts its population according to the defined problem. The framework of MOIA is shown in Fig. 1. The details of the MOIA are as follows:

(a) **Population Production**

Initially at gen = 0, a population of p_s antibodies, pop(gen) $= \left(A_1; A_2; \ldots; A_\mathrm{m}; \ldots; A_{p_\mathrm{s}}\right)$, is randomly produced using a real coding. Each antibody $A_\mathrm{m} = \left\{x_1, x_2, \ldots, x_j, \ldots, x_\mathrm{n}\right\}$ consists of n numbers to represent the decision variable vector (x).

(b) **Population Evaluation**

In order to find the affinity between antibody and antigen, each antibody in the population is evaluated based on the value of the objective function $(F(x))$.

(c) **Antibody Selection**

The generation of strong offspring antibodies depends on parent antibodies. Therefore, the selection of the best parents is performed using the roulette wheel selection

Table 1 Comparison of the node deployment algorithm for WSNs

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
PSO_Voronoi algorithm 2009 [10]	C	Area coverage	Binary model	Maximizes network coverage	Free obstacle fields	• Adjusts the locations of the sensor nodes using PSO • Maximizes network coverage	• Neglected the dissipated energy in the mobility and sensing of the sensor nodes • It did not consider the effect of obstacles • Requires location information
NSGA-II algorithm 2009 [11]	Semi-C	Area coverage	Binary model	Maximize the network coverage using a minimum number of the sensor nodes and adjust the sensing range	Free obstacle fields	• Adjusts the sensing ranges of active sensor nodes to maximize the network coverage • Activates the required sensor nodes for achieving the network coverage	• It did not consider the effect of obstacles • The authors did not explain the dividing process of the sensor field into squares and the size of each square
Jin et al. algorithm 2010 [3]	C	Area coverage	Binary model	Maximizes the network coverage and reduces the energy consumption of the sensing and the redundant coverage	Free obstacle fields	• Considers the consumption energy in the sensing, the redundant coverage besides the area coverage • Adjusts the sensing ranges and the locations of nodes	• Neglected the mobility cost of the sensor nodes • It did not consider the effect of obstacles • Requires location information

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
Qu et al. algorithm 2011 [2]	C	Area coverage	Binary model	Maximizes network coverage and minimizes the traveled distance	Free obstacle fields and full of obstacle fields	• Considers both the coverage and energy conservation • Considers the fields with obstacles	• Requires a large number of calculations especially when dealing with the sensing fields that contain obstacles • Requires location information
WSNPSO algorithm 2011 [12]	C	Area coverage	Binary model	Maximizes the network coverage and minimizes the maximum traveled distance	Free obstacle fields	• Utilizes the GA to maximize the coverage area with a minimum mobility cost • Uses the Voronoi diagram in the calculation of the coverage area.	• Requires location information • It did not consider the dissipated energy in the sensing process • It did not consider the effect of obstacles
NN algorithm 2012 [13]	D	Area coverage	Binary model	Maximizes the network coverage	Free obstacle fields and full of obstacle fields	• It is a self-distributed algorithm • Utilizes the neural network and the genetic algorithm to achieve its objectives	• It assumes that the sensing area of each sensor nodes is square area and this is unrealistic

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
BBO algorithm 2012 [1]	C	Area coverage	Binary model	Maximizes the network coverage	Free obstacle fields	• Utilizes the BBO algorithm to enhance the coverage of WSNs	• Does not consider the dissipated energy in the sensing and the mobility • It has a slow rate of convergence • It did not consider the effect of obstacles
VF-BBO algorithm 2013 [14]	C	Area coverage	Binary model	Maximizes the network coverage	Free obstacle fields	• Utilizes the BBO algorithm to enhance the coverage of WSNs and the VF algorithm to fast the convergence to the best solution.	• It did not consider the dissipated energy in sensing and mobility • It did not consider the effect of obstacles
$WSNPSO_{con}$ algorithm 2013 [15]	C	Area coverage	Binary model	Maximizes the network coverage and minimizes the moved distance	Free obstacle fields	• Maximizes the network coverage and minimizes the moved distance of the sensor nodes	• It did not consider the dissipated energy in the sensing process • It did not consider the effect of obstacles • Requires location information

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
GA-deployment algorithm 2013 [16]	C	Target coverage	Binary model	Finds the number and location of the additional mobile nodes	Free obstacle fields	• Utilizes the GA to maximize the target coverage Alleviates the coverage holes	• Requires location information • It did not consider the mobility cost in the objective function
Self-adjusting algorithm 2015 [17]	D	Area coverage	Binary model	Maximizes the network coverage based on a threshold distance	Free obstacle fields	• It does not need global topology information of the sensor nodes • Considers a threshold distance in improving the coverage	• It did not consider the effect of obstacles • The moving of the sensor nodes is random which does not ensure the network coverage
EBC algorithm 2016 [18]	D	Area coverage	Binary model	Maximize the network coverage and minimize the mobility cost	Free obstacle fields	• Tends the sensor nodes to move to the center of the Voronoi polygon • It has a fast convergence rate	• Each sensor node requires the location information of its neighbors at each iteration to construct the Voronoi polygon

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
IGA-BAC algorithm 2016 [19]	C	Area coverage	Binary model	Maximizes the network coverage and deactivates the redundant nodes	Free obstacle fields	• Minimizes energy consumption by activating a minimum number of the nodes • It has a fast convergence rate	• The authors did not consider the case when the number of sensor nodes is small to cover the field • Requires location information of the sensor nodes
CSAPO algorithm 2013 [20]	C	Area coverage	Probabilistic model	Maximize the network coverage and minimize the traveled distance	Free obstacle fields	• Considers both the coverage and the mobility cost in its objective function • Considers the APO algorithm to avoid trapping in a local optimum	• Requires information of the sensor nodes • It did not consider sensing fields with obstacles
QCA algorithm 2013 [21]	C	Area coverage	Probabilistic model	Maximize the network coverage and minimize the redundancy coverage	Free obstacle fields	• Considers the QCA algorithm to improve the network coverage and minimizes the redundancy coverage	• Considered only network coverage • Requires information of the sensor nodes

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
PSO-S algorithm 2014 [22]	D	Target coverage	Probabilistic model	Maximize the target coverage and minimize the mobility cost	Free obstacle fields	• It is a distributed algorithm and requires only the local information of neighbors • Considers the coverage and the traveled distance in its objective function	• It did not consider fields with obstacles • Authors do not consider the dissipated energy in sensing
EEBC algorithm 2015 [23]	–	Barrier coverage	Binary model	Maximize the barrier coverage with minimum mobility cost	Free obstacle fields	• Solves the maximum lifetime barrier coverage problem • Considers long irregular strip region	• Assumes that each sensor node knows its location • It did not consider the effect of obstacles in the field
HS algorithm 2017 [24]	C	Area coverage	Binary and probabilistic model	Improving the network coverage and minimizing the network cost	Free obstacle fields	• It maximizes the coverage ratio with the optimal number of deployed sensor nodes • It studied the effect of cell size and the sensing range on the network coverage	• It did not consider the obstacle networks • Requires information of the sensor nodes

(continued)

Table 1 (continued)

Algorithm	Control manner	Network coverage	Coverage model	Deployment objectives	Sensor field type	Features	Drawbacks
3D deployment algorithm 2018 [25]	C	Area coverage	Probabilistic model	Improving coverage and lifetime of the network	Full of obstacle fields	• It improves the coverage of 3D industrial space with obstacles • It deploys relay nodes to maximize the network lifetime	• It requires the locations of the sensor nodes • The deployment process consumes more time

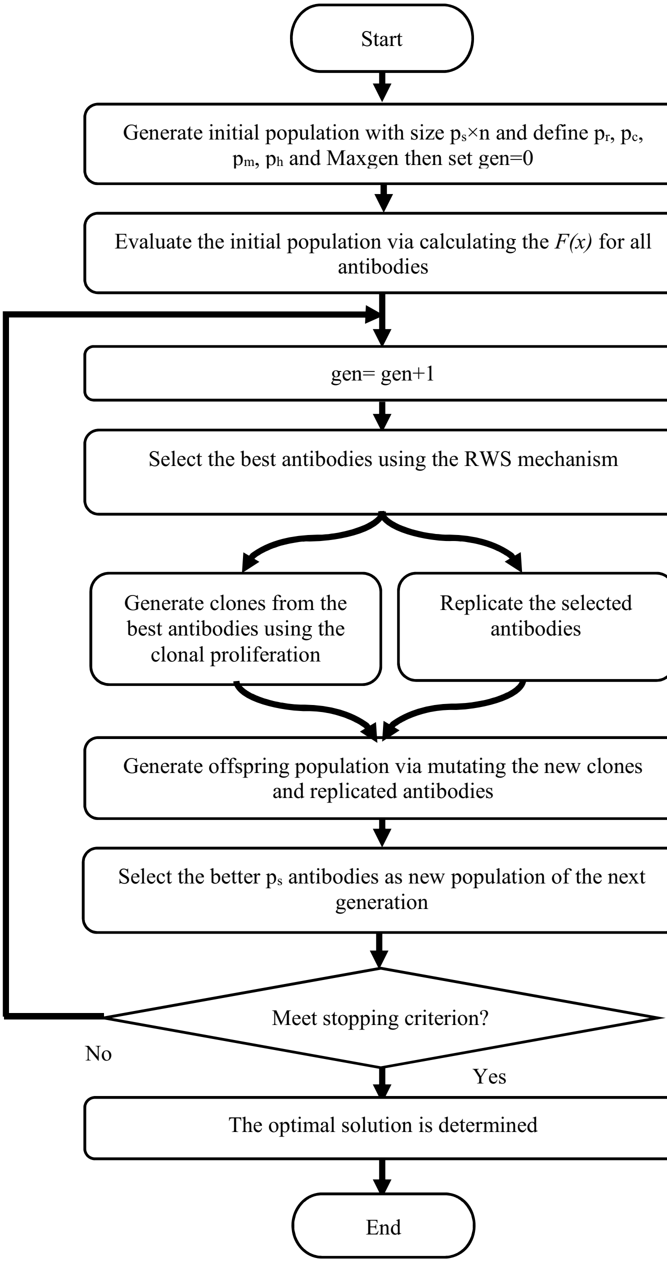


Fig. 1 Flowchart of the MOIA

(RWS) mechanism [9]. The principle of RWS is based on the probability of each A_{m} antibody that proportions to its fitness value ($\mathrm{Fit}(x) = 1/F(x)$). The antibodies with the highest fitness values are selected to form pop_sel $= \left(\dot{A}_1; \dot{A}_2; \ldots; \dot{A}_{\mathrm{m}}; \ldots; \dot{A}_{p_{\mathrm{s}}}\right)$ which will be used for generating the offspring antibodies in the replication and the clonal proliferation processes.

(d) **Replication**

In the replication process, the best $p_{\mathrm{r}}.p_{\mathrm{s}}$ antibodies are selected from pop_sel to construct pop_rep $= \left(\ddot{A}_1; \ddot{A}_2; \ldots; \ddot{A}_{\mathrm{m}}; \ldots; \ddot{A}_{p_{\mathrm{r}}.p_{\mathrm{s}}}\right)$ based on the replication rate (p_{r}). These antibodies are used to generate offspring through the mutation process.

(e) **Clonal Proliferation within Hypermutation**

Clonal proliferation operation is used to make the selected antibodies (pop_sel) proliferate and produce clone to increase the antibodies' diversity. Depending on the clonal rate (p_{c}), the best $p_{\mathrm{s}}.p_{\mathrm{c}}$ antibodies are chosen from pop_sel to mutate through the hypermutation process and form pop_hyper $= \left(\bar{A}_1; \bar{A}_2; \ldots; \bar{A}_{\mathrm{m}}; \ldots; \bar{A}_{p_{\mathrm{c}}.p_{\mathrm{s}}.N_{\mathrm{clon}}}\right)$. The pop_hyper is formed via producing N_{clon} clones for each antibody through mutating some bits in each antibody using the non-uniform mutation [26] based on the hypermutation rate (p_{h}). If the antibody $\dot{A}_{\mathrm{m}} = \{x_1, x_2, \ldots, x_j, \ldots, x_{\mathrm{n}}\}$ is chosen depending on p_{c}, the resulting clone is $\bar{A}_{\mathrm{m}} = \left\{x_1, x_2, \ldots, x_j^{'}, \ldots, x_{\mathrm{n}}\right\}$, where the new gene $x_j^{'}$ is given by Eq. 1; where α and β are the random numbers in the range of [0, 1], gen is the generation number, $\mathrm{Max}_{\mathrm{gen}}$ is the maximum number of generations, *ub* and *lb* are the upper and lower values of variable x_j and η is a system parameter determining the degree of non-uniformity. In our algorithm, η is set by 3.

$$x_j^{'} = \begin{cases} x_j + \alpha\left(ub - x_j\right)\left(1 - \frac{\mathrm{gen}}{\mathrm{Max}_{\mathrm{gen}}}\right)^{\eta}, & \text{if} \beta < 0.5 \\ x_j - \alpha\left(x_j - lb\right)\left(1 - \frac{\mathrm{gen}}{\mathrm{Max}_{\mathrm{gen}}}\right)^{\eta}, & \text{if} \beta \geq 0.5 \end{cases} \tag{1}$$

(f) **Mutation Operation**

In this step, the replicated population (pop_rep) and hypermutated population (pop_hyper) are mutated using the non-uniform mutation operation to provide exploration and produce an offspring population pop_off $= \left(\bar{\bar{A}}_1; \bar{\bar{A}}_2; \ldots; \bar{\bar{A}}_{\mathrm{m}}; \ldots; \bar{\bar{A}}_{l_{\mathrm{off}}}\right)$, where $l_{\mathrm{off}} = p_{\mathrm{s}}.p_{\mathrm{c}}.N_{\mathrm{clon}} + p_{\mathrm{s}}.p_{\mathrm{r}}$. For a given antibody $\bar{\bar{A}}_{\mathrm{m}} = \{x_1, x_2, \ldots, x_j, \ldots, x_{\mathrm{n}}\}$, if a gene x_j is chosen randomly based on the mutation rate (p_{m}) for mutation, the newly generated offspring is given as $\bar{\bar{A}}_{\mathrm{m}} = \left\{x_1, x_2, \ldots, x_j^{'}, \ldots, x_{\mathrm{n}}\right\}$, where $x_j^{'}$ is given by Eq. 1. The value of p_{m} can be fixed or adaptive. In the fixed case [27], the value of p_{m} depends on the size of the unknown variables ($p_{\mathrm{m}} = 1/n$), while in the adaptive case [28], the value of p_{m} is changed adaptively based on the value of the fitness function as illustrated by Eq. 2, where $\mathrm{Fit}(x) = 1/F(x)$ is the fitness value of the antibody, $\mathrm{Fit}_{\mathrm{avg}}$ is the average fitness value of the population, and $\mathrm{Fit}_{\mathrm{max}}$ is the maximum fitness value of the population. The values of k_1 and k_2 are in the range [0, 1] and are selected to prevent the MOIA from getting stuck at a local optimum.

$$p_{\rm m} = \begin{cases} k_1\left(\frac{{\rm Fit}_{\max}-{\rm Fit}(A_{\rm m})}{{\rm Fit}_{\max}-{\rm Fit}_{\rm avg}}\right); & \text{if } {\rm Fit}(A_{\rm m}) \geq {\rm Fit}_{\rm avg} \\ k_2 & \text{if } {\rm Fit}(A_{\rm m}) < {\rm Fit}_{\rm avg} \end{cases} \quad (2)$$

(g) **Population Updating**

A population is updated at the end of completing one generation. Based on the objective function, the fittest $p_{\rm s}$ antibodies are selected from the initial population $\left({\rm pop}({\rm gen}) = \left(A_1; A_2; \ldots, A_{\rm m}; \ldots; A_{p_{\rm s}}\right)\right)$ and the offspring population $\left({\rm pop\_off}({\rm gen}) = \left(\bar{\bar{A}}_1; \bar{\bar{A}}_2; \ldots; \bar{\bar{A}}_{\rm n}; \ldots; \bar{\bar{A}}_{l_{\rm off}}\right)\right)$ to prepare the population of the next generation $\left({\rm pop}({\rm gen}+1) = \left(A_1; A_2; \ldots, A_{\rm m}; \ldots; A_{p_{\rm s}}\right)\right)$; where $A_{\rm m}$ is given by:

$$A_{\rm m} = \begin{cases} \bar{\bar{A}}_{\rm n}, \forall n \in [1 l_{\rm off}], & \text{if} F\left(\bar{\bar{A}}_{\rm n}\right) < F(A_{\rm m}) \\ A_{\rm m}, & \text{otherwise} \end{cases}; m = 1, 2, \ldots, p_{\rm s} \quad (3)$$

(h) **Stopping Criterion**

The optimal solutions (pop*) are obtained when F(pop) does not update for a certain number of generations (gen$_{\rm cert}$) or when the generations exceed the maximum generations (Max$_{\rm gen}$) as illustrated in Eq. 4.

$$\text{pop}^* = \begin{cases} \text{pop(gen), if}\left(\sum_{g=0}^{\text{gen}_{\text{cert}}} (F(\text{pop}(\text{gen}-g)) - F(\text{pop}(\text{gen}-g-1))) < \varepsilon\right) \wedge \left(\text{gen} < \text{Max}_{\text{gen}}\right) \\ \text{pop}\left(\text{Max}_{\text{gen}}\right), \text{ ifgen} = \text{Max}_{\text{gen}} \end{cases} \quad (4)$$

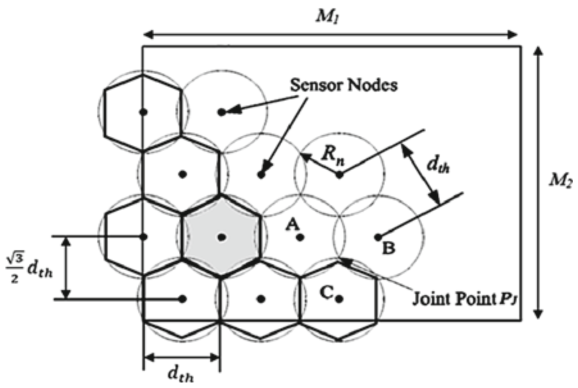
3 Optimal Deployment

The number of the sensor nodes and the distance between the adjacent sensor nodes affect the coverage of the network.

3.1 Adjacent Distance

In the case of homogeneous sensor nodes with no obstacles, the maximum coverage can be achieved by deploying the nodes regularly in the field with keeping the adjacent distance by $d_{\rm th}$ [29, 30] as shown in Fig. 2, where $d_{\rm th}$ is calculated depending on the radius of the sensor node ($R_{\rm n}$) as illustrated by Eq. 5. The value of $R_{\rm n}$ is based on the used sensing model for the senor node.

Fig. 2 Optimal deployment



$$d_{\text{th}} = \sqrt{3} \times R_{\text{n}} \tag{5}$$

(1) **Binary Model-based Adjacent Distance**

In the binary model, a sensor s can detect the target T inside its sensing range with a probability 1, and it is not able to detect any target that is outside of its sensing range. The possibility that the target T located at coordinates $(x_{\text{t}}, y_{\text{t}})$ is covered by a sensor $s_i(x_i, y_i)$ is described by [1–3, 10–19]:

$$P(x_t, y_t, s_i) = \begin{cases} 1, & \text{if} \exists i \in \{1, \ldots, N\}, d(s_i, T) \le R_{s_i} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $d(s_i, T) = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}$ is the Euclidean distance between the sensor and the target. In this model, the value of R_{n} equals the value of the sensing radius (R_{s}). Thus, d_{th} is calculated as:

$$d_{\text{th}} = \sqrt{3} \times R_{\text{s}} \tag{7}$$

(2) **Probabilistic Model-based Adjacent Distance**

The probability model is a realistic sensing model, where the detection error range (r_{e_i}) is introduced to measure the uncertainty detection of a sensor. The covered possibility of the target T by a sensor $s_i(x_i, y_i)$ using the probabilistic model is given by [20–22]:

$$P(x_t, y_t, s_i) = \begin{cases} 1, & \text{if}\, d(s_i, T) \le R_{\text{s}_i} - r_{\text{e}_i} \\ \exp(-a_1 \lambda^{a_2}), & \text{if}\, R_{\text{s}_i} - r_{\text{e}_i} < d(s_i, T) < R_{\text{s}_i} + r_{\text{e}_i} \\ 0, & \text{if}\, d(s_i, T) \ge R_{\text{s}_i} + r_{\text{e}_i} \end{cases} \tag{8}$$

where a_1, a_2, and $\lambda = r_{\text{e}_i} - R_{\text{s}_i} + d(s_i, T)$ are the parameters of the coverage model. Assume the joint point P_{J} of three nodes A, B, and C has a detection probability of

P_{th} as shown in Fig. 2. The covered probability of P_{J} is calculated as:

$$P_{\mathrm{J}} = 1 - \prod_{i=1}^{3}(1 - P(x, y, s_i)) = P_{\mathrm{th}} \tag{9}$$

Since the nodes A, B, and C are located at the same distances of P_{J}, the value of $P(x, y, s_i)$ at the three nodes is the same. Therefore, the formula in (9) is simplified to:

$$P(x, y, s_i) = 1 - \sqrt[3]{1 - P_{\mathrm{th}}} \tag{10}$$

From Eqs. 9 and 10, we get the R_{n} of each sensor as follows:

$$R_{\mathrm{n}} = R_{\mathrm{s}} - r_{\mathrm{e}} + \left(\frac{-1}{a_1}\ln\left(1 - \sqrt[3]{1 - P_{\mathrm{th}}}\right)\right)^{\frac{1}{a_2}} \tag{11}$$

Thus, d_{th} can be calculated by:

$$d_{\mathrm{th}} = \sqrt{3}\left(R_{\mathrm{s}} - r_{\mathrm{e}} + \left(\frac{-1}{a_1}\ln\left(1 - \sqrt[3]{1 - P_{\mathrm{th}}}\right)\right)^{\frac{1}{a_2}}\right) \tag{12}$$

3.2 Number of Sensor Nodes

The number of sensor nodes is the second important factor that affects the coverage of the sensor field. The optimal number of sensor nodes that cover a $M_1 \times M_2$ sensor field as illustrated in Fig. 2 is given by:

$$N_{\mathrm{opt}} = \left\lceil \frac{M_1}{d_{\mathrm{th}}} \right\rceil \left\lceil \frac{M_2}{0.5\sqrt{3}d_{\mathrm{th}}} \right\rceil \tag{13}$$

where the brackets $\lceil x \rceil$ are used for rounding x to the nearest integer number greater than or equal to x. Since the coverage of the sensor node can be represented as the area of a hexagonal shape which is given by Eq. 14, the total network coverage and the coverage ratio can be calculated by Eqs. 15 and 16, respectively, using N_{opt} sensors.

$$A_{\mathrm{s\text{-}cov}} = 6(0.5d_{\mathrm{th}} \times 0.5R) = \frac{\sqrt{3}}{2}d_{\mathrm{th}}^2 \tag{14}$$

$$A_{\mathrm{opt-cov}} = N_{\mathrm{opt}}A_{\mathrm{s\text{-}cov}} = \frac{\sqrt{3}}{2}\frac{M_1}{d_{\mathrm{th}}}\frac{M_2}{0.5\sqrt{3}d_{\mathrm{th}}}d_{\mathrm{th}}^2 \tag{15}$$

$$R_{\text{opt-cov}} = \frac{A_{\text{opt-cov}}}{M_1 \times M_2} \tag{16}$$

In the above analysis, it assumed a regular field with no obstacles. However, the obstacle case is a complicated problem.

4 First Algorithm: Immune-Based Node Deployment Algorithm

In this section, the details of the first deployment algorithm called immune-based node deployment algorithm (INDA) [31] are explained.

4.1 Problem Formulation

The problem is how to redeploy the sensor nodes in the field subject to optimizing the energy consumption and network coverage. Let the size of the sensor field be $M_1 \times M_2$ containing N sensor nodes ($S = \{s_1, s_2, \ldots, s_i, \ldots, s_N\}$) with the same sensing radius (R_s). To formulate the coverage–energy problem, the following assumptions about the sensor nodes are fixed:

(a) The coverage of each sensor is a circle with a radius of R_s.
(b) To guarantee network connectivity, the minimum communication range (R_c) of each node is adjusted by $2R_s$ ($R_c \geq 2R_s$) [32, 33].
(c) Each sensor node uses the localization algorithms [34, 35] to be aware of its location.
(d) Sensor nodes have the ability to detect obstacles.

Therefore, the problem is how to redeploy the sensor nodes by considering the obstacles and the boundaries of the sensor field to optimize the trade-off between the coverage area and the mobility cost. This problem can be described as a multi-objective problem (MOP). This problem can be solved using the MOIA by determining the optimal positions of the sensor nodes ($P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$) subject to the following objectives:

Objective 1: Maximizing Network Coverage

Let the field be partitioned into equal ($g_1 \times g_2$) grids as illustrated in Fig. 3. The coverage of the field is proportional to the number of covered grid points. The possibility that the grid point $G(x, y)$ can be covered by a sensor $s_i(x_i, y_i)$ is calculated as follows:

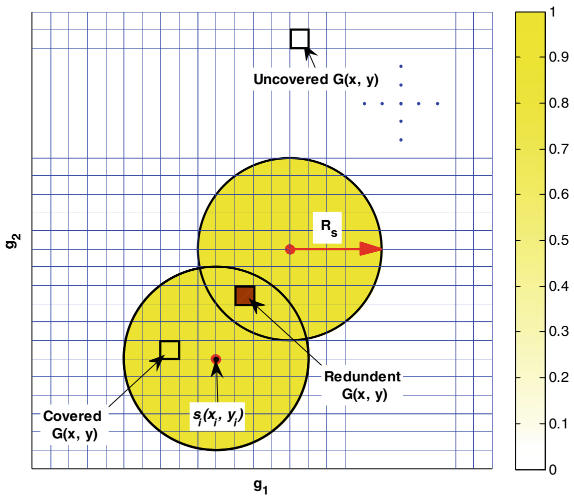


Fig. 3 Calculation of network coverage

$$P(x, y, s_i) = \begin{cases} 1, \text{if } \exists i \in \{1, \ldots, N\}, d(s_i, G) \leq R_s \\ 0, \text{ otherwise} \end{cases} \tag{17}$$

where $d(s_i, G) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the distance between the sensor $s_i(x_i, y_i)$ and grid $G(x, y)$. The probability that $G(x, y)$ is covered by N sensor nodes is given by:

$$P(x, y, S) = 1 - \prod_{i=1}^{N} (1 - P(x, y, s_i)) \tag{18}$$

It should be pointed out that the area covered by each sensor is $A_{s_i} = \pi R_{s_i}^2$. So, the maximum coverage ratio is given by:

$$R_{\text{max-cov}}(S) = \frac{A_{\max}(S)}{A_{\text{tot}}} = \frac{U_{i=1}^{N} A_{S_i}}{g_1 \times g_2} \tag{19}$$

where $A_{\max}(S)$ is the maximum coverage area and A_{tot} is the area of the field. The grid $G(x, y)$ is covered if it has a probability greater than or equal to P_{th}. Therefore, the total covered area and the coverage ratio are given by Eqs. 20 and 21, respectively.

$$A_{\text{Cov}}(S) = \sum_{x=1}^{g_1} \sum_{y=1}^{g_2} P(x, y, S) | \forall P(x, y, S) \geq P_{\text{th}} \tag{20}$$

$$R_{\text{Cov}}(S) = A_{\text{Cov}}(S) / A_{\text{tot}} \tag{21}$$

The probability ($P_{\text{red}}(x, y, S)$) that a grid point *G(x, y)* is covered by more than one sensor node and the rate of the redundant covered area ($R_{\text{red}}(S)$) as shown in Fig. 3 are given as follows:

$$P_{\text{red}}(x, y, S) = \begin{cases} 1, \text{ if}\left(\sum_{i=1}^{N}(P(x, y, s_i)|\forall \text{P}(x, y, s_i) \geq P_{\text{th}})\right) > 1, \\ 0, \text{ otherwise} \end{cases} \quad (22)$$

$$R_{\text{red}}(S) = \frac{\sum_{x=1}^{g_1}\sum_{y=1}^{g_2} P_{\text{red}}(x, y, S)}{A_{\text{tot}}} \quad (23)$$

Therefore, the network area coverage can be maximized by minimizing the uncovered area ratio as follows:

$$\text{minimize}(R_{\text{Uncov}}(P) = 1 - R_{\text{Cov}}(P)) \quad (24)$$

Objective 2: Minimizing Mobility Cost

The moved distance ($d_{\text{mov}}(i)$) of a sensor s_i is calculated from the initial location (x_{int_i}, y_{int_i}) to the final location (x_{f_i}, y_{f_i}). The consumption energy of mobility (E_{mov}) for a sensor s_i proportionates linearly with the moved distance when acceleration is negligible [36] as follows:

$$E_{\text{mov}}(s_i) = k_{\text{mov}} d_{\text{mov}}(i) \quad (25)$$

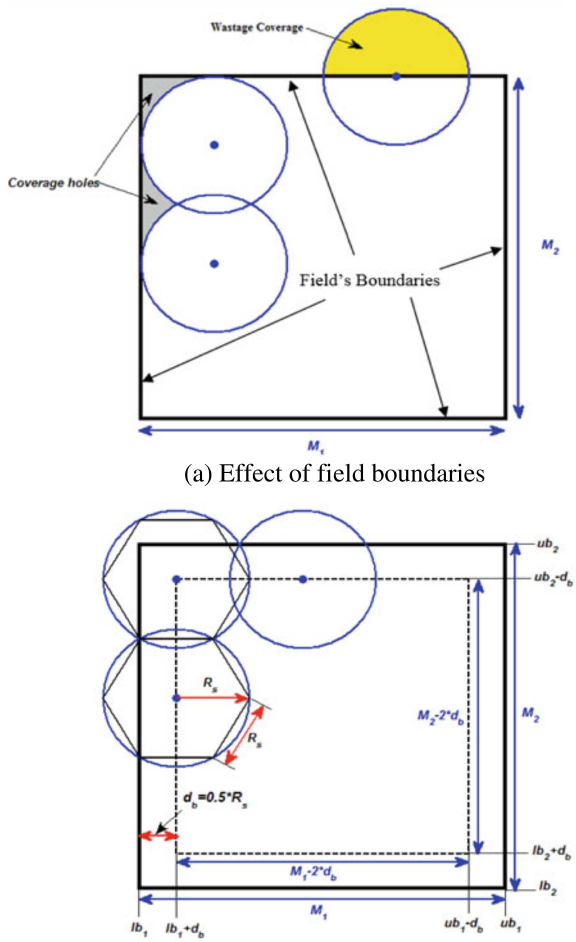
where k_{mov} is a rate of energy consumption. Thus, minimizing the mobility cost can be done by controlling the mobility of each sensor to be within its R_{c} to ensure the connectivity as:

$$\text{minimize}\left(E_{\text{dis}_{\text{mov}}}(P) = \left(\sum_{i=1}^{N} d_{\text{mov}}(i)/N\right)/R_{\text{c}}\right) \quad (26)$$

Objective 3: Obstacles and Boundary Effect

The close sensors to the field's boundaries can waste their coverage or produce coverage holes as illustrated in Fig. 4a. The effect of the boundaries should be considered during the rearrangement process of the sensor nodes. Therefore, the sensor nodes should be located on d_{b} from the boundaries of the field to optimize between the coverage holes and the number of sensor nodes as illustrated in Fig. 4b. This means that the search space of the nodes' positions should be $[lb_1 + d_{\text{b}} ub_1 - d_{\text{b}}] \times [lb_2 + d_{\text{b}} ub_2 - d_{\text{b}}]$ instead of $[lb_1 ub_1] \times [lb_2 ub_2]$, where ub_1 and lb_1 are the upper and lower horizontal boundaries of the field, respectively, and ub_2 and lb_2 are the upper and lower vertical boundaries of the field, respectively. d_{b} is the margin distance, and its value is calculated as follows:

Fig. 4 Effect of the field boundaries



(a) Effect of field boundaries

(b) Search space of the nodes' positions

$$d_{\mathrm{b}} = 0.5 \times R_{\mathrm{s}} \tag{27}$$

On the other hand, if the sensor field contains some obstacles, these obstacles block the sensing of the sensor nodes and restrict the moving of the nodes. Therefore, the following rules should be considered during the relocation process of the sensor nodes.

Rule 1: A grid point is not covered if an obstacle is placed between it and a sensor node as shown on the left side of Fig. 5.

Rule 2: If the obstacle blocks the path between the initial and the final locations of a sensor, the sensor should follow the short path around the obstacle as shown on the right side of Fig. 5.

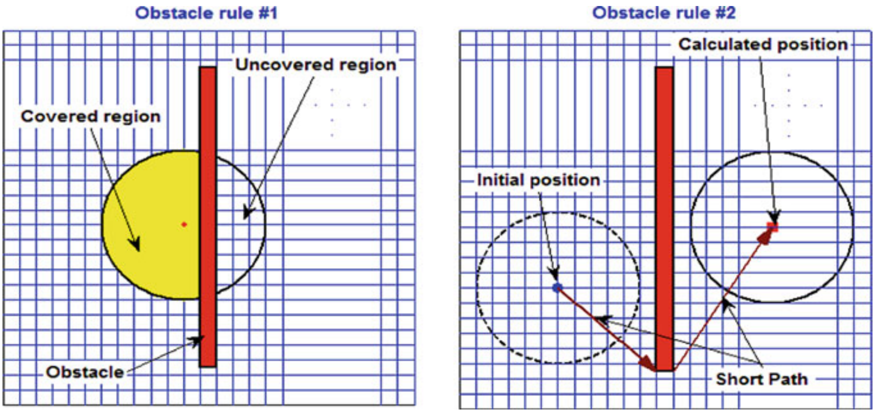


Fig. 5 Obstacle rules

As a result, the deployment problem can be solved by minimizing the sum weighted of the above-mentioned objectives as follows:

$$\begin{aligned} &\text{minimize}\big(F_{\text{INDA}}(P) = wR_{\text{Uncov}}(P) + (1-w)E_{\text{dis}_{\text{mov}}}(P)\big) \\ &\text{subject to } (lb_1 + d_{\text{b}}) \le x_i \le (ub1 - d_b) \text{ and} \\ &(lb_2 + d_{\text{b}}) \le y_i \le (ub2 - d_{\text{b}});\ i = 1, 2, \ldots, N \end{aligned} \tag{28}$$

4.2 Immune-Based Node Deployment Algorithm

The immune-based node deployment algorithm (INDA) utilizes the MOIA to solve the deployment problem by optimizing the network coverage and the mobility cost of the sensor nodes. Initially, a Hello_Msg $(ID_{\text{BS}}, x_{\text{BS}}, y_{\text{BS}})$ message is broadcasted from the base station (BS) to collect information of all sensor nodes in the network using the flooding method as shown in Fig. 6. Each node received the hello message will reply by an Info_Msg (ID, x, y) message and contains its initial location and ID, to BS. Depending on the gathered information, BS produces a population of p_{s} antibodies, pop(gen) $= \big(A_1; A_2; \ldots; A_{\text{m}}; \ldots; A_{p_{\text{s}}}\big)$, randomly to represent the positions (P) of the sensor nodes. Each antibody (A_{m}) consists of $2N$ floating-point numbers which represent the coordinates of the N nodes as shown in Table 2. Then, BS applies the steps of the MOIA to optimize the trade-off between the network coverage and mobility cost by minimizing the objective function given by Eq. 28. Finally, BS sends the determined locations to the sensor nodes.

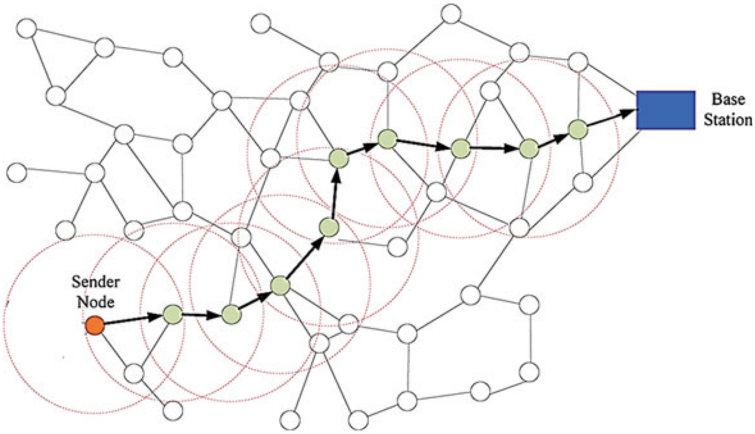


Fig. 6 Information gathering using the flooding method

Table 2 Antibody representation

	s_1		s_2		...	s_N	
Positions (P)	x_1	y_1	x_2	y_2	...	x_N	y_N
Location antibody (A)	L_1	L_2	L_3	L_4	...	L_{2N-1}	L_{2N}

5 Second Algorithm: Centralized Voronoi-Based Immune Deployment Algorithm

The INDA, which is presented in the previous section, neglected the dissipated energy in the sensing and the redundant data. Therefore, this section presents an improved version of the INDA based on utilizing the MOIA and the Voronoi diagram. The improved algorithm is called a centralized Voronoi-based immune deployment algorithm (CVIDA) [37]. In the CVIDA, the consumed energies in the sensing, the mobility, and the redundant coverage are considered in the objective function besides the network coverage. CVIDA relocates the sensor nodes to improve the coverage by utilizing an optimal number of the nodes based on reducing the mobility, adjusting the sensing range, and controlling the radio (i.e., active/sleep) of each node.

5.1 Problem Formulation

The problem is how to rearrange the sensor nodes to improve the network coverage and save the residual energy of sensor nodes. In order to solve this problem, let a sensor field with size $M_1 \times M_2$ contain N sensor nodes ($S = \{s_1, s_2, \ldots, s_i, \ldots, s_N\}$)

with the sensing radius set ($R_s = \{R_{s1}, R_{s2}, \ldots, R_{si}, \ldots, R_{sN}\}$), where $R_{si} \in \left[R_{s_{\min}} R_{s_{\max}}\right]$. The following assumptions about the sensor nodes are fixed:

(a) All sensor nodes can control their sensing range to be within the range $\left[R_{s_{\min}} R_{s_{\max}}\right]$.
(b) To guarantee network connectivity, the communication range (R_c) of each node is adjusted at least by $2R_s$ ($R_c \geq 2R_s$) [32, 33].
(c) Each sensor node uses the localization algorithms [34, 35] to be aware of its location.
(d) Sensor nodes have the ability to detect obstacles.

Since the sensor nodes consume a significant amount of energy in the redundant data and the sensing tasks, the CVIDA considers the redundant data and the sensing costs during the relocation process besides the network coverage with limiting the mobility of the sensor nodes. Therefore, the problem is how to redeploy the sensor nodes with considering the effect of the obstacles and the field's boundaries subject to maximizing the network coverage; reducing the energy cost of the mobility and the sensing processes; and reducing the redundant coverage by finding the optimal working set $\hat{S}\left(\hat{S} \subset S\right)$ of $\hat{N}$ sensor nodes. This problem will be solved in two phases using the MOIA.

In the first phase, the optimal sensors' positions ($P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$) are determined subject to:

Objective 1: Maximizing Network Coverage

$$\text{minimize}(R_{\text{Uncov}}(P) = 1 - R_{\text{Cov}}(P)) \tag{29}$$

Objective 2: Minimizing Mobility Cost

$$\text{minimize}\left(E_{\text{dis}_{\text{mov}}}(P) = \left(\sum_{i=1}^{N} d_{\text{mov}}(i)/N\right)/R_{c_{\max}}\right) \tag{30}$$

where d_{mov} is the moved distance of the sensor node and $R_{c_{\max}}$ is the allowable maximum communication range of the sensor node.

Objective 3: Minimizing Sensing Cost

The dissipated energy of a sensor (s_i) in sensing k-bits message based on the quadratic model [36, 38] by:

$$E_{\text{Sen}}(s_i) = E_{\text{sn}} k R_{s_i}^2 \tag{31}$$

where E_{sn} is the dissipated energy in sensing one bit within 1 m range. Therefore, minimizing the consumed energy in the sensing process can be done by minimizing the average of R_s^2 of all nodes relative to $R_{s_{\max}}^2$ as follows:

$$\text{minimize}\left(E_{\text{dis}_{\text{sen}}}(P) = \left(\sum_{i=1}^{N} R_{s_i}^2/N\right)/R_{s_{\max}}^2\right) \quad (32)$$

As a result, the optimum positions of the sensor nodes $(P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\})$ can be determined by optimizing the following function using the MOIA.

$$\text{minimize}\left(F_{\text{CIVA - I}}(P) = w_1 R_{\text{Uncov}}(P) + w_2 E_{\text{dis}_{\text{mov}}}(P) + (1 - w_1 - w_2) E_{\text{dis}_{\text{sen}}}(P)\right)$$

$$\text{subject to } (lb_1 + d_{\text{b}}) \le x_i \le (ub_1 - d_{\text{b}}),$$
$$(lb_2 + d_{\text{b}}) \le y_i \le (ub_2 - d_{\text{b}}) \text{ and } R_{s_{\min}} \le R_{s_i} \le R_{s_{\max}} \quad (33)$$

where w_1 and w_2 are the weight factors in the range of [0, 1].

In the second phase of the CVIDA, the subset $\hat{S}\left(\hat{S} \subset S\right)$ of $\hat{N}$ working nodes is found subject to the following objectives:

Objective 1: Maximizing Network Coverage

$$\text{minimize}\left(R_{\text{Uncov}}\left(\hat{S}\right) = 1 - R_{\text{Cov}}\left(\hat{S}\right)\right) \quad (34)$$

Objective 2: Saving Sensing Cost

This can be achieved by activating sensor nodes that have minimum sensing ranges to minimize the dissipated energy in the sensing process as follows:

$$\text{minimize}\left(E_{\text{dis}_{\text{sen}}}\left(\hat{S}\right) = \left(\sum_{i=1}^{N} b_i . R_{s_i}^2/\hat{N}\right)/R_{s_{\max}}^2\right);$$
$$\text{where } b_i = \begin{cases} 1, \text{ if node } s_i \text{ is active} \\ 0, \quad \text{otherwise} \end{cases} \quad (35)$$

Objective 3: Minimizing Financial Cost

The financial cost and the consumption energy of the redundant data can be reduced by finding the optimal working sensor nodes $\left(\hat{N}\right)$ as follows:

$$\text{minimize}\left(N_{\text{act}}\left(\hat{S}\right) = \hat{N}/N\right) \quad (36)$$

So, the set $\hat{S} = \left\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_i, \ldots, \hat{s}_{\hat{N}}\right\}$ of the working sensor nodes can be determined using the MOIA based on minimizing the weighted sum of the above three objectives as follows:

$$\text{minimize}\left(F_{\text{CIVA - II}}\left(\hat{S}\right) = w_3 R_{\text{Uncov}}\left(\hat{S}\right) + w_4 E_{\text{dis}_{\text{sen}}}\left(\hat{S}\right) + (1 - w_3 - w_4) N_{\text{act}}\left(\hat{S}\right)\right) \tag{37}$$

where w_3 and w_4 are the weight factors in the range of [0, 1]. Weight factors serve as sensitivity multipliers where they determine how sensitive the value of the cost function is relative to other variables. Thus, the appropriate values of weight factors (w_1, w_2, $w_{3,}$ and w_4) are based on the variability of each cost function.

5.2 *Centralized Voronoi-Based Immune Deployment Algorithm*

The framework of the CVIDA consists of two phases to solve the energy–coverage problem in a serialized way. The first phase is used to maximize the coverage of the network and save the cost of the mobility and sensing based on relocating the sensor nodes and adjusting their sensing ranges, while the second phase is utilized to save the redundant data via activating the sensor nodes that have a minimum sensing range with preserving the coverage at a high level. The flowchart of the CVIDA is shown in Fig. 7.

5.2.1 CVIDA: Phase I

In phase I of the CVIDA, the optimal sensors' locations ($P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$) are determined using the MOIA by minimizing the objective function ($F_{\text{CVIDA-I}}(P)$) that is given by Eq. 33. The procedure of this phase is the same as the procedure of the INDA that is described in Sect. 4 except in this phase, and the Voronoi diagram is used to control the sensing range of the nodes with considering the effect of obstacles as follows:

(1) **Sensing Range Calculation**

Voronoi diagram (VD) [30, 39] is the most generally used computational geometry method in WSNs. VD partitions the field into N subareas where a single sensor node is located in each subarea. The whole field is covered if each subarea is covered by its sensor node. Three strategies will be considered here for adjusting R_{s_i} of each node as illustrated in Fig. 8.

Strategy 1: Maximum Strategy

In strategy 1, R_{s_i} of a sensor in each subarea is set at least by the distance between the furthest subarea vertex ($V_j(x_{vj}, y_{vj})$) and a sensor $s_i(x_i, y_i)$ as illustrated by the dotted circle in Fig. 8.

$$R_{s_i} = \min\left(R_{s_{\max}}, \max\left(\max\left(dv_{ij}\right), R_{s_{\min}}\right)\right),$$

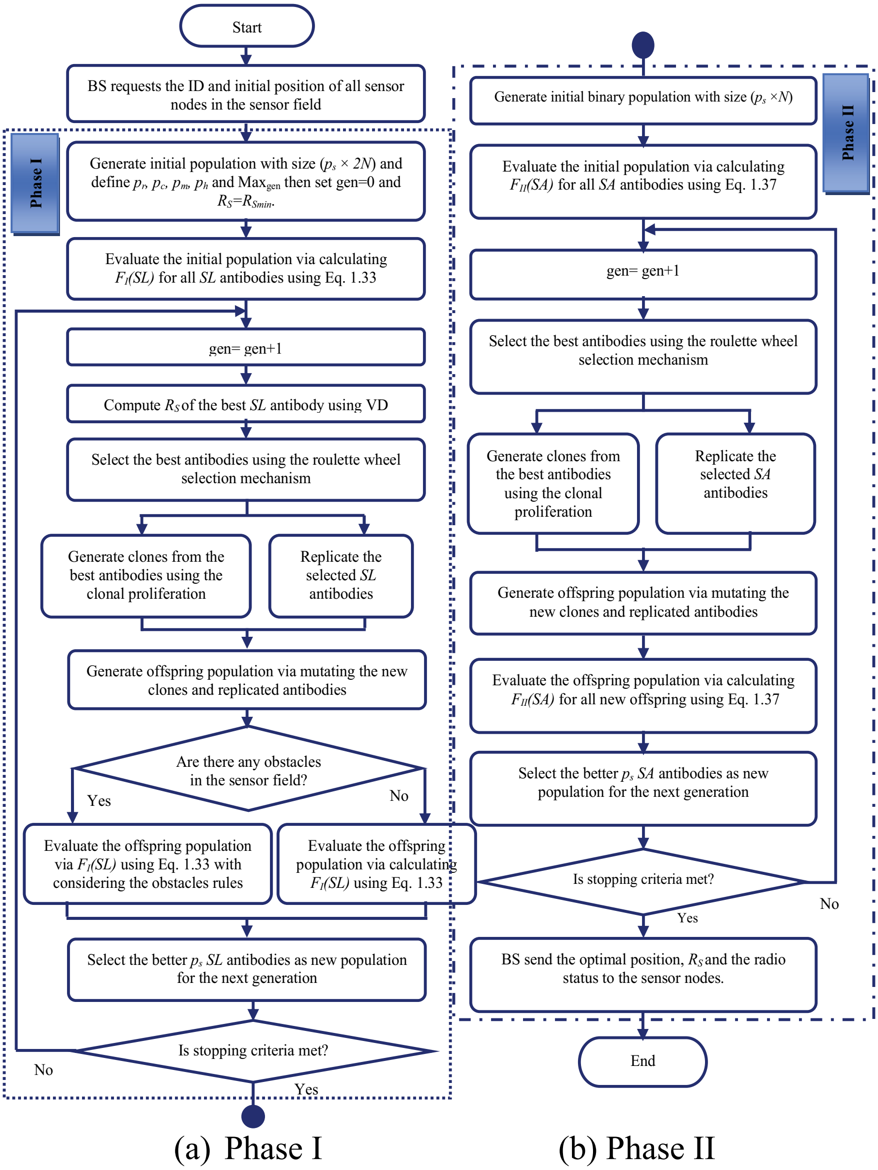
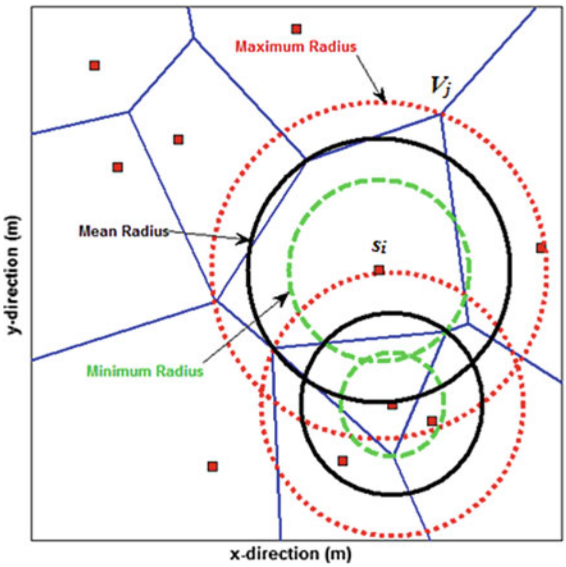


Fig. 7 Flowchart of CVIDA

Fig. 8 Strategies of calculating the sensing range using Voronoi diagram



$$\forall i = 1, 2, \ldots, N \wedge j = 1, 2, \ldots, n_v(i) \tag{38}$$

where dv_{ij} is the Euclidean distance between the subarea vertex V_j and a sensor s_i and $n_v(i)$ is the number of subarea's vertices. From Fig. 8, it is noticed that this strategy improves the coverage but the consumed energy in the sensing process is increased according to Eq. 29.

Strategy 2: Minimum Strategy

Here, R_{s_i} is adjusted depending on the distance between the nearest vertex and a sensor $\left(\min\left(dv_{ij}\right)\right)$ as given by Eq. 39. The performance of this strategy is the opposite to the performance of strategy 1. Strategy 2 saves the sensing cost, but it does not guarantee the coverage of the network as illustrated by the dashed circle in Fig. 8.

$$R_{s_i} = \min\left(R_{S_{\max}}, \max\left(\min\left(dv_{ij}\right), R_{S_{\min}}\right)\right),$$
$$\forall i = 1, 2, \ldots, N \wedge j = 1, 2, \ldots, n_v(i) \tag{39}$$

Strategy 3: Mean Strategy

The mean strategy can find the trade-off between the sensing cost and network coverage. Here, R_{s_i} of each node is calculated using the average of the distances between all vertices and a sensor $\left(\text{mean}\left(dv_{ij}\right)\right)$ as explained by the solid circle in Fig. 8.

$$R_{s_i} = \min\left(R_{S_{\max}}, \max\left(\text{mean}\left(dv_{ij}\right), R_{S_{\min}}\right)\right),$$

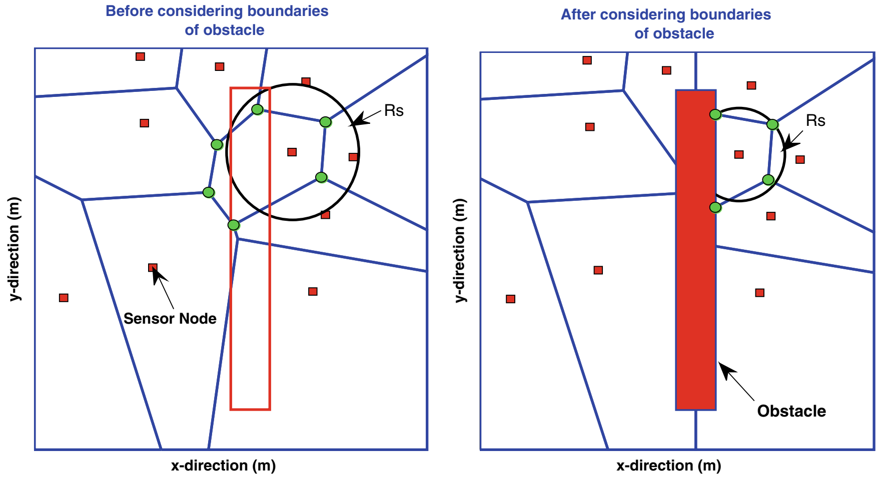


Fig. 9 Effect of obstacle on VD

$$\forall i = 1, 2, \ldots, N \wedge j = 1, 2, \ldots, n_v(i) \tag{40}$$

(2) **Effect of Obstacle on Voronoi Diagram**

In the obstacle case, the closest sensor nodes to the obstacles do not cover their subareas as shown in Fig. 9. Thus, the field's boundaries and the obstacles should be considered during using VD. This can reduce the sensing radius of a sensor and saves the sensing cost as illustrated in Fig. 9.

5.2.2 CVIDA: Phase II

In phase II of the CVIDA, BS uses the MOIA to control the radio of the sensor nodes based on minimizing the number of working nodes, saving the sensing cost, and maximizing the network coverage as shown in Fig. 7b. Firstly, BS encodes the N nodes by N bits and produces p_s antibodies as shown in Table 3, in which the active node is represented by "1" and the sleep node is represented by "0".

The optimal working sensor nodes that satisfy the network coverage and save the energy of sensor nodes are determined by evaluating each antibody using the function $F_{\text{CIVA-II}}(\text{SA})$ that is illustrated by Eq. 37. The antibodies with higher fitness function are more likely to be selected as the parents. The selected antibodies will

Table 3 SA antibody representation

Sensor node	s_1	s_2	s_3	s_4	…	s_N
Working sensor antibody (SA)	b_1	b_2	b_3	b_4	…	b_N
	1	0	1	1	…	0

undergo the operations of the MOIA for producing offspring population by mutating each bit from 0 to 1 or vice versa.

6 Experimental Results

Many experiments and simulations based on the binary and the probabilistic models are conducted to validate the INDA and CVIDA and to compare their performance with the state-of-the-art algorithms [10, 11, 15, 20, 30]. Each simulation was conducted for 20 independent runs, and the average of the obtained results is taken as the final results. The parameters of the MOIA are set as $p_s = 30$, $p_c = 0.05$, $p_h = 0.7$, $p_r = 0.8$, $N_{clon} = 5$, and $Max_{gen} = 100$.

6.1 Binary Model-Based Simulations

Experiment 1: Sparse Network

This experiment studies the effect of varying the density of the nodes within sparse networks on the performance of the proposed algorithms compared to the PSO_Voronoi [10] and $WSNPSO_{con}$ [15] algorithms. This can be done by considering six networks with different node densities, namely 10, 20, 30, 60, 80, and 100 nodes as explained in Table 4. In this experiment, a binary model with $R_s \in [27]$ m is utilized and the weight factors of the objective functions are set as $w_1 = 0.65$, $w_2 = 0.2$, $w_3 = 0.8$, and $w_4 = 0.1$. The calculated coverage $R_{opt\text{-}cov}$, using Eq. 16, and obtained coverage ratios (R_{Cov}) using the proposed and other algorithms are shown in Fig. 10. Moreover, the maximum (d_{max}) and the average (d_{avg}) moved distances, the mobility cost ($M_{cost} = E_{mov}/k_{mov}$) of the network, $R_{S\text{-}avg}$, and the saved sensing cost E_{sen} are listed in Table 5.

Compared to [10], the CVIDA enhances R_{Cov} of networks I, IV, V, and VI by 3.2%, 1.6%, 1.5%, and 0.77%, respectively, while the other two networks II and III achieved approximately the same coverages as in [10]. However, the CVIDA saves the mobility cost compared to the algorithm PSO_Voronoi in [10] as illustrated by

Table 4 Network specification of experiment 1 (binary model)

Network	Field size	N
I	50 × 50	10
II		20
III		30
IV	100 × 100	60
V		80
VI		100

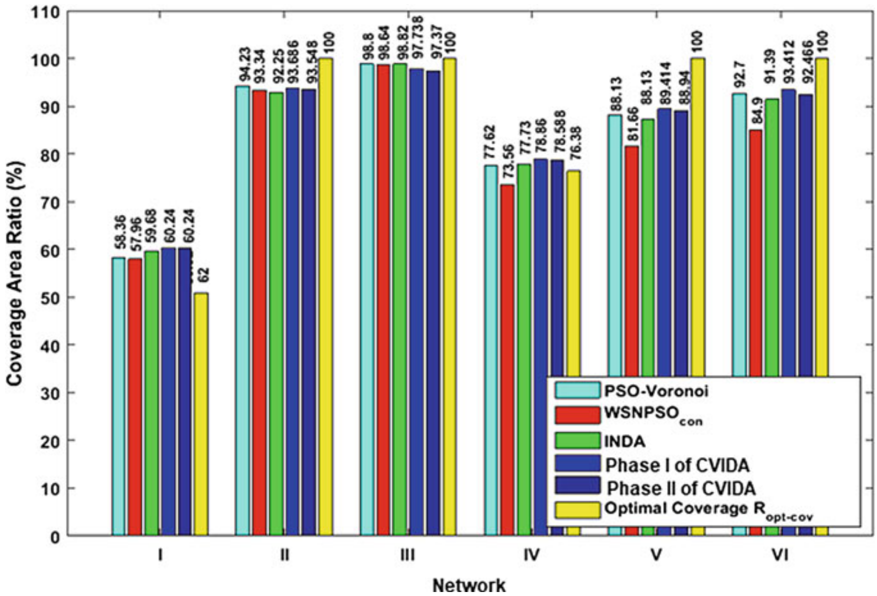


Fig. 10 Coverage area ratios of experiment 1

the values of the $d_{\max}$ in Table 5. Compared to the $\text{WSNPSO}_{\text{con}}$ algorithm [15], the CVIDA enhances the coverage ratios of all networks and saves the mobility cost of the network. For example, R_{Cov} of network VI is improved by 10% with improving $d_{\max}$ by 5.9%. The values of d_{avg} obtained using the INDA are better than that obtained using the CVIDA because it considers only the coverage and the mobility cost in its objective function. However, the CVIDA outperforms the INDA in terms of the coverage (R_{Cov}), the number of the active nodes ($\hat{N}$), and the sensing range ($R_{\text{S-avg}}$). Moreover, the CVIDA has the ability to save the sensing cost (E_{sen}) of the network as illustrated in Table 5. The initial and final network coverages and Voronoi shapes of network VI obtained using the CVIDA are shown in Fig. 11. Increasing the number of sensor nodes in the network leads to increase the redundant coverage. Deactivating of these redundant sensor nodes has no a significant effect on the coverage of the network R_{Cov} as shown in Fig. 10. The proposed algorithms have fast convergence and small execution time (T_{exe} sec) compared to the other algorithms.

Experiment 2: Dense Network

In this experiment, a dense network of size $150 \times 150\ \text{m}^2$ is considered with varying the deployed nodes from 110 to 200 by step 10. Each sensor node has a value of $R_s \in [823]$ m. For a fair comparison with the NSGA-II algorithm [11] that assumed all nodes are static, we consider only the phase II of the CVIDA and the weight factors w_3 and w_4 are set by 0.95 and 0.8, respectively. The obtained R_{Cov} and the sensing cost per area ($\sum_{i=1}^{N} R_{s_i}^2 / A_{\text{tot}}$) using the two algorithms at the different deployed nodes are shown in Fig. 12. It is observed that the obtained coverage using the CVIDA

Table 5 Obtained results of experiment 1

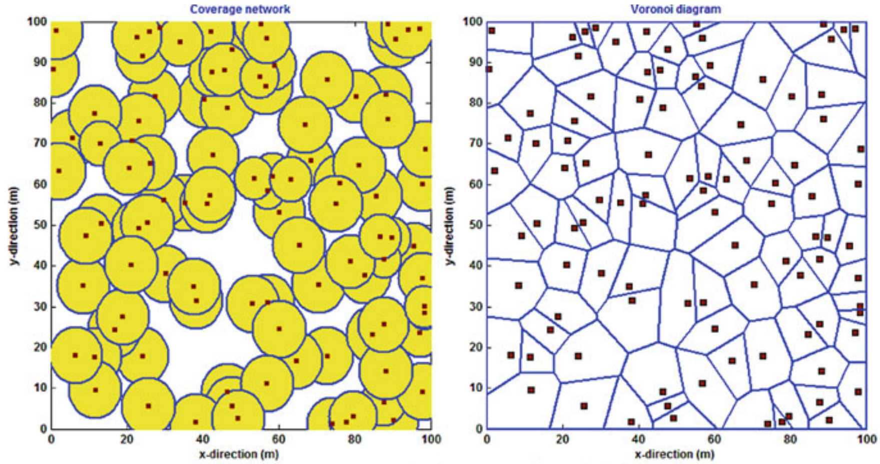
Network	Algorithm	d_{avg} (m)	d_{max} (m)	M_{cost}	$R_{s\text{-}avg}$ (m)	E^{a}_{sen} (%)	$\widehat{N}^{a}$	T_{exe} (s)	N^{a}_{opt}
I	PSO_Voronoi	–	26.59	26.59	7	0	10	14.42	25
	$WSNPSO_{con}$	–	19.29	19.29	7	0	10	15.96	
	INDA	3.18	8.626	8.626	7	0	10	7.82	
	CVIDA	3.14	8.995	8.995	7	0	10	8.27	
II	PSO Voronoi	–	28.38	28.38	7	0	20	28.56	
	WSNPSOcon	–	18.9	18.9	7	0	20	30.08	
	INDA	3.72	19.91	19.91	7	0	20	14.48	
	CVIDA	4.02	14.276	14.27	6.93	2.05	19.7	17.15	
III	PSO_Voronoi	–	24.84	24.84	7	0	30	44.19	
	WSNPSOcon	–	19.18	19.18	7	0	30	46.48	
	INDA	1.96	15.35	15.35	7	0	30	25.125	
	CVIDA	2.62	13.299	13.29	6.23	20.72	27.6	30.42	
IV	PSO_Voronoi	–	43.43	43.43	7	0	80	112.87	90
	$WSNPSO_{con}$	–	19.65	19.65	7	0	80	114.77	
	INDA	2.18	19.319	19.31	7	0	80	71.53	
	CVIDA	3.1 9	18.677	18.67	6.98	0.69	59.1	80.65	
V	PSO_Voronoi	–	38.31	38.31	7	0	90	174.92	
	$WSNPSO_{con}$	–	19.79	19.79	7	0	90	176.77	
	INDA	1.70	18.927	18.92	7	0	90	101.68	
	CVIDA	2.95	18.809	18.80	6.82	5.04	75.75	137. 75	
VI	PSO_Voronoi	–	42.51	42.51	7	0	100	247.45	
	WSNPSO_con	–	19.77	19.77	7	0	100	250.46	
	INDA	1.69	21.757	21.75	7	0	100	176.81	
	CVIDA	2.03	16.622	16.62	6.56	12.08	89	201.43	

[a] $\widehat{N}$ is the simulated working nodes; N_{opt} is the analytical optimal working nodes and $E_{sen} = (R^2_{smax} - R^2_{s-max})/R^2_{smax}$ is the saved sensing cost

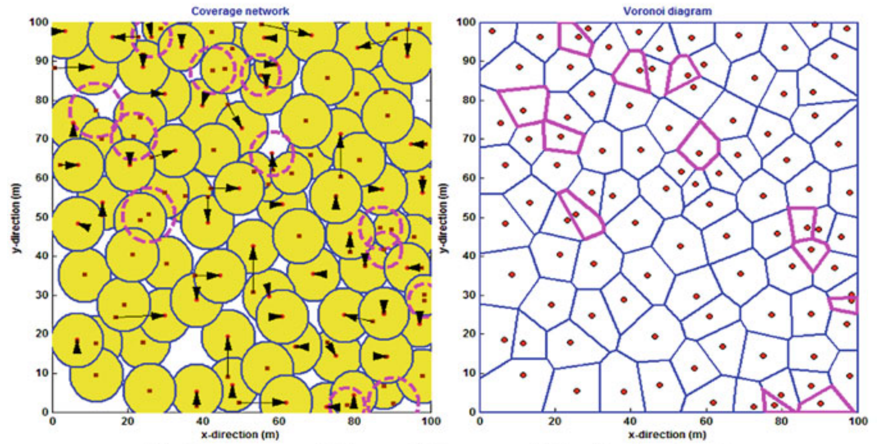
is high and approximately constant in different cases of deployed nodes. Moreover, the CVIDA reduces the sensing cost compared to the NSGA-II algorithm for the different nodes' density. The performance of the NSGA-II algorithm is improved as the node density increases, while the CVIDA works well at all values of the node density.

Experiment 3: **Network with Obstacles**

Here, we study the CVIDA's performance compared to the distributed algorithm [30] for the obstacle network. A 100 × 100 m^2 network has two obstacles A and B with a size of 5 × 20 m^2 as shown in Fig. 13a. Twenty sensor nodes deployed randomly at the center of the network with initial coverage of 31.7% and the total theoretical coverage of 98%. Each sensor node is modeled as a binary with $R_{s_i} \in [425]$ m and



(a) Coverage and Voronoi diagram of the initial network

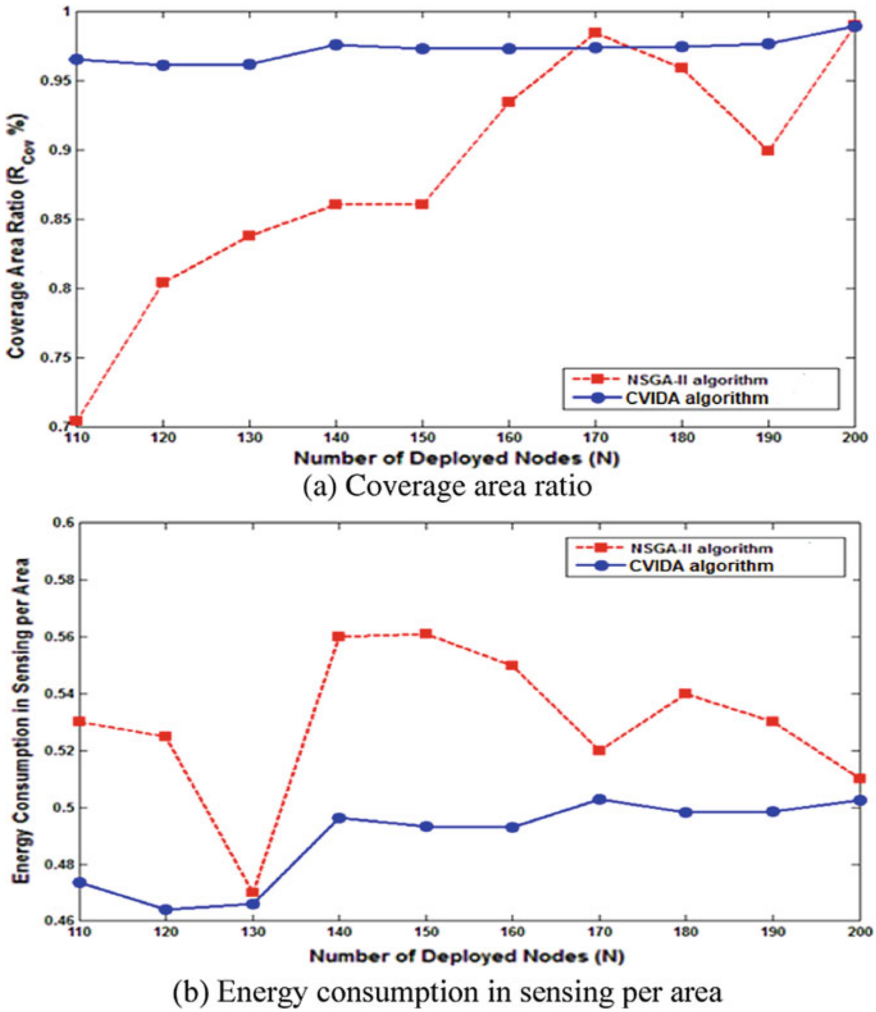


(b) Coverage and Voronoi diagram of the final network

Fig. 11 Initial and final coverage and Voronoi shape of network VI in experiment 1. Arrow represents the movement path of sensor nodes from the initial position (■) to the calculated position (●); dashed circles represent the sleep nodes

maximum communication range (R_{cmax}) of 55 m. The objective weight factors of the proposed algorithms w_1, w_2, w_3, and w_4 are set by 0.8, 0.1, 0.8, and 0.1, respectively, and Max_{gen} is set by 50. Also in this experiment, we study the performance of strategy 1 and strategy 3 of the sensing range calculation compared to no adjustment strategy (strategy 0).

Figure 13 shows the initial and final coverages and VD shapes of the CVIDA using strategy 3. Table 6 lists the obtained results of the three adjustment strategies using the two algorithms. It is observed that the CVIDA without adjustment (strategy 0) uses half of the nodes to improve the coverage by 3.8% and reduces the moved

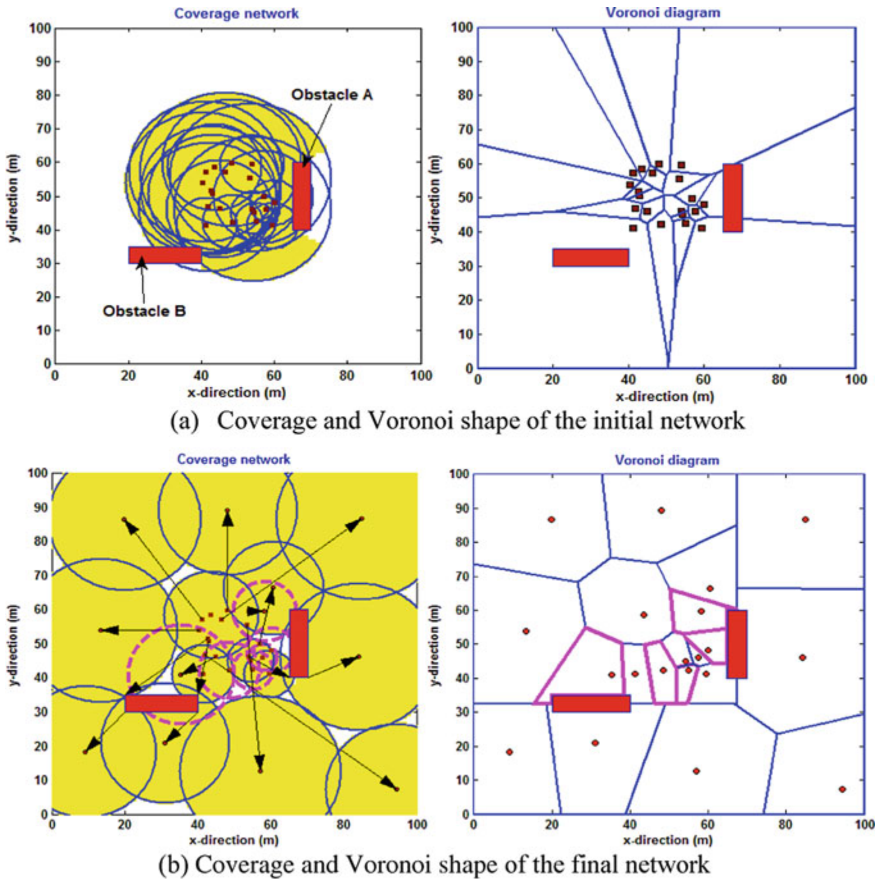


(a) Coverage area ratio

(b) Energy consumption in sensing per area

Fig. 12 Results of experiment 2

distance (d_{avg}) by 22.0% compared to the other algorithm. In strategy 1, only half of the nodes are used to give better results than the distributed algorithm. In strategy 3, 75% of the sensor nodes are activated to enhance R_{Cov} by 5.4% and save $E_{sensing}$ by 15.4% compared to the other algorithm.



(a) Coverage and Voronoi shape of the initial network

(b) Coverage and Voronoi shape of the final network

Fig. 13 Initial and final network of experiment 3

6.2 Probabilistic Model-Based Experiment

Compared to the CSAPO algorithm [20], we study the performance of the INDA and CVIDA based on the probabilistic model of a sensor node. A 50×50 m^2 network with different node density, namely 50, 60, 70, 80, and 90, is considered. Each sensor node uses the probabilistic model with $R_s \in [13]m$, $r_e = 0.6R_s$, $\alpha = 0.5$, $\beta = 0.5$ and $P_{th} = 0.85$. The weight factors w_1, w_2, w_3, and w_4 are set as 0.75, 0.15, 0.8, and 0.1, respectively, and Max_{gen} is set by 150 generations. The initial coverage, $R_{opt\text{-}cov}$, and R_{Cov} of three algorithms versus the node degree are shown in Fig. 14a. Figure 14b shows the total moving traced distances during the redeployment process versus the nodes' density.

It is seen that the CVIDA enhances R_{Cov} over the CSAPO algorithm at different node densities. Moreover, the sum of the moved distances during the rearrangement process for the CVIDA is smaller than that used on the other algorithms. The obtained

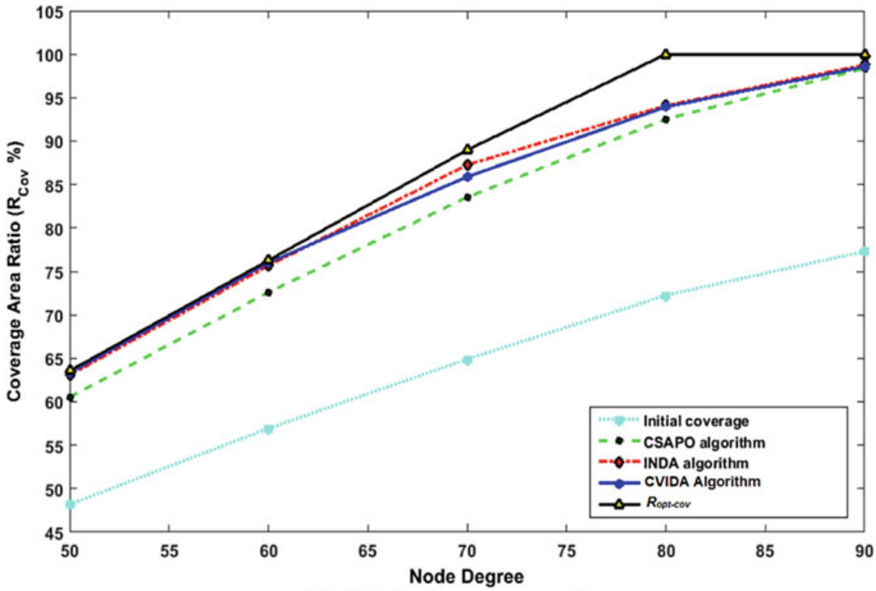
Table 6 Obtained results of experiment 3

		CVIDA	Distributed algorithm [30]
Strategy 0	R_{cov} (%)	97.54	94
	d_{avg} (m)	19.19	24.59
	M_{cost}	19.19	24.59
	$\widehat{N}$	10	20
Strategy 1	$R_{s\text{-}avg}$	19.7	19.8
	R_{cov} (%)	94.38	94
	d_{avg} (m)	26.7	–
	M_{cost}	26.7	–
	$\widehat{N}$	10	20
	$E_{sensing}$	37.9	37.3
Strategy 3	$R_{s\text{-}avg}$	14.62	16.4
	R_{cov} (%)	93.48	88.7
	d_{avg} (m)	24.08	–
	M_{cost}	24.08	–
	$\widehat{N}$	14	20
	$E_{sensing}$	65.8	57

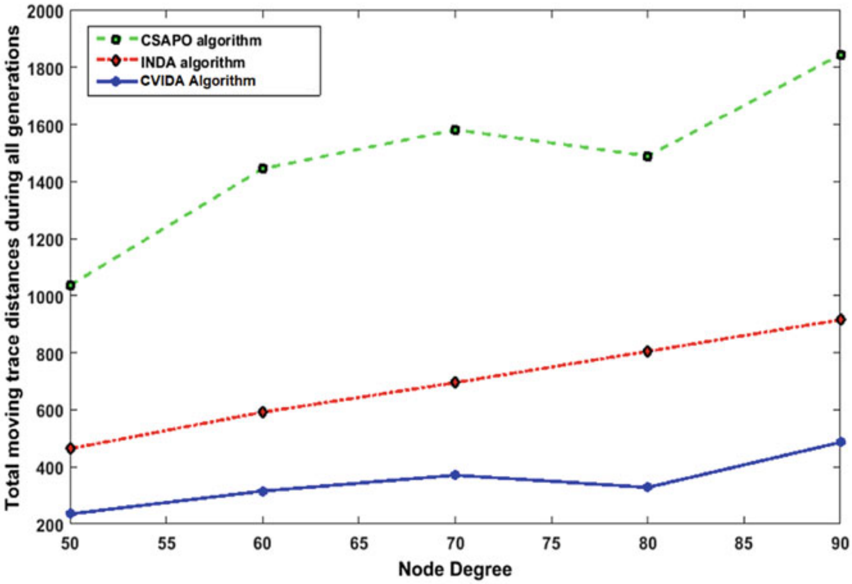
coverage ratios of the INDA are similar that are obtained using the CVIDA as clarified in Fig. 15. According to Eqs. 12 and 13, the number of required nodes to cover the entire network is 90 nodes. Therefore, the CVIDA deactivates only one node and 3 nodes for the networks with 80 and 90 nodes, respectively, while in the first three-network cases with 50, 60, and 70, all nodes are activated to cover the network. The initial and the final network coverages of the CVIDA for the case of 90 nodes are shown in Fig. 15. Based on adjusting the sensing range, the CVIDA improves $E_{sensing}$ by 0.2%, 0.9%, 3.6%, 0.6%, and 2.2% compared to other algorithms at different node densities, respectively. The obtained results cleared that the proposed algorithms have the ability to balance the trade-off between the network coverage and the energy cost using the MOIA. The proposed algorithms improve the gap of the other methods by considering the energy of sensing, mobility, and redundant coverage besides the network coverage as MOP with considering the effect of field boundaries and obstacles.

7 Conclusion

Two centralized energy-efficient deployment algorithms based on the multi-objective immune algorithm (MOIA) are presented in this chapter. The idea of the two algorithms is optimizing the trade-off between the coverage and the energy cost. The first deployment algorithm is called an immune-based node deployment algorithm



(a) Obtained coverage ratio



(b) Obtained traced moving distance

Fig. 14 Obtained results of a probabilistic model-based experiment

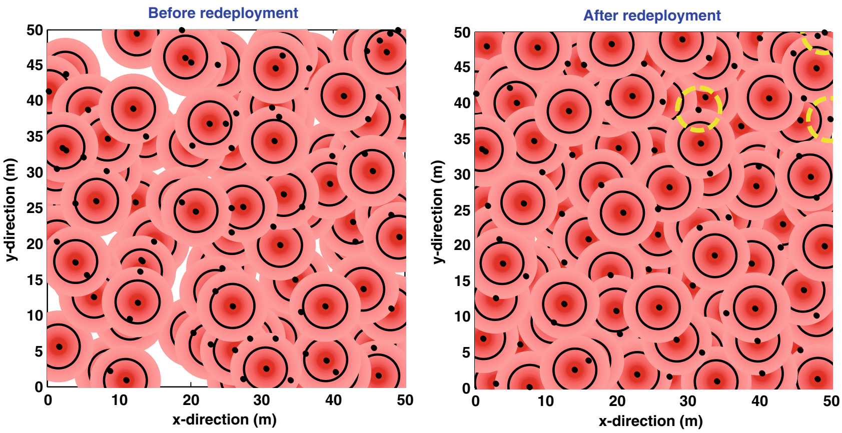


Fig. 15 Initial and final coverage for the network with 90 nodes. The black circles represent the active nodes with range R_S, and the dashed yellow circles are the sleep nodes

(INDA) and considers the mobility cost besides the network coverage. The second deployment algorithm is called a centralized Voronoi-based immune deployment algorithm (CVIDA) which utilizes the Voronoi diagram to adjust the sensing range of sensor nodes and the MOIA to control the mobility cost, the sensing range, and the communication radio of each node. The experiment results validated the performance of the proposed algorithms compared to the other algorithms. Future work should be continued to implement the developed algorithms and to consider the routing problem of wireless sensor networks.

References

1. Wang, G., Guo, L., Duan, H., Liu, L., & Wang, H. (2012). Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm. *Journal of Sensor and Actuator Networks, 1,* 86–96.
2. Qu, Y., & Georgakopoulos, S. V. (2011). Relocation of wireless sensor network nodes using a genetic algorithm. In *2011 IEEE 12th Annual Wireless and Microwave Technology Conference (WAMICON)* (pp. 1–5) FL: Clearwater Beach
3. Jin, L., Jia, J., & Sun, D. (2010). Node distribution optimization in mobile sensor network based on multi-objective differential evolution algorithm. In *2010 Fourth International Conference on Genetic and Evolutionary Computing (ICGEC)* (pp. 51–54) Shenzhen.
4. Natalizio, E., & Loscrí, V. (2013). Controlled mobility in mobile sensor networks: advantages, issues and challenges. *Telecommunication Systems, 52,* 2411–2418.
5. De, D., Mukherjee, A., Das, S. K., & Dey, N (2020). Wireless sensor network: Applications, challenges, and algorithms. In D. De, A. Mukherjee, S. Kumar Das, & N. Dey (Eds.), *Nature inspired computing for wireless sensor networks* (pp. 1–18). Singapore: Springer Singapore.
6. Fabio, F., Coello, C. A. C., & Repetto, M. (2009). Multiobjective optimization and artificial immune system: A review. In *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies* (pp 1–21). IGI Publishing.

7. Lu, H. (2009). An Adaptive multi-objective immune optimization algorithm. In *IITA International Conference on Control, Automation and Systems Engineering, CASE 2009* (pp. 140–143). Zhangjiajie
8. Sabor, N., & Abo-Zahhad, M. (2020). A Comprehensive survey of intelligent-based hierarchical routing protocols for wireless sensor networks. In D. De, A. Mukherjee, S. Kumar Das, & N. Dey (Eds.), *Nature inspired computing for wireless sensor networks* (pp. 197–257). Singapore: Springer Singapore.
9. Jyotishree, R. K. A. (2012). Blending roulette wheel selection and rank selection in genetic algorithms. *Machine Learning and Computing, 2,* 365–370.
10. Aziz, N. A. B. A., Mohemmed, A. W., & Alias, M. Y. (2009). A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram. In *International Conference on Networking, Sensing and Control* (pp. 602–607). Okayama.
11. Jia, J., Chen, J., Chang, G., Wen, Y., & Song, J (2009). Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius. *Computers & Mathematics with Applications, 57,* 1767–1775.
12. Aziz, N. A. A., Mohemmed, A. W., Alias, M. Y., Aziz, K. A., & Syahali, S. (2011). Coverage maximization and energy conservation for mobile wireless sensor networks: A two phase particle swarm optimization algorithm. In *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)* (pp. 64–69). Penang.
13. Costanzo, C., Loscrí, V., Natalizio, E., & Razafindralambo, T. (2012). Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network. *Computer Communications, 35,* 1047–1055.
14. Song, M., Yang, L., Li, W., & Gulliver, T. A. (2013). Improving wireless sensor network coverage using the VF-BBO algorithm. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 318–321). BC: Victoria.
15. NaA, Aziz. (2013). Wireless sensor networks coverage-energy algorithms based on particle swarm optimization. *Emirates Journal for Engineering Research, 18,* 41–52.
16. Banimelhem, O., Mowafi, M., & Aljoby, W. (2013). Genetic algorithm based node deployment in hybrid wireless sensor networks. *Communications and Network, 2013,* 273–279.
17. Qian, K., Shen, S., & Dai, Z. (2015). A self-adjusting node deployment algorithm for wireless sensor network. In F. Bian & Y. Xie (Eds.), *Second international conference on geo-informatics in resource management and sustainable ecosystem, GRMSE 2014* (pp. 505–512). Berlin Heidelberg, Berlin, Heidelberg: Springer.
18. Aliyu, M. S., Abdullah, A. H., Chizari, H., Sabbah, T., & Altameem, A. (2016). Coverage enhancement algorithms for distributed mobile sensors deployment in wireless sensor networks. *International Journal of Distributed Sensor Networks, 2016,* 1–9.
19. Tian, J., Gao, M., & Ge, G. (2016). Wireless sensor network node optimal coverage based on improved genetic algorithm and binary ant colony algorithm. *EURASIP Journal on Wireless Communications and Networking, 2016,* 1–11.
20. Hui ZXaLL, L. (2013). A hybrid deployment algorithm based on clonal selection and artificial physics optimization for WSN. *Information Technology, 12*, 917–925
21. Guo, Y.-N., Liu, D., Liu, Y., & Chen, M. (2013). The coverage optimization for wireless sensor networks based on quantum-inspired cultural algorithm. In *Proceedings of 2013 Chinese Intelligent Automation Conference* (pp. 87–96). Springer Berlin Heidelberg
22. Loscri, V., Natalizio, E., & Mitton, N. (2014) Performance evaluation of novel distributed coverage techniques for swarms of flying robots. In *IEEE Wireless Communications and Networking Conference (WCNC).* Turkey: Istanbul.
23. Zhao, L., Bai, G., Shen, H., & Tang, Z. (2015). Energy efficient barrier coverage in hybrid directional sensor networks. In *2015 International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1–5). Nanjing.
24. Alia, O. M., & Al-Ajouri, A. (2017). Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm. *IEEE Sensors Journal, 17,* 882–896.
25. Cao, B., Zhao, J., Lv, Z., Liu, X., Kang, X., & Yang, S. (2018). Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism. *Journal of Network and Computer Applications, 103,* 225–238.

26. Zhao, X., Gao, X.-S., & Hu, Z.-C. (2007). Evolutionary programming based on non-uniform mutation. *Applied Mathematics and Computation, 192,* 1–11.
27. Abo-Zahhad, M., Ahmed, S. M., Sabor, N., & Al-Ajlouni, A. F. (2010). The convergence speed of single- and multi-objective immune algorithm based optimization problems. *Signal Processing: An International Journal, 4,* 247–266.
28. Vafaee, F., & Nelson, P. C. (2009). A genetic algorithm that incorporates an adaptive mutation based on an evolutionary model. *International Conference on Machine Learning and Applications* (pp. 101–107). FL: Miami Beach.
29. Xueqing, W., Fayi, S., & Xiangsong, K. (2009). Research on optimal coverage problem of wireless sensor networks. In *WRI International Conference on Communications and Mobile Computing, CMC '09* (pp 548–551)
30. Qu, Y. (2013). Wireless sensor network deployment in: Faculty of engineering, electrical engineering department. *FLORIDA International University* Ph.D. thesis
31. Abo-Zahhad, M., Ahmed, S. M., Sabor, N., & Sasaki, S. (2015). Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm. *Computers & Electrical Engineering, 43,* 76–89.
32. Honghai, Z., & Jennifer, C. H. (2005) Maintaining sensing coverage and connectivity in large sensor networks. In *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks* (Auerbach Publications)
33. Ghosh, A., & Das, S. K. (2006). Coverage and connectivity issues in wireless sensor networks. In R. Shorey (Ed.), *Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions* (pp. 221–256). Wiley.
34. Wang, B. (2011). Coverage problems in sensor networks: A survey. *ACM Computing Surveys, 43,* 1–53.
35. Ghosh, A., & Das, S. K. (2008). Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing, 4,* 303–334.
36. Yipeng, Q., & Georgakopoulos, S. V. (2012). A centralized algorithm for prolonging the lifetime of wireless sensor networks using particle swarm optimization. In *IEEE 13th Annual Wireless and Microwave Technology Conference (WAMICON)* (pp. 1–6). FL: Cocoa Beach.
37. Abo-Zahhad, M., Sabor, N., Sasaki, S., & Ahmed, S. M. (2016). A centralized immune-Voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks. *Information Fusion, 30,* 36–51.
38. Dhawan, A., Vu, C. T., Zelikovsky, A., Li, Y., & Prasad, S. K (2006). Maximum lifetime of sensor networks with adjustable sensing range. In *Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* (pp. 285–289). NV: Las Vegas.
39. Dhiman, A., & Kang, S. S. (2009). Deployment of wireless sensor nodes using voronoi diagram. *Advanced Research in Computer Science and Software Engineering, 4,* 98–102.

An Efficient Routing in Wireless Sensor Network: An Application of Grey Wolf Optimization

Samiran Bera, Santosh Kumar Das, and Joydev Ghosh

1 Introduction

In the last few years, the applications of Wireless Sensor Network (WSN) increases rapidly due to its self-organizing, distributive, spatially, and low energy characteristic [1]. Figure 1 shows an illustration of WSN, where multiple sensor nodes are distributed based on some color. In this figure, three colored nodes are shown such as source, destination, and neighbor nodes and one Base Station (BS). The purpose of the source node is to send data packet to the destination node. The purpose of the destination node is to receive data packet from the source node. And the purpose of the neighbor node is to establish communication between networks and helps in link establishment from one sensor node to another sensor node. This sensor is a low capacity device which is scatters all over the workable zone or area. The purpose of this zone is to achieve specific goal of the users or customers. The main goal of the sensor nodes is to observe and sense multiple phenomenon of the environment such as sound, temperature, vibration, humidity, position, pressure, etc. [2]. These parameters are work as input parameters of the proposed work that deals in the workable areas. The stated features help to deal several applications such as crisis handling, medical systems, military handling and management, transportation, entertainment may be inside or outside of the home or offices, defense, etc. [3].

S. Bera (✉)
Indian Institute of Technology (Indian School of Mines), Dhanbad, India
e-mail: samironbera@gmail.com

S. K. Das
National Institute of Science and Technology, Behrampur, India
e-mail: sunsantosh2014@gmail.com

J. Ghosh
School of Computer Science and Robotics, National Research Tomsk Polytechnic University (TPU), Tomsk, Russia
e-mail: joydev.ghosh.ece@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_8

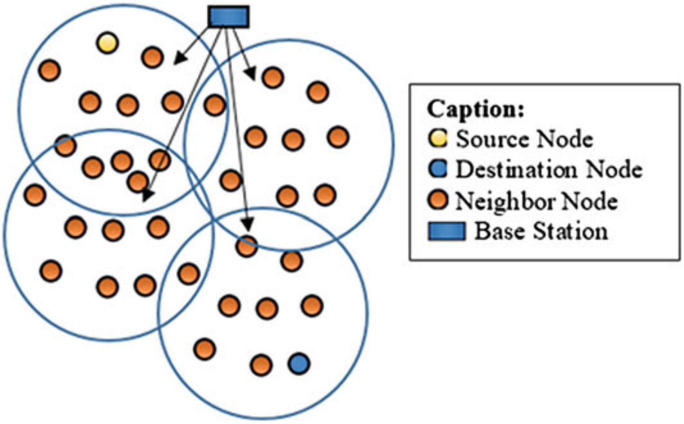


Fig. 1 Wireless sensor network

1.1 Motivation for the Study

Although, several advantages or applications of WSN. But it has also some limitations such as low bandwidth, low memory, and energy capacity of the nodes, and hazardous technique for sensing parameters in a heterogeneous environment. Due to these limitations, several issues are raised such as deployment, calibration, synchronization, localization, dissemination, and aggregation. The stated issues cause some attacks like denial, compromise, impersonation, and some other attacks that are based on protocol-specific [4, 5].

Hence, the proposed method is used to design an intelligent and efficient routing technique using a Grey Wolf Optimization technique. The main key element of the proposed method is the grey wolf nature-inspired technique. It helps to reduce the uncertainty of the network and control several network parameter variations. This technique helps to optimize local as well as global variables simultaneously and find optimal as well as feasible routes of the WSN.

1.2 Contribution of This Study

The contributions of this study are as follows:

1. In this study, the energy efficient routing of data packets in a wireless sensor network is studied by considering network dynamics and uncertainties by developing a Mixed Integer Programming (MIP) problem.
2. The impact of network failure or transfer latency is explored by focusing on the cascaded effect on partial or entire network lifetime and performance by comparing model variants.

3. Grey wolf optimizer (GWO) algorithm is modified to solve real world problems that comprises of large number of nodes to establish communication i.e. data transfer. The developed algorithm optimizes the solution efficiently irrespective of problem size making it robust and scalable.

The limit of the paper is organized as follows. Section 2 encloses review of relevant literature. Section 3 outlines the problem environment followed by mathematical formulation. Section 4 provides computational results with useful managerial insights. Section 5 encapsulate proposed solution technique, i.e. the 'modified Grey Wolf Optimizer (GWO)' algorithm to solve industry size problem and results obtained. Section 6 concludes the discussion and provides direction for future research.

2 Literature Review

Over the last decade, several literatures have focused to minimize the energy required to transmit data over multiple nodes. To this end, the authors have either pursued classical approaches such as Bender's Decomposition, Outer Approximation (OA), Branch and Bound (B&B) methods, or heuristic/meta-heuristic approaches to obtain optimality. Relevant literatures are categorized into two groups based on the objective of this study: (i) application of popular Evolutionary and Swarm Intelligence in wireless networking problem, and (ii) the potential of Grey Wolf Optimizer to obtain optimality in the wireless networking problem.

2.1 Application of Evolutionary Algorithms & Swarm Intelligence

In this section, several works are discussed in terms of nature-inspired, bio-inspired, meta-heuristic, and heuristic approaches as follows. Gawas and Govekar [6] proposed ACO based routing method for VANET. This is a nature-inspired based method that helps to find an optimal route between source and destination nodes. This proposal is based on IoV that stands for the Internet of Vehicle, this is one part of VANET. This proposal is based on a traditional routing protocol which named as AODV routing. ACO algorithm helps to optimize network parameters and produce the optimal path. Zhang et al. [7] proposed a GA based QoS perception routing protocol named as GABR. The main idea is to increase the QoS by concentrating on packet transmission and maintaining link stability. GA is used to optimize the global paths. It also focuses on the intersections by using greedy carry forward algorithm to deliver the packets efficiently. Tripathi and Das [8] proposed five input parameters based intelligence routing using multiple criteria of the ad-hoc network. This is based on a soft set method mixed by extended fuzzy set i.e. intuitionistic fuzzy set and two techniques of the multi-criteria decision system. Each input parameter is mapped into the soft set in terms of three elements such as true membership value, false membership value, and between both which known as hesitation membership value. Finally, it helps to resolve the uncertainty of the network efficiently and derive the optimal route of the network. Sarkar et al. [9] designed an ant-based routing protocol for MANET that is based on traditional routing protocol AODV. These ants are working based on the principle of the AOC algorithm. Two basic parameters are used in this routing method like hop-count and residual energy. Simulation results outperform some existing routing mechanisms such as AODV, DSR, and EADSR by reducing congestion and enhancing the quality of services as well as transmission rate. Bello et al. [10] proposed an improved GA based route optimization technique. This protocol determines an optimal path to communicate among vehicles between road intersections. It uses parameters like frequency of communication, received signal strength, transmit power, and path loss. This algorithm helps to cluster the chromosomes into two number overlapping clusters using the K-means algorithm. Abbas and Fan [11] proposed a clustering method for VANET which purpose is for low latency. This is based on the ACO technique. In this method, the traditional AOMDV routing method is used for handling latency and security. The main purpose is to find cluster-head and cluster nodes for data communications. Finally, it enhances some network metrics such as energy consumption, QoS, throughput, and delay of the network. Das et al. [12] proposed a details survey for soft computing technique based on its different inherent paradigms such as fuzzy logic, genetic algorithm, and neural network apart of these also contain some other method based on soft computing results. It guides the user or reader as well as new researchers about basic concepts of soft computing and its different elements and their usages. Tripathi and Das [13] proposed a vague set based routing technique for the ad-hoc network. The vague set is one of the extended versions of the fuzzy set where the fuzzy set deals with the degree of membership value and vague set deals with the degree of membership and degree of non-membership value. Later, this work is extended in [14] for evaluating more network metrics. The combination of both helps to recognize the imprecise network parameters efficiently, especially energy and distance both are the crucial parameters of the network. Finally, it helps to enhance the network metrics and network lifetime. Kaiwartya et al. [14] proposed a geocast routing protocol as GeoPSO for the optimal selection of the next-hop node. It mainly helps to enhance the network load and packet delivery ratio of the VANET. Lobiyal et al. [15] proposed a meta-heuristic method i.e. PSO for finding an optimal combination in AOMDV. It is implemented and tested on a real map and observed an improvement in QoS. Mandhare et al. [16] designed a meta-heuristic based routing protocol for MANET. The purpose of this routing is to enhance QoS of the network. The proposed method reduces the issue of non-deterministic NP hard issue. The key technique is used in this method that is Cuckoo search method. The method used in the AODV routing technique with the help RREQ and RREP packet for finding the shortest path. Finally, it compares with some nature-inspired techniques such as PSO, ACO, and simple AODV and enhance the matrices scalability and mobility, and reduce congestion of the network. Tripathi and Das [17] designed a robustness method for routing in ad-hoc network. Here, the network is based on a transparent system and heterogeneous. The basic key elements of this network are the fusion of linear programming and game theory. It consists of two objective functions for two players where both players are competing to each other. Finally, it helps to reduce conflicting nature and pave the complexity of the network and find the optimal route. Wagh and Gomathi [18] proposed a method for route designing in VANET. In this method, a novel nature-inspired technique is used such as the lion optimization technique. This novel optimization technique mixed with fuzzy logic for reducing the uncertainty of the network. Fuzzification method of the fuzzy logic is used for reducing several costs of the network such as awareness, collision, travel, congestion, and increase the reliability as well as QoS of the network. Li and Leu [19] proposed and based routing method for WANET. This is a dominated based routing protocol which uses forward and backward ant mechanism. The proposed method uses a strategy for reducing network overhead and increase the reliability of the network. In this strategy, both ants participated and also used the clustering method in this routing technique and selection of an optimal path. To this end, clustering algorithms such as COVR-AHC [20–22] can be used. Other relevant studies can be found in [23–26]. Dongyao et al. [27] designed a shuffled technique as ISFLA with the main aim of this work is to avoid congestion during data transmission. The fitness function for particles involved in path establishment depends on the residual energy of the frogs. Frogs are distinct as the adaptation numbers that are required to create a normalization for the path-selection model. The proposed leapfrog method is updated during local optimization with discrete processing. The performance of the network is enhanced based on the threshold strategy between better and weaker nodes. Similarly for global optimization, a multi-path routing idea is selected for getting the optimal path. Saritha et al. [28] proposed a PSO based optimized multipath routing protocol using learning automata. It is used to obtain optimal multiple paths for transmission depending on link stability. It employed to classify the ideal number of paths required for transmission. It performs well for QoS parameters by increasing the packet delivery ratio and throughput. Bera et al. [29] designed an intelligent based optimization technique for WSN. This is especially based new optimization technique i.e. African Buffalo Optimization (ABO) which is more efficient and helps to optimize local as well as global optimset for optimization. This is one of the most helpful nature-inspired algorithms that rapidly uses in several areas. Kadono et al. [30] proposed the routing method for ad-hoc network. This routing is based on GPSs system. The proposed method uses ACO technique for handling the network path of the ad-hoc network. In this network, the ant works as an agent that handles and compromise several network parameters and maintain parameters changeability. In this work, due to GPS ant behave as intelligent agent uses their pheromone intelligently. Finally, it helps in the packet delivery ratio. Vinoba and Vijayaraj [31] designed a topology control based method for ad-hoc network. The proposed method is the fusion of ACO and Bayesian reasoning techniques. The combination of both is used to manage the dynamic position of the network and control the residual energy of the nodes. Finally, it outperforms by several network metrics based on several parameters. Bello-Salau et al. [10] designed a routing method for VANET based on an optimization technique. This technique is based on GA optimization where several parameters are used to design constraints along with objective function. The names of the parameters are loss of path, link frequency, residual energy, the strength of the signal. The aim of this method to reduce anomaly and help to reduce noise in the network.

2.2 Grey Wolf Optimizer—Swarm Intelligence Technique with Hierarchy

Grey Wolf Optimizer (GWO) is developed by [32], and belongs to the family of meta-heuristics known as Swarm Intelligent (SI) techniques [33]. GWO distinguishes itself from other SI technique by employing a social hierarchy structure. Further, GWO require fewer parameters and is less sensitive to information derived at initial search. GWO, therefore, has been applied to wide variety of optimization problems, such as Scheduling, [34–36], Power Engineering [37, 38], Bioinformatics [39], Environmental Applications [40], Machine Learning [41], Networking [42], Wireless Sensor Network [43], and Image Processing [44]. By achieving right balance between exploration and exploitation, GWO can achieve faster convergence. Literature survey to this end could be found in [33].Further, some variants have been proposed by modifying GWO algorithm. These modifications can be categorized into five groups based on:

1. *Social hierarchy* [45] defines the structure of wolf pack based on fitness value and experience, and roles of individual wolf that needs to be carried out to search solution space in an efficient manner i.e. by mitigating the number of evaluation required, thereby, reducing time and computational complexity,
2. *Pack Initialization* [46–48] defines initial position (solution represented) of the wolf pack which is crucial to obtain an optimal solution, as it can accelerate (by covering larger search space) or inhibit (by getting stuck in local optima) the search process,
3. *Control Factor* [49, 50] defines the rate (linear or nonlinear) at which the GWO algorithm drifts from exploration (divergent search) to exploitation (convergent search) the search space for optimality,
4. *Position updation* [51, 52] defines the function which modifies wolf position based on (i) pack hierarchy i.e. pack leaders (such as alpha, beta and delta) having sufficient knowledge of search space, and (ii) individual's experience which improves with time,
5. *Hybridization* [53–59] with (i) classical methods such as branch & bound, benders decomposition, outer approximation, and (ii) heuristics/meta-heuristics such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and others to leverage the benefits.

Next follows a brief overview of these modifications in each group: In first group, social hierarchy structure is modified by dividing the pack into independent subgroups for co-operative hunting (i.e. deep exploitation) and random scout (i.e. wide exploration) [45]. The second group focuses on initialization to provide better initial points. The authors adopted tent maps [46] and good point set theory [48] to this end. In the third group, control factor is updated using non-linear function [46, 49] exponential decay function [60], Fuzzy logic for dynamic adaptation [50], to balance between exploration and exploitation. In the forth group, position of omega wolves is updated: (i) by assigning weight derived from multiplying control coefficients A and C [51], and (ii) using weighted average method, fitness function, fuzzy logic [50]. Finally, in the fifth group, new operators such as Crossover Operators [53], Genetic Algorithm [54], Differential Evolution Algorithm [55], Evolutionary Population Dynamics [57], Powell Optimization Algorithm [61], Pattern Search Algorithm [58] are employed to improve the convergence rate. However, most literatures neglect individual's experience.

3 Research Problem

In this section, the description and mathematical formulation of the problem is provided. Model variants are also formulated for comparison in following Sects. 3.2.1 and 3.2.2. All set and symbols used in the modeling is provided as in Table 1.

3.1 Problem Statement

In this research work, an uni-direction network is considered with N nodes and E edges. Multiple connection *(p)* can be established by sets of nodes and edges in the network, each originating at a source node (S_p) and ending at a destination node (D_p). These source nodes (S_p) and destination nodes (D_p) of individual connection cannot be same, i.e. ($S_p \neq D_p;\ \forall p$) as each connection is mutually exclusive and collectively inclusive. However, the connection(s) may share nodes to transmit information. Energy is consumed to transmit information from one node to another node, i.e. via edges. The objective is to reduce energy consumption or loss of energy in overall network. However, energy loss at each node is not considered in this study.

Next, a summary of sets and symbols are presented in Table 1 which are used in the mathematical model in Sect. 3.1.

3.2 Model Formulation

Objective function: The objective is to minimize energy consumption/loss during transmission of multiple source-destination network. This can be computed by aggregating energy loss at every destination as shown by Eq. (1).

$$\text{Z} = \sum_{p=1}^{P} \sum_{j=D_p} G_{pj} \tag{1}$$

Table 1 Notations

Index	*Description*
p	Index of connection $(1, 2, 3, \ldots, P)$
i,j	Index of nodes $(1, 2, 3, \ldots, N)$
Parameter	*Description*
P	Number of connections
N	Number of nodes in the network
D_p	Destination node for connection *p*
S_p	Source node for connection *p*
E_{ij}	Energy required to transmit information from node *i* to node *j*
L_{ij}	Variation of energy required between node *i* to node *j*
V_j	*1* if node *j* is available, *0* otherwise
C_{ij} & E_{ij}	Energy required to transmit data from node i to node j
M	Big-M i.e. sufficiently large value
Variable	*Description*
G_{pj}	Energy used at node *j* for connection *p*, where $G_{pj} \in R^{+}$
Binary Variable	*Description*
x_{pij}	1 if connection *p* exist from node *i* to node *j*, 0 otherwise

Subject to constraints:

1. *Number of succeeding nodes:* Number of nodes that carry out transmission from current node *i* is limited to one, i.e. node *j*, as shown by Eq. (2).

$$\sum_{j=1}^{N} x_{pij} \leq 1; \forall\, p, i \tag{2}$$

2. *Transmission Completion:* Eq. (3) ensures completion of transmission from source S_p to destination D_p.

$$\sum_{i=1}^{N} x_{pij} = 1; \forall\, p, j, \text{ where } j = D_p \tag{3}$$

3. *Source and destination node:* Eqs. (4–5) eliminates any incoming and outgoing transmission from source and destination node, respectively.

$$\sum_{i=1}^{N} x_{pij} = 0; \forall\, p, j, \text{ where } j = S_p \tag{4}$$

$$\sum_{j=1}^{N} x_{pij} = 0; \forall\, p, i, \text{ where } i = D_p \tag{5}$$

4. *Flow conservation:* Number of incoming nodes is equal to number of outgoing nodes. It is captured by Eq. (6).

$$\sum_{i=1}^{N} x_{pij} = \sum_{h=1}^{N} x_{pjh};\ \forall\ p,\ j, \text{ where } j \neq S_p \ \&\ j \neq D_p \tag{6}$$

5. *Energy usage at source:* Amount of energy used at source node is initialized to zero (0) using Eq. (7).

$$G_{pj} = 0;\ \forall p,\ j = S_p \tag{7}$$

6. *Energy usage when transmission:* Amount of energy used at source, intermediate and destination node while transmitting information form S_p to D_p shown by Eq. (8).

$$G_{pj} \geq \sum_{i=1}^{J} x_{pij}(G_{pi} + E_{ij});\ \forall\ p, \text{ where } j \neq S_p \ \&\ j \neq D_p \tag{8}$$

It can be observed that Eq. (8) is non-linear in nature. However, it's linear equivalent can be obtained introducing the McCormick's envelope.

In the next section, two model variants are described as follows:

1. In the first model variant, the dynamic nature of a wireless network is considered, i.e. when two or more nodes do not maintain a fixed distance over time. Ignoring the impact of nodal dynamics in wireless network may result in infeasibility or sub-optimal solution, and therefore is imperative for study.
2. In the second model variant, the uncertainty in performance of active nodes is considered i.e. when one or more nodes in the network may become offline due to technical issues. Neglecting the impact of uncertain offline nodes in wireless network may result in infeasibility or sub-optimal solution, and therefore is also imperative for the study.

3.2.1 Dynamic Nodes

In reality, nodes transmitting information are mobile in nature or encounter certain delay, i.e. dynamic in terms of response time. To capture network dynamics while transmitting information via connection (p) from source (S_p) to destination (D_p) in a simple manner, Eq. (9) is introduced replacing Eq. (8).

$$G_{pj} \geq \sum_{i=1}^{J} x_{pij}(G_{pi} + E_{ij} + L_{ij});\ \forall\ p, \text{ where } j \neq S_p \ \&\ j \neq D_p \tag{9}$$

where, L_{ij} represents variation of distance between node i and node j.

Taking dynamic nature of wireless network into account, the optimal path for transmission of information is determined. The dynamic characteristic captured using L_{ij} is also reflected in the objective function as G_{pj} which constitutes both E_{ij}, L_{ij}. With increase in energy required (i.e. by L_{ij} due to variation), the edge $(i - j)$ becomes less likely to chosen for transmission.

3.2.2 Offline Nodes

Often, transmitting node can fail over time i.e sensors can go offline which prevents data transmission, making it less suitable for operations. To counteract, alternate nodes can be employed to transmit information successfully, even at the cost of higher energy usage.

Therefore, any unavailable node denoted by (V_j) is eliminated from consideration. It is because, an inclusion of any unavailable node may render the model infeasible. Further, this will prevent network failure, and thus can improve network reliability. The node failure is captured by Eq. (10) using V_j, based on which alternative path is determined.

$$x_{pij} \leq V_j; \forall\, p, i, j \neq S_p \;\&\; j \neq D_p \tag{10}$$

The source and destination node is functional here, which otherwise implies no information is transmitted from source to destination.

3.2.3 Solution Scheme

The mathematical model is formulated in General Algebraic Modeling System (GAMS) environment and solved using SBB solver. SBB is a commercial solver that employs Simple Branch and Bound technique to obtain optimal solution. As the objective function and constraints are all linear (including Eq. 8 after linearization), thus convex, the solution obtained is the global optima.

However, for a large size problem, a modified Grey Wolf Optimizer (mGWO) is implemented on the base model with single route / connection (i.e. $P = 1$) between source and destination node. The work mechanism of the modified Grey Wolf Optimizer (mGWO) is provided in Sect. 5.

4 Results

4.1 Experimental Design

An uni-directional wireless network consisting four nodes and 6 edges is considered. Location of nodes in the network are dynamic (i.e. mobile) and nodes transmitting information are subject to failure. Therefore, three model variants exists, each capturing distinct problem characteristics, as given by Table 2, with reference to Sect. 3.2.1.

To illustrate model variants, problem parameter considered are as follows. The value of energy consumption (E_{ij}), variance in node location (L_{ij}) and node availability (V_j) is aggregated in Table 3 respectively.

Further, two connection (say *Connection-A* and *Connection-B*) is considered as follows:

1. *Connection-A:* From node 2 (as source) to destination 1 (as destination), and
2. *Connection-B:* From node 1 (as source) to destination 4 (as destination).

It is to be noted that, connection A & B share node 2 & 3. However, at node 2 & 3, each connection may have different energy consumption values.

Table 2 Model variants with specific characteristics

	Features					
Model/ Network	*N*	*E*	*Source-Destination*	*Energy consumption*	*Node distance*	*Node availability*
Basic Model	6	4	N-N	Yes	Fixed	Yes
Dynamic Nodes	6	4	N-N	Yes	Variable	Yes
Offline Nodes	6	4	N-N	Yes	Fixed	Yes / No

Table 3 Parameter Design

Nodes	1	2	3	4		*Nodes*	1	2	3	4		*Nodes*	*Available*
1	-	5	-	-		1	-	-	-	-		1	1
2	15	-	5	15		2	-	-	30	-		2	1
3	5	-	-	5		3	-	-	-	-		3	0
4	-	-	-	-		4	-	-	-	-		4	1
(A) Energy consumption(E_{ij})						(B) Variation(L_{ij})						(C) Node availability(V_j)	

4.2 *Computational Results:*

4.2.1 Basic Network

The optimal path for basic network is as follows: Connection 1 follow the path 2 → 3 → 1, and Connection 2 follow the path 1 → 2 → 3 → 4. Total energy consumption is 25 units. This can be observed in Fig. 2a.

It can be computed that node 2 & 3 is shared by connection A & B. Edges with low energy consumption is chosen (ignoring any energy loss at the nodes). As, a result edge (2 → 1) & (2 → 4) remains inactive.

4.2.2 Network with Dynamic Nodes

The optimal path for basic network is as follows: Connection 1 follow the path 2 → 1, and Connection 2 follow the path 1 → 2 → 4. Total energy consumption is 35 units. This can be observed in Fig. 2b.

Similarly, edges with low energy consumption is chosen (ignoring any energy loss at the nodes). However, as energy consumption increases by 30 due to network dynamics at edge (2 → 3). Thus, it becomes less preferable. As a result edges (2 → 1) & 2 → 4 is chosen for connection A & B, respectively, eliminating edge (2 → 3). This results in total energy consumption of 35 units, which otherwise would have been 85 units (i.e. incurring loss of 50 units). Thus, network dynamics must be accommodated to prevent any additional loss of energy.

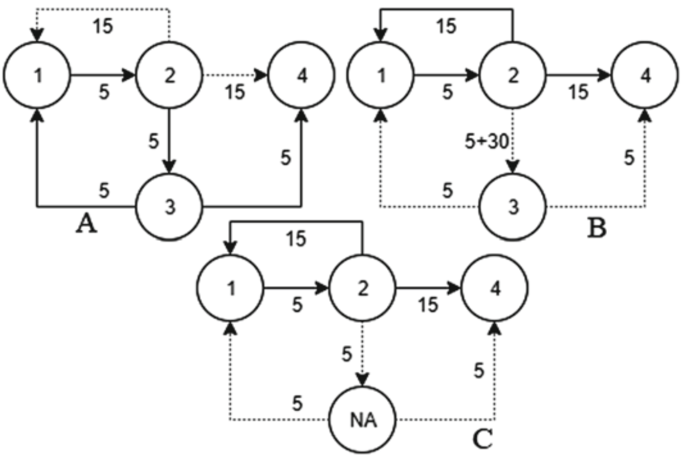


Fig. 2 Data Transmission in a network without node dynamics and uncertainty (**a**), with network dynamics (**b**) and offline nodes (**c**)

4.2.3 Network with Offline Nodes

The optimal path for basic network is as follows: Connection 1 follow the path $2 \rightarrow 1$, and Connection 2 follow the path $1 \rightarrow 2 \rightarrow 4$. Total energy consumption is 35 units. This can be observed in Fig. 2c.

In Fig. 1c, node 3 is eliminated to capture node unavailability. As, node 3 is unavailable, optimal path excluding node 3 is determined. Thus, edges $(2 \rightarrow 1)$ & $2 \rightarrow 4$ is chosen for connection A & B, respectively. This results in total energy consumption of 35 units, which otherwise would have been 85 units (i.e. incurring loss of 50 units). In contrast to previous model, an inclusion of unavailable node would render the network as infeasible, and result in network failure. Thus, any unavailable node must be taken into account to prevent infeasibility and improve network performance.

In a nutshell, by comparing three model variants, it is found that ignoring dynamic behavior or node unavailability of a network may lead to inferior and infeasible solution. Thus, it would be beneficial to incorporate these characteristics into the model.

5 Solution Approach

A grey wolf optimization (GWO) algorithm is employed for solving aforesaid networking problems. Grey wolf optimizer was initially developed by [32] to solve classical engineering design problems and provide a real application in optical engineering. GWO is inspired by the leadership hierarchy and hunting mechanism of grey wolves in nature. The GWO algorithm enhances the ability to diversify the search and prevent the population individuals falling into stagnation. Next, Sect. 5.1 presents the outline of the GWO algorithm, followed by Sect. 5.2 which presents key aspects of the GWO algorithm, and Sect. 5.3 which includes the proposed GWO algorithm in detail.

5.1 Grey Wolf Optimization

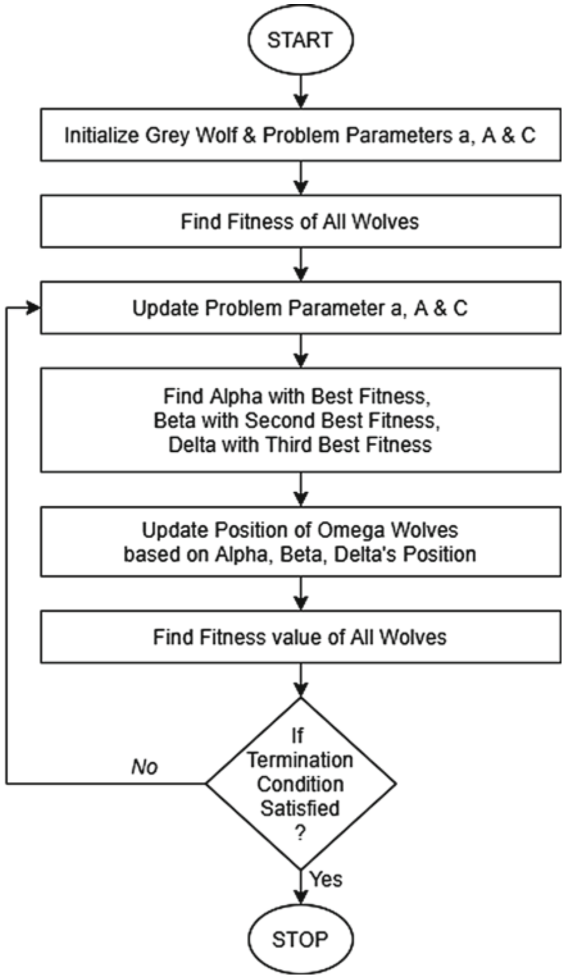
Grey wolves are a predatory species that prefers to live and hunt as a pack. A grey wolf pack usually consists of five to twelve members. The pack members maintain a strict social hierarchy which defines an individual's roles at each level,

1. *Alpha*: The leader of the pack is known as the Alpha(α), who is responsible for making crucial decisions and dictating the pack.
2. *Beta* : Beta (β) acts as subordinate leader, responsible for assisting α and pack activities. When α passes away or becomes too old, it is often succeeded by β.

3. *Delta*: Delta (δ) serves multiple roles, such as securing the territory, alarming the pack of incoming danger, and attending to wounded/weak pack members.
4. *Omega*: δ follows α and β, but dominates Omega (ω) wolves. ω lies at the bottom of the hierarchy, and submit to all dominant grey wolves.

The primary activities of grey wolf pack is to encircle, hunt and attack the prey. α, β and δ have knowledge about potential location of the prey, thus guides the pack in these activities. Therefore, grey wolf optimizer (GWO) algorithm mimics the characteristics of natural grey wolves (Fig. 3).

Fig. 3 Work mechanism of Grey Wolf Optimization (GWO) algorithm



5.2 Aspects of Grey Wolf Optimization

The key aspects of GWO algorithm is outlined as follows.

1. *Social Hierarchy*: The social hierarchy or pack order is ranked based on individual's fitness. The best solution is considered as α, consequently, the second and third best solution is considered as β and δ, respectively. Remaining individuals without any rank are considered as ω.
2. *Encircling prey*: The pack encircles at a distance $(D(t))$ around potential location of the prey $(Y_{nj}(t))$ using Eq. (11–12),

$$D(t) = |C \times X_j^p - X_{nj}(t)| \tag{11}$$

$$Y_{nj}(t) = X_j^p - A \times D(t) \tag{12}$$

where t denotes current iteration of maximum $T_{\max}$ iterations, n represents an individual of the pack consisting of N members, j represents search area based on number of destination, X_{njt}^p is potential position of prey, and X_{njt} is individual's current position. A and C are control coefficients which are computed using Eq. (13) and Eqs. (14–15), respectively,

$$C = 2r_1 \tag{13}$$

$$A = 2ar_2 - a \tag{14}$$

$$a = 2(1 - \frac{t}{T_{\max}}) \tag{15}$$

where, r_1 and r_2 are randOm variables in range [0, 1], a decreases linearly from $a_{\max} = 2$ to $a_{\min} = 0$ [32] over $T_{\max}$ number of iterations [62]. In Eq. (12), A acts as the convergence factor that drifts GWO algorithm from exploration to exploitation of optimal solution. Encircling establishes the strategy in which the wolf pack hunts.
3. *Hunting*: The α, β and δ wolves have better knowledge about potential location of the prey, thus guides ω wolves. This is carried out in three steps:

   (i) Calculating the distance $(D_{njt}^\alpha, D_{njt}^\beta, D_{njt}^\delta)$ from an individual ω wolf to α, β, δ wolves using Eqs. (16–18).

$$D_{nj}^\alpha(t) = |C^\alpha \times X_j^\alpha - X_{nj}(t)| \tag{16}$$

$$D_{nj}^\beta(t) = |C^\beta \times X_j^\beta - X_{nj}(t)| \tag{17}$$

$$D^{\delta}_{nj}(t) = |C^{\delta} \times X^{\delta}_{j} - X_{nj}(t)| \quad (18)$$

(ii) Identifying potential location of the prey $(Y^{\alpha}_{njt}, Y^{\beta}_{njt}, Y^{\delta}_{njt})$ guided by α, β and δ, respectively, derived from Eqs. (16–18).

$$Y^{\alpha}_{nj}(t) = X^{\alpha}_{j} - A^{\alpha} \times D^{\alpha}_{nj}(t) \quad (19)$$

$$Y^{\beta}_{nj}(t) = X^{\beta}_{j} - A^{\beta} \times D^{\beta}_{nj}(t) \quad (20)$$

$$Y^{\delta}_{nj}(t) = X^{\delta}_{j} - A^{\delta} \times D^{\delta}_{nj}(t) \quad (21)$$

(iii) Updating the position for subsequent iteration $(t + 1)$ by aggregating results predicted from α, β and δ wolves, i.e. by taking average of $(Y^{\alpha}_{njt}, Y^{\beta}_{njt}, Y^{\delta}_{njt})$, as shown in Eq. (22).

$$X_{nj}(t + 1) = \frac{Y^{\alpha}_{nj}(t) + Y^{\beta}_{nj}(t) + Y^{\delta}_{nj}(t)}{3} \quad (22)$$

4. *Attacking:* Grey wolves complete hunting by attacking the prey. It is evident when the pack stops moving i.e. with decrease in value of 'a' (i.e. less than 1) and relative motion of the pack is 0 or closer to it. The position of attack is the optimal solution obtained.

5.3 Modified Grey Wolf Optimization

The GWO algorithm efficiently searches for the optimal solution by leveraging on exploration and exploiting the search space through a hierarchical guided-search strategy. However, the basic GWO algorithm cannot be used due to obvious incompatibility. Thus, the following modifications were carried out.

1. Continuous GWO algorithm have been modified to optimize binary variables by encoding continuous variables into its binary equivalent, which is shown in *'Get Binary Matrix'* algorithm. This allows smooth optimization reducing the level of complexity.
2. GWO algorithm is modified by incorporating a mechanism to eliminate region of space from search, which are explored by another wolf currently. This improves the search capability significantly by removing duplicate solutions leading to a much faster convergence.

3. GWO replaces wolf position(s) which results in an infeasibility by introducing new position(s) which is generated randomly from an uniform distribution. This enhances the ability of GWO algorithm to explore wider solution search space and prevent local stagnation.

Next, the detailed GWO algorithm is provided, which is further followed by its components in subsequent sections.

Data: N = Number of Wolves
T = Number of Iterations
J = Length of Encoded String
Result: Optimal Solution = $\{X^{\alpha}, F^{\alpha}\}$
Initialize best solution = ∞^{+}
position = random(N, J)
Set $i = 0$
while $i < N$ **do**
  Increment i = i + 1
  binary matrix = Get Binary Matrix (position[i])
  fitness[i] = objective function (*binary matrix*)
end
$\{F^{\alpha}, X^{\alpha}\}, \{F^{\beta}, X^{\beta}\}, \{F^{\delta}, X^{\delta}\}$ = Get Rank (position,fitness)
Reset $i = 0$
while $i < T$ **do**
  Increment i = i + 1
  Assign a = $2\{1 - \frac{i}{T}\}$
  Update position[i] using Eq. (22)
  if *position*[i] *exists* **then**
    Delete duplicate position[i]
    Insert random(1, J) as position[i]
  end
  binary matrix = GetBinaryMatrix (position[i])
  fitness[i] = objective function (*binary matrix*)
  $\{F^{\alpha}, X^{\alpha}\}, \{F^{\beta}, X^{\beta}\}, \{F^{\delta}, X^{\delta}\}$ = Get Rank (position,fitness)
end

Algorithm 1: Grey Wolf Optimizer

5.3.1 Pack Representation

The pack of wolves or population is represented by a set of continuous variables in range [0, 1], which is generated randomly from an uniform distribution. The pack representation is presented in Fig. 4, where each rows (n) enumerates the number of wolves and entire column (j) of a row reflects position of wolves (as from-to combination array) through a single dimensional array. Therefore, the pack schema can be represented as $X_{n,j}$ (*i.e.* $n \times j$ continuous values).

Fig. 4 Schematic diagram of the node representations as in wolf pack structure

$$\begin{bmatrix} wolves \downarrow & variable\ positions & \rightarrow & & & & \\ 1 & X_{11} & X_{12} & X_{13} & \cdots & X_{1j} & \cdots & X_{1J} \\ 2 & X_{21} & X_{22} & X_{23} & \cdots & X_{2j} & \cdots & X_{2J} \\ 3 & X_{31} & X_{32} & X_{33} & \cdots & X_{3j} & \cdots & X_{3J} \\ \vdots & & & & & & & \\ n & X_{n1} & X_{n2} & X_{n3} & \cdots & X_{nj} & \cdots & X_{nJ} \\ \vdots & & & & & & & \\ N & X_{N1} & X_{N2} & X_{N3} & \cdots & X_{Nj} & \cdots & X_{NJ} \end{bmatrix}$$

The continuous values are converted to its binary form through encoding process which is provided in the next section, for computing the fitness value of individual wolf. For example, if $X_{1,j} = [0.3, 0.5, 0.1]$ for wolf ($n = 1$) in a 5 node problem, it implies that data is transferred in order = $\{s \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow n\}$, where node s and n represents source and destination nodes respectively.

5.3.2 Encoding Scheme

In contrast to continuous GWO, where positions are represented by real and continuous numbers, positions in this study are bounded to binary values 0 & 1. Therefore, position of wolves is mapped from real value to its binary equivalent matrix using 'Get Binary Matrix' algorithm. A simple ranking method conversion is used for this purpose.

Data: A = Input Matrix
N = Number of Nodes
T = Threshold
Sequence = [ ]
Result: X = Binary Matrix [$N \times N$]
Append Source node to Sequence
while $i < N$ **do**
  Increment $i = i + 1$
  Compute $M = max(A)$
  if $M > T$ **then**
    Index = Search M position in A
    Append Index to Sequence
  end
  Append Destination node to Sequence
  while $i < N$ **do**
    Set R = Sequence[i]
      C = Sequence[$i + 1$]
    Assign $X_{R,C}$ = 1
  end
end

Algorithm 2: Get Binary Matrix

To explain, the previous example is considered, where $X_{1,j} = [0.3, 0.5, 0.1]$ for wolf ($n = 1$) in a 5 node problem implies that data is transferred in order = $\{s \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow n\}$, where node s and n represents source and destination nodes respectively. This is encoded in an array (A) of shape ($n \times n$) by assigning $A_{s,2} = A_{2,1} = A_{1,3} = A_{3,n} = 1$ and remaining $A_{i,j} = 0$.

5.3.3 Position Updation

To update the position of wolf pack, Eq. (22) is followed. For this purpose, the position (X^α, X^β, X^δ) and fitness (F^α, F^β, F^δ) of three wolves - α, β, δ are obtained using 'Get Rank' algorithm as shown below.

Data: X = Position of Wolves
  N = Number of Wolves
  fitness = Fitness of Wolf Pack
Result: $\{F^\alpha, F^\beta, F^\delta\}$
  $\{X^\alpha, X^\beta, X^\delta\}$
Initialize $i = 0$
  $F^\alpha, F^\beta, F^\delta$ by ∞
  $X^\alpha, X^\beta, X^\delta$ by $\{\emptyset\}$
while $i < N$ **do**
  Increment i = i + 1
  if *fitness*$[i] < F^\delta$ **then**
    Update F^δ = *fitness*$[i]$
      $X^\delta = X[i]$
    if *fitness*$[i] < F^\beta$ **then**
      Update $F^\delta = F^\beta$
        $X^\delta = X^\beta$
        F^β = *fitness*$[i]$
        $X^\beta = X[i]$
      if *fitness*$[i] < F^\alpha$ **then**
        Update $F^\delta = F^\beta$
          $X^\delta = X^\beta$
          $F^\beta = F^\alpha$
          $X^\beta = X^\alpha$
          F^α = *fitness*$[i]$
          $X^\alpha = X[i]$
      end
    end
  end
end

Algorithm 3: Get Rank

After obtaining the position of α, β, δ wolves, Eqs. (16–22) is used to update the position of wolves (except α, β, δ) in the pack. Therefore, the position update procedure serves as a motivation for the pack to hunt closer to the prey based on the fitness value of pack leaders i.e. (α, β, δ). This can accelerate the convergence towards optimal solution.

For example, consider a schematic representation of the wolf pack for a network comprising of 7 nodes excluding source and destination as $\{(0.3, 0.5, 0.6), (0.7, 0.3, 0.5), (0.3, 0.1, 0.6), (0.9, 0.5, 0.3), (0.5, 0.6, 0.2)\}$ with respective fitness values $\{4, 7, 1, 3, 5\}$. The position and fitness computed for α is $(0.7, 0.3, 0.5)$ and 7, β is $(0.5, 0.6, 0.2)$ and 5, and δ is $(0.3, 0.5, 0.6)$ and 4, respectively, by using 'Get Rank' algorithm, which is used in Eqs. (16–22) to update position of ω wolves i.e. for wolf $n = 3, 4$ with position $(0.3, 0.1, 0.6)$, $(0.9, 0.5, 0.3)$ respectively.

5.3.4 Fitness Function

The fitness value of individual wolf is computed using '*Objective function*' algorithm. Fitness is the product of binary matrix obtained from '*Get Binary Matrix*' algorithm and Cost matrix which indicates the loss of energy while data is transmitted from one node to another. Therefore, the objective function in this study aims to reduce the amount of energy during transmission, which means that it is a minimization problem.

Data: X = Position of Wolves
N = Number of Wolves
Result: Z = Fitness of Wolf Pack
Initialize $n = 0$
while $n \leq N$ **do**
  Increment n = n + 1
  B = Get Binary Matrix $(X[n])$
  $Z[n] = \sum_{i=1}^{J} \sum_{j=1}^{J} (B_{i,j} \times Cost_{i,j})$
end

Algorithm 4: Objective function

To illustrate, the computation of fitness value for a single wolf (say $n = 2$) through 'Objective function' for a 5 node problem including source $(i = 1)$ and destination $(i = 5)$ is shown in Fig. 5 below.

5.4 *Results Obtained*

To evaluate the grey wolf optimizer algorithm, 3 problems set of 25, 50, and 100 destination nodes where each having 10 problem instances are generated randomly. Therefore, the total number of the problem considered for this study is $3 \times 10 = 30$. Every problem instance from each set is executed 10 times to confirm the robustness of the algorithm, which is captured using mean and standard deviation and reported

C_{ij} × B_{ij} = $C_{ij} \times B_{ij}$ → $\sum_{i=1}^{J}\sum_{j=1}^{J}(C_{ij} \times B_{ij})$

C_{ij}:

	1	2	3	4	5
1	-	2	1	1	3
2	2	-	3	2	3
3	2	4	-	1	2
4	3	2	3	-	1
5	1	3	1	1	-

B_{ij}:

	1	2	3	4	5
1	-	0	1	0	0
2	0	-	0	0	1
3	0	0	-	1	0
4	0	1	0	-	0
5	0	0	0	0	-

$C_{ij} \times B_{ij}$:

	1	2	3	4	5
1	-	0	1	0	0
2	0	-	0	0	3
3	0	0	-	1	0
4	0	2	0	-	0
5	0	0	0	0	-

$\sum_{i=1}^{J}\sum_{j=1}^{J}(C_{ij} \times B_{ij})$ = 7

Fig. 5 Product of binary matrix and cost term which denotes energy consumption in the total network

Table 4 Mean and Standard Deviation of results obtained from 30 problem set

Problem	Nodes = 25			Nodes = 50			Nodes = 100	
Set	Mean (μ)	- Std Dev (σ)		Mean (μ)	- Std Dev (σ)		Mean (μ)	- Std Dev (σ)
1	367	45		900	127		2152	109
2	366	31		932	61		2222	94
3	322	52		890	86		2121	78
4	366	40		919	73		2217	73
5	345	30		872	105		2127	172
6	445	49		971	65		2164	93
7	389	44		906	100		2205	128
8	363	63		918	93		2138	111
9	340	28		913	76		2168	139
10	353	42		973	73		2168	98

in Table 4. The convergence curve of the optimal solution for each problem (P1 to P30) is also presented in Figs. 6, 7 and 8.

From Table 4, it can be observed that the GWO algorithm produces optimal solution with significantly low standard deviation for all problem size. Thus, the GWO algorithm responds well in terms of scalability.

The average of standards deviation from all problem set reflects the efficiency of the GWO algorithm to produce results closer to the optimal solution. Thus, the GWO algorithm implemented in study is not only computationally efficient, but also robust to parameter changes and reliable to optimize routing.

From Figs. 6, 7 and 8, it can observed that the algorithm can escape from stagnation, due to hierarchical improvisation from α, β, δ wolves and by eliminating duplicate solutions from consideration. Further, it improves performance of the algorithm, which can be observed as a faster convergence rate due to lesser computations.

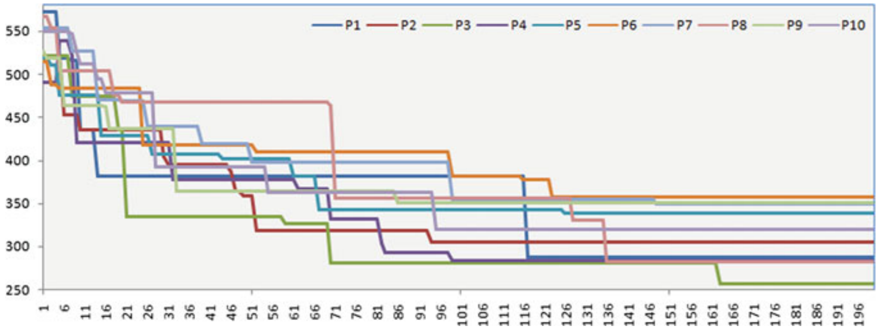


Fig. 6 Convergence curves of 10 problems (P1 to P10) with 25 nodes

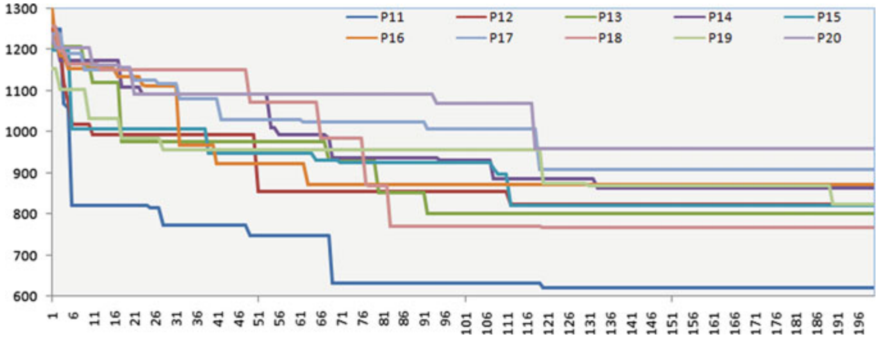


Fig. 7 Convergence curves of 10 problems (P11 to P20) with 50 nodes

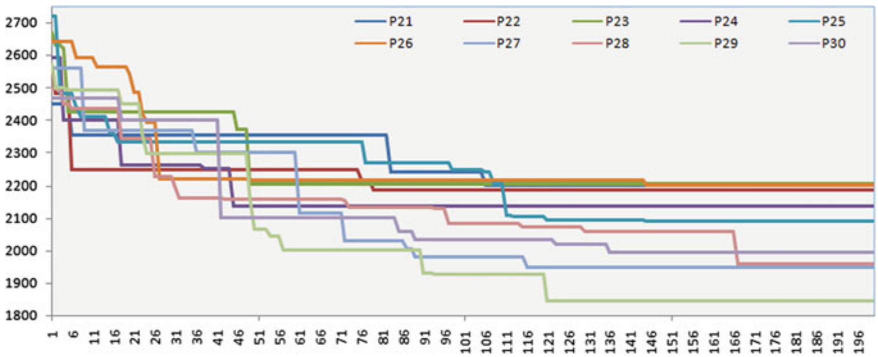


Fig. 8 Convergence curves of 10 problems (P21 to P30) with 100 nodes

6 Conclusion

In this study, the strategies of efficient routing are investigated with two model variants by formulating as a Mixed-Integer Programming (MIP) problem. By comparing model variants, the loss of energy when network dynamics and offline nodes are not taken into account is studied. Ignoring network dynamics and uncertainty may lead to network failure due to transmission overload, inconsistency and delays in data transmission, single or grouped or cascaded node failures, and even complete or partial network shutdown. Therefore, this study addresses a crucial aspect of wireless networking problem and provide actionable insights into the problem. Further, a modified grey wolf optimization technique has been employed to solve a moderately large size problem, which commercial solvers cannot. The plausible dimension to this study extends to several avenues such as exploring non-linear delay functions, considering uncertainty in network failures, factors such as power management when optimizing routing, application of other heuristic techniques.

References

1. Nguyen, L., & Nguyen, H. Y. (2020). Mobility based network lifetime in wireless sensor networks: A review. In *Computer networks* (p. 107236).
2. Banerjee, P. S., Mandal, S. N., De, D., & Maiti, B. (2020). Rl-sleep: Temperature adaptive sleep scheduling using reinforcement learning for sustainable connectivity in wireless sensor networks. In *Sustainable computing: Informatics and systems*, (Vol. 26, p. 100380).
3. Tekin, N., & Gungor, V. C. (2020). The impact of error control schemes on lifetime of energy harvesting wireless sensor networks in industrial environments. In *Computer Standards & Interfaces* (p. 103417).
4. Poornima, I. G. A., & Paramasivan, B. (2020). Anomaly detection in wireless sensor network using machine learning algorithm. In *Computer communications*.
5. Moridi, E., Haghparast, M., Hosseinzadeh, M., & Jassbi, S. J. (2020). Fault management frameworks in wireless sensor networks: A survey. In *Computer communications*.
6. Gawas, M. A., & Govekar, S. S. (2019) A novel selective cross layer based routing scheme using aco method for vehicular networks. *Journal of Network and Computer Applications, 143*, 34–46.
7. Zhang, G., Min, W., Duan, W., & Huang, X. (2018). Genetic algorithm based qos perception routing protocol for vanets. *Wireless Communications and Mobile Computing*.
8. Santosh Kumar Das and Sachin Tripathi. (2018). Intelligent energy-aware efficient routing for manet. *Wireless Networks*, *24*(4), 1139–1159.
9. Sarkar, D., Choudhury, S., & Majumder, A. (2018) Enhanced-ant-aodv for optimal route selection in mobile ad-hoc network. *Journal of King Saud University-Computer and Information Sciences*.
10. Bello-Salau, H., Aibinu, A. M., Wang, Z., Onumanyi, A. J., Onwuka, E. N., & Dukiya, J. J. (2019). An optimized routing algorithm for vehicle ad-hoc networks. *Engineering Science and Technology, an International Journal*, *22*(3), 754–766.
11. Abbas, F., & Fan, P. (2018). Clustering-based reliable low-latency routing scheme using aco method for vehicular networks. *Vehicular Communications*, *12*, 66–74.
12. Das, S. K., Kumar, A., Das, B., & Burnwal, A. P. (2013). On soft computing techniques in various areas. *Comput. Sci. Inf. Technol., 3*, 59.

13. Das, S. K., & Tripathi, S. (2016). Energy efficient routing protocol for manet using vague set. In *Proceedings of fifth international conference on soft computing for problem solving* (pp. 235–245). Heidelberg: Springer.
14. Kaiwartya, O., & Kumar, S. (2014). Geocasting in vehicular adhoc networks using particle swarm optimization. In *Proceedings of the international conference on information systems and design of communication* (pp. 62–66).
15. Lobiyal, D. K., Katti, C. P., & Giri, A. K. (2015). Parameter value optimization of ad-hoc on demand multipath distance vector routing using particle swarm optimization. *Procedia Computer Science*, *46*, 151–158.
16. Mandhare, V. V., Thool, V. R., & Manthalkar, R. R. (2016). Qos routing enhancement using metaheuristic approach in mobile ad-hoc network. *Computer Networks*, *110*, 180–191.
17. Santosh Kumar Das & Sachin Tripathi. (2018). Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming. *Applied Intelligence*, *48*(7), 1825–1845.
18. Wagh, M. B., & Gomathi, N. (2018). Route discovery for vehicular ad hoc networks using modified lion algorithm. *Alexandria Engineering Journal, 57*(4), 3075–3087.
19. Li, K.-H., & Leu, J.-S. (2015). Weakly connected dominating set-assisted ant-based routing protocol for wireless ad-hoc networks. *Computers & Electrical Engineering*, *48*, 62–76.
20. Bera, S., Chattopadhyay, M., & Dan, P. K. (2018). A two-stage novel approach using centre ordering of vectors on agglomerative hierarchical clustering for manufacturing cell formation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 232*(14), 2651–2662.
21. Wei, D., Jin, Y., Vural, S., Moessner, K., & Tafazolli, R. (2011). An energy-efficient clustering solution for wireless sensor networks. *IEEE Transactions on Wireless Communications*, *10*(11), 3973–3983.
22. Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications, 30*(14–15), 2826–2841.
23. Singh, J., Singh, A., & Shree, R. (2011). An assessment of frequently adopted unsecure patterns in mobile ad hoc network: Requirement and security management perspective. *International Journal of Computer Applications*, *24*(9), 0975–8887.
24. Singh, J., Banka, H., & Verma, A. K. (2019). A bbo-based algorithm for slope stability analysis by locating critical failure surface. *Neural Computing and Applications, 31*(10), 6401–6418.
25. Binh, H. T. T., Hanh, N. T., Dey, N., et al. (2018). Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. *Neural Computing and Applications, 30*(7), 2305–2317.
26. Yang, W., Wang, X., Song, X., Yang, Y., & Patnaik, S. (2018). Design of intelligent transportation system supported by new generation wireless communication technology. In *Intelligent systems: Concepts, methodologies, tools, and applications* (pp. 715–732). IGI Global.
27. Jia, D., Zou, S., Li, M., & Zhu, H. (2016). Adaptive multi-path routing based on an improved leapfrog algorithm. *Information Sciences, 367*, 615–629.
28. Saritha, V., Venkata Krishna, P., Misra, S., & Obaidat, M. S. (2017). Learning automata based optimized multipath routingusing leapfrog algorithm for vanets. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1–5). IEEE.
29. Bera, S., Das, S. K., & Karati, A. (2020). Intelligent routing in wireless sensor network based on african buffalo optimization. In *Nature Inspired Computing for Wireless Sensor Networks* (pp. 119–142). Berlin: Springer.
30. Kadono, D., Izumi, T., Ooshita, F., Kakugawa, H., & Masuzawa, T. (2010). An ant colony optimization routing based on robustness for ad hoc networks with gpss. *Ad Hoc Networks*, *8*(1), 63–76.
31. Vinoba, R., & Vijayaraj, M. (2020). Novel control topology with obstacle detection using rdpso-gba in mobile ad-hoc network. *Computer Communications*.
32. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

33. Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications, 30*(2), 413–435.
34. Chao, L., Gao, L., Li, X., & Xiao, S. (2017). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Engineering Applications of Artificial Intelligence*, *57*, 61–79.
35. Jiang, T., & Zhang, C. (2018). Application of grey wolf optimization for solving combinatorial problems: Job shop and flexible job shop scheduling cases. *IEEE Access*, *6*, 26231–26240.
36. Pradhan, M., Roy, P. K., & Pal, T. (2016). Grey wolf optimization applied to economic load dispatch problems. *International Journal of Electrical Power & Energy Systems, 83*, 325–334.
37. Mohamed, A. A. A., El-Gaafary, A. A. M., Mohamed, Y. S., & Hemeida, A. M. (2016). Multi-objective modified grey wolf optimizer for optimal power flow. In *2016 eighteenth international middle east power systems conference (MEPCON)* (pp 982–990). IEEE.
38. Gupta, E., & Saxena, A. (2016). Grey wolf optimizer based regulator design for automatic generation control of interconnected power system. *Cogent Engineering*, *3*(1), 1151612.
39. Qiang, T., Chen, X., & Liu, X. (2019). Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Applied Soft Computing*, *76*, 16–30.
40. Yao, P., Wang, H., & Ji, H. (2016). Multi-uavs tracking target in urban environment by model predictive control and improved grey wolf optimizer. *Aerospace Science and Technology*, *55*, 131–143.
41. Katarya, R., & Verma, O. P. (2018). Recommender system with grey wolf optimizer and fcm. *Neural Computing and Applications, 30*(5), 1679–1687.
42. Mirjalili, S. (2015). How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, *43*(1), 150–161.
43. Rajakumar, R., Amudhavel, J., Dhavachelvan, P., & Vengattaraman, T. (2017). Gwo-lpwsn: Grey wolf optimization algorithm for node localization problem in wireless sensor networks. *Journal of Computer Networks and Communications*.
44. Emary, E., Zawbaa, H. M., Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381.
45. Yang, B., Zhang, X., Tao, Y., Shu, H., & Fang, Z. (2017). Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine. *Energy conversion and management*, *133*, 427–443.
46. Teng, Z.-J., Lv, J.-L., & Guo, L.-W. (2018). An improved hybrid grey wolf optimization algorithm. In *Soft Computing* (pp. 1–15).
47. Shan, L., Qiang, H., Li, J., & Wang, Z.-Q. (2005). Chaotic optimization algorithm based on tent map. *Control and Decision*, *20*(2), 179–182.
48. Zhu, A., Chuanpei, X., Li, Z., Jun, W., & Liu, Z. (2015). Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3d stacked soc. *Journal of Systems Engineering and Electronics*, *26*(2), 317–328.
49. Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Vol. 1. MIT Press.
50. Rodríguez, L., Castillo, O., & Soria, J. (2016). Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3116–3123). IEEE.
51. Malik, M. R. S., Rasul Mohideen, E., & Ali, L. (2015). Weighted distance grey wolf optimizer for global optimization problems. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1–6). IEEE.
52. Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C. I., Martinez, G. E., Soto, J. (2017). A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing, 57*, 315–328.
53. Kishor, A., & Singh, P. K. (2016). Empirical study of grey wolf optimizer. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving* (pp. 1037–1049). Berlin: Springer.
54. Tawhid, M. A., & Ali, A. F. (2017). A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. *Memetic Computing, 9*(4):347–359, 2017.

55. Jitkongchuen, D. (2015). A hybrid differential evolution with grey wolf optimizer for continuous global optimization. In *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 51–54). IEEE.
56. Kamboj, V. K. (2016). A novel hybrid pso-gwo approach for unit commitment problem. *Neural Computing and Applications, 27*(6), 1643–1655.
57. Saremi, S., Mirjalili, S. Z., & Mirjalili, S. M. (2015) .Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications, 26*(5), 1257–1263.
58. Mahdad, B., & Srairi, K. (2015). Blackout risk prevention in a smart grid based flexible optimal strategy using grey wolf-pattern search algorithms. *Energy Conversion and Management, 98*, 411–429.
59. Singh, N., & Singh, S. B. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*.
60. Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing, 8*.
61. Zhang, S., & Zhou, Y. (2015). Grey wolf optimizer based on powell local optimization method for clustering analysis. *Discrete Dynamics in Nature and Society*.
62. Sahoo, A., & Chandra, S. (2017). Multi-objective grey wolf optimizer for improved cervix lesion classification. *Applied Soft Computing*, *52*, 64–80.

Coverage Optimization using Nature-Inspired Algorithm for Directional Sensor Networks

Hifzan Ahmad, Chaya Shivalingagowda, Narendra Kohli, and Dinesh Kumar Sah

1 Introduction

During the advancement of the technologies, wireless sensor networks (WSNs) end up being increasingly prominent nowadays, since even more predominant embedded platforms with high capacities are being structured at quickly diminishing expenses. For the most part, the sensor nodes are conveyed in an objective region to gather condition data for further preparation [1]. WSNs are currently assuming an undeniably critical role in ecological observing, debacle safeguarding, target following, industrial process control, battlefield observation, keen spaces, and so on. It is ordinarily made out of countless sensor nodes that have detecting, information preparation, and communication functionalities [2–5]. Depending upon the configuration of the reception apparatus, the sensors are probably grouped into two categories, directional sensors, and omnidirectional sensors. Conventional sensor networks usually expect the omnidirectional detecting model. The omnidirectional sensors have a round disk of detecting range [6]. The sensors outfitted with the directional reception apparatus are known as a directional sensor node [7].

H. Ahmad
A.P.J. Abdul Kalam Technical University (AKTU), Lucknow , India
e-mail: hifzan@gmail.com

C. Shivalingagowda
Kalsekar Engineering College, New Panvel Mumbai and GITAM university, Vizag, India
e-mail: chaya.ravindra@gmail.com

N. Kohli
Harcourt Butler Technical University, Kanpur , India
e-mail: kohli.hbti@gmail.com

D. Kumar Sah (✉)
Indian Institute of Technology (ISM), Dhanbad , India
e-mail: dksah.iitd@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_9

Unlike traditional omnidirectional sensors that dependably have an omni-angle of detecting range, the directional sensors most likely change toward a few directions and every direction possesses a constrained angle of detecting range [8]. Also, numerous techniques for traditional sensor networks are not appropriate for directional sensor networks (DSNs). Nonetheless, with the expanding request and progressively different applications, WSNs have a couple of obstacles in terms of data access, for example, within the image and video detecting applications [1].

In view of its adaptability with rotation and lower energy utilization, DSNs are typically examined within the literature. A DSNs have attracted expanding consideration in ongoing years because of their wide application prospects in framework observing, medicinal services checking, disaster response, and so on [8]. A DSNs are composed out of countless sensors that convey detected occasion reports to a sink. Directional sensors used in DSNs like cameras have some exceptional natures, for example, restricted detecting angle, directionality, and line-of-sight properties [28]. Not quite the same as omnidirectional WSNs, within which the detecting coverage essentially relies upon the detecting range (R_s) and the sensor location (X, Y), in DSNs, detecting coverage is influenced not just by the location and detecting range (R_s) yet in addition by the angle of view (AoV) and working direction. Besides, under the state of irregular deployment, the directional sensors are able to enhance the coverage more and more with updating its directionality property to change their working directions [1].

The directional sensor works toward a predefined direction for a provided period of time t. It can modify their working directions subject to the necessities of the application and this capability of the sensor is named motility. The coverage upgrading strategies abuse motility to restrain the impediment and overlapped regions. Moreover, it appeared differently in relation to the WSNs, and the sensors in DSNs have the potential of directional detecting, which ends in cheaper energy utilization and mutual interruption. The directional detecting and directional communication straightforwardly influence the coverage, availability of the network, and continuance of network [9]. The detecting coverage remains a basic issue in WSNs which exhibits how well the area is checked and fills in as a reason for applications, for example, physical phenomenon or target identification, grouping, and tracking. Because of the decent variety of sensor network applications, the idea of sensing coverage is liable to an extensive scope of interpretations. The coverage feature straightforwardly influences the deployment of the sensors and necessitates innovative coverage management plans for sensor networks [10].

In this chapter, nature-inspired algorithms namely particle swarm optimization (PSO) [45–47, 51–53] and memetic algorithm (MA) [48] have been presented to optimize the coverage in DSN. There are different issues identified with coverage, for example, area coverage, target coverage, and barrier coverage which are talked about in Sect. 3. This chapter centers around area coverage using PSO and target coverage optimization using MA.

2 Directional Sensor Network

As of late, sensor networks have pulled in huge research interests because of its immense potential applications. Be that as it may, numerous techniques for the traditional sensor networks are not appropriate for DSNs. In this way, a DSN likewise requests novel arrangements, particularly, for the sensor's scheduling and deployment approach.

Wireless sensors typically are available in two detecting shapes: directional and omnidirectional. The omnidirectional sensors comply with the circular detecting model, while directional sensors have sector-like detecting conduct. A sensor may have a restricted angle of detecting range because of the specialized imperatives or cost contemplation, which are meant by directional sensors, for example, infrared sensors, image sensors, and video sensors. The directional sensors may have many working directions which may be modified depending on the necessities of the application during their operations.

A DSN made out of a vast arrangement of directional sensors N that are deployed within the region of interest (ROI) to cover the region. A DSN includes a base station, to that, every sensor forwards their detected information in a multi-hop manner [11]. A DSN has been introduced concerning various applications, for example, traffic controlling [12, 13], underwater environmental surveillance [14, 15], and security checking [16]. A standard opinion for every such application is that every sensor can identify an occurrence happening within its detecting range.

The mobility property of a directional sensor is costly and progressively inclined to failure. In addition, the energy consumed in transmitting 1 Kb of information, 30 times more energy exhausted in moving a sensor just 1 m. Regardless of these obstructions, mobility expands the flexibility of the sensor network. In spite of the fact that mobility has a noteworthy enhancement for coverage, merely changing the orientation of the sensors does not offer the full coverage of the targeted region. To mend the coverage gaps, the coverage issues of DSNs need additional think about mobility [9].

2.1 Directional Detecting Model

An arbitrary directional sensor S is mostly portrayed by a 5-tuple $(X, Y, R, \text{LOS}, \delta)$, where (X, Y) signifies the coordinate of S, R imparts the detecting range, LOS is the sensing direction, and δ indicates the offset angle [17]. A target T_1 is affirmed to be secured by sensor S if and only if the distance between (X, Y) and T_1 is less than R, i.e., $d((X, Y), P1) \leq R$, and the angel between $(X, Y)P1$ and LOS is within [-δ, δ]. A directional detecting model is described in Fig. 1.

In numerous past works [18, 19], the detecting model of a directional sensor is generally viewed as a segment model. The region covered by a directional sensor is not just managed by the detecting range R, yet in addition, its detecting offset angle

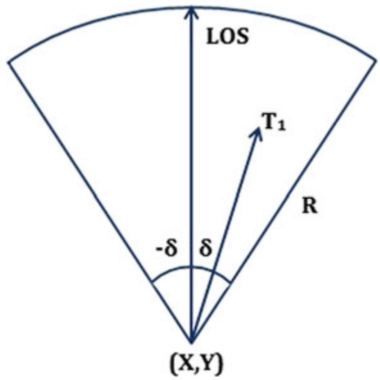


Fig. 1 Directional detecting model

2δ. Due to the segment like detecting the behavior of a directional sensor, an event that occurred in the segment observing region will be recognized by the sensor. In any case, an event happening at a separation $R + \epsilon$ cannot be identified by any stretch of the imagination, notwithstanding for a little ϵ value. A similar circumstance will happen for the extremely little angle value, which influences the execution of DSNs.

Figure 2a demonstrates a sensor S with segment model and four targets (T_1, T_2, T_3, T_4). Pursue the current models and deployment situations, targets T_2, T_3 and T_4 are often distinguished by the sensor S. Since the target T_1 belongs to outside of the segment, it cannot be distinguished by the sensor S. When the detecting direction of the sensor changes to a very little value, the coverage data will be not quite the same as the first, as represented in Fig. 2b. As the target T_1 enters the coverage region of the sensor, it tends to be identified by the sensor S. Notwithstanding, the target T_3 belongs to outside of the segment, so it cannot be distinguished by the sensor S. The little deviation of coverage can have an incredibly negative effect on the nature of the DSNs.

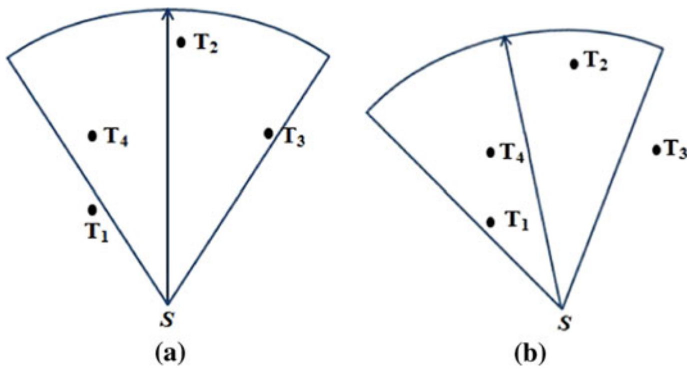


Fig. 2 Sector sensing model

2.1.1 Coverage Rate for Directional Sensing

It is very hard or unrealistic to cover the observed region completely whether or not the number of sensors deployed in the sensor network is extremely high. In this approach, it is a fundamental issue to reflect the way to make sure the given coverage rate of a sensor network for a given targeted region. Accept that the region of interest is ROI, and therefore the locations of randomly deployed sensors adopt uniform distribution. In this way, the probability p to cover the target region once N directional sensors are deployed is described in Eq. 1:

$$p = 1 - \left(1 - \frac{\delta R^2}{\mathrm{ROI}}\right)^N \tag{1}$$

The coverage rate p to deploy N sensors, for omni-detecting sensors with $\delta = \pi$, is definitely obtained by Eq. 1. In this manner, the number of deployed directional sensors if the coverage rate of the targeted region is a minimum of p is likewise gotten by Eq. 2.

$$N \geq \frac{ln(1-p)}{ln(\mathrm{ROI} - \delta R^2) - \ln\mathrm{ROI}} \tag{2}$$

2.2 *Coverage Rate for Sensing Adjustment*

The detecting expansions of numerous sensors, just as the directional sensor, are customizable. In the event that the coverage rate should be a presented value p now and again, the detecting range of the sensor can be modified to accomplish the objective in Eq. 3:

$$R = \sqrt{\frac{\mathrm{ROI}}{\delta}\left(1 - (1-p)^{\frac{1}{N}}\right)} \tag{3}$$

So on restricting the mean energy utilization in this manner broaden the lifespan of the sensor network, the N number of sensors can be isolated into n combinations, and these n combinations of sensors continue to operate alternatively. As indicated by some traditional model [20, 21], the energy utilization E of sensor is in extent to the k power of its detecting range R, i.e., $E = CR^k$, where C is the constant and $k \geq 2$.

3 Coverage Issues in DSN

In WSN, there exists a relation between coverage and continuance of network, if more sensors are involved in the network for coverage purpose then more energy is consumed and the continuance of network decreases. So, the continuance of the network cannot predict in advance manner, it depends on the operation and behavior of the network [22].

A fundamental issue getting an all-encompassing thought starting late is the issue of coverage, which fixates on how well the sensors screen the physical space they deployed. Coverage may be a key issue of the sensor network. It has pulled in a great deal of research thought on account of its association with the improvement of resources in an observing region [23].

The coverage amplification while keeping up a lower cost of deployment has reliably been an issue, especially when the sensing region is dark and possibly dangerous. A convincing technique for energy preservation in WSNs is the coverage deployment technique. There are expansive quantities of investigation about the coverage issue in omnidirectional sensor systems. The detecting coverage issue has been all around concentrated in omnidirectional sensor systems [24]. The coverage issue in DSNs is not quite the same as traditional omnidirectional WSNs due to line-of-sight (LOS), working direction, and narrow-angle of view [25].

As of late, remarkable directional sensors have ascended in view of the prerequisites of collecting methods, size, and cost. The restricted angle of view is the most perceiving trademark for the directional sensors. The detecting area of directional sensors is accepted to be the piece of a distinguishing circle with the range being proportional to the detecting range [26].

On the off chance that a sensor faces a direction in the occasion, we tend to express that the sensor works toward this direction, and thus the direction is called its working direction. Along these lines, the region covered with a directional sensor is managed by the two parameters it is working direction and location. The detecting range of the working direction of a sensor is known as its detecting region, at the point when a sensor works toward a direction. The detected region of various sensors might be covered or overlapped with one another when the random deployment of the sensor takes place. In this manner, there is a need to design sensors to face in specific directions to enhance the coverage administration in an arbitrary deployed sensor network, for instance, expand the region covered by the full network also fulfill the k-coverage of a couple of targets as well as region. In conventional coverage techniques, the omnidirectional sensor has simply two states: active or sleep. Regardless, the omnidirectional sensor cannot be associated with the directional sensor in light of the fact that a directional sensor can be in sleep state or work in certain directions [27].

Clearly, the chance of blind and overlapped areas would expand in the random deployment of directional sensors; thus, toward enhancing the overall coverage of the network, it is essential to modify the directions of these sensors deployed in ROI [28]. Since the arrangements of the coverage improvement issue in the omnidirectional sensor network cannot be connected to DSNs, numerous novel techniques are planned for maximal coverage of DSNs [29–31].

The issues of coverage are often generally delegated the area coverage, target coverage, and barrier coverage [25]. Coverage which is dependent on watching the entire region of interest (ROI) is known to be the area coverage, the target coverage fixates on checking only certain specific targets within the region, while a coverage which ensures to distinguish any interloper trying to go across the barrier in the area of intrigue is called the barrier coverage.

3.1 Area Coverage

A couple of studies [26] insinuate the area coverage as a field coverage. It is crucial for the DSNs to satisfy the predefined detecting tasks in order to enhance the area coverage. The main intention is to achieve the most extreme ROI while the detecting range of the sensors is limited. A few of the research works utilized the percentage of the region covered by the sensors to the overall ROI being a measurement toward the quality of coverage [32]. Nonetheless, some research analysis centers around the worst-case coverage, since the worst-case coverage goes for estimating a probability that an object would traverse a region or an occasion would occur without comprising recognized [33].

A couple of algorithms [34–37] are planned to maximize the covered region with a decrease of the overlapping. The analysts present one of the pioneer works and proposed a new technique dependent on a rotatable directional detecting model [38]. A technique to partition a DSN into a few segments referred to as sensing connected sub-graphs (SCSGs) has been proposed by them. The partitioning of a DSN into a few SCSGs is dividing and conquering a unified problem into a dispersed one, along these lines diminishing the time complexity. The execution of the area coverage is reflected by the number of SCSGs. The fewer number of SCSGs is, the more terrible the coverage rate progresses toward becoming, i.e., the more coverage gaps happen. In addition, to address the improvement of coverage problems, they demonstrate each SCSG as a multi-layered convex hull set. To acquire the maximal detecting coverage, the detecting directions of sensors are rotated once a multi-layer convex hull set in each SCSG has been formed. The directional sensors reposition themselves on the inverted direction of the interior angle bisector to accomplish the less overlapping region between two neighboring directional sensors on the same convex hull. The interior angle bisector is determined based on the location of the two neighbor sensors as appeared in Fig. 3.

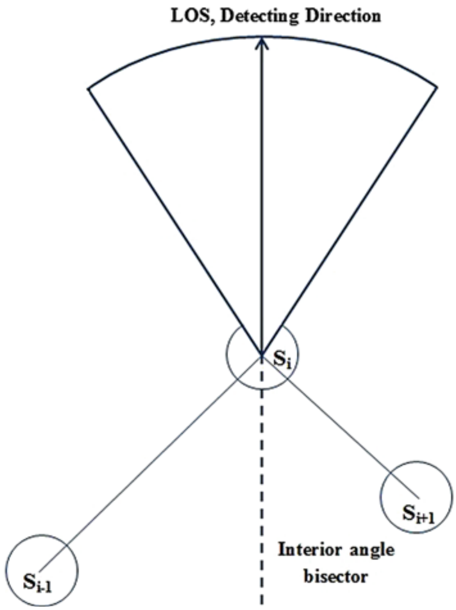


Fig. 3 Interior angle bisector

3.2 Target Coverage

Some of the sensor applications are just inspired by stationary target focuses, for example, structures, flags, boxes, and gates, though different applications go for following mobile targets like interlopers. The targets which are stationary in nature can be found anywhere in the observed region. The analysts have characterized target-based coverage issues to cover just the intrigued targets rather than the entire region. In a few investigations, the target coverage is named as a point coverage by the analysts [39]. In contrast to the area coverage, this problem sets an accentuation on the most proficient method to comprise the greatest number of targets. In this coverage, every target is checked constantly with something like one sensor. Be that as it may, some DSN applications may need in any event k sensors for every target so as to build some dependability of the network. The k-coverage issue has been planned dependent on this necessity.

A sensor is said to be in the active state if it utilizing anyone of its orientations to comprise any targets, while a sensor operates in a sleep state if it is not utilizing any one of its orientations. Target coverage with four different orientations is presented in Fig. 4. As described in Fig. 4a, the sensors S_1, S_2, and S_3 remain in active mode while sensor S_4 is in idle or sleep mode. Though, the coverage rate and active sensor rates are 70% and 75%, respectively. However, as depicted in Fig. 4b, all the four sensors are in active mode (active sensor rate is 100%) but the coverage rate is still 70%. It means that the selection policy of the sensor and its working direction or

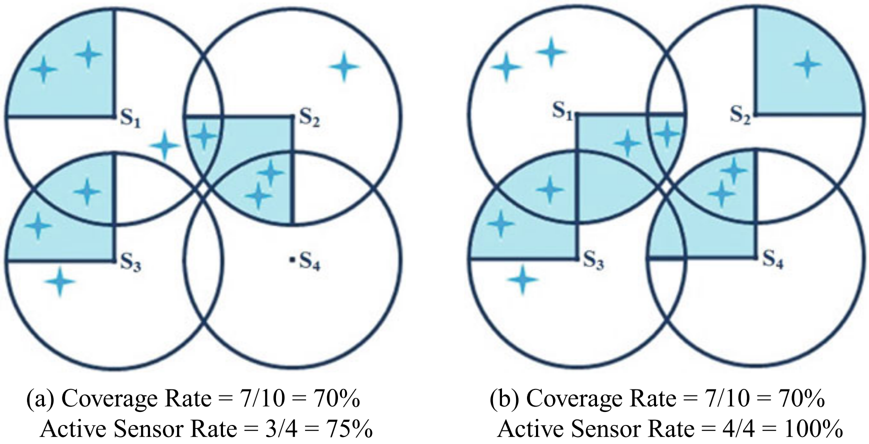


Fig. 4 Target coverage with four different orientation selection policies

orientation is optimal and energy-efficient in Fig. 4a as compared to the selection policy used in Fig. 4b.

The maximum coverage with minimum sensors (MCMS) issue has been proposed by the analysts [10]. Given a set of targets $T = \{t_1, t_2, ..., t_m\}$ and a set of N homogeneous directional sensors, every one of which has p conceivable orientations, MCMS goes for augmenting the number of covered targets while limiting the number of active directional sensors.

The two algorithms for the optimization of direction namely greedy direction adjusting (GDA) and equitable direction optimizing (EDO) have been introduced by the analysts [40]. The GDA algorithm optimized the directions as per the measure of covered targets, though the EDO algorithm changes the directions of sensors to comprise the basic targets and assigns detecting resources among sensors reasonably to limit the coverage difference between sensors. To limit the coverage overlapping, as appeared in Fig. 5, the equivalent coverage model has been introduced which states that all targets ought to be covered by just a single sensor, though they were covered by something like two sensors respecting coverage overlapping. The essential thought of EDO is assessing the use for every sensor through developing a target-direction mapping which contains the target number and the status of the target as regardless of whether being secured by the neighboring sensors. Rather than GDA, EDO improves the coverage on an average by 30%.

3.3 Barrier Coverage

Barrier coverage is a standout among the most imperative problems for different kinds of applications in sensor networks, for example, security observation, resource

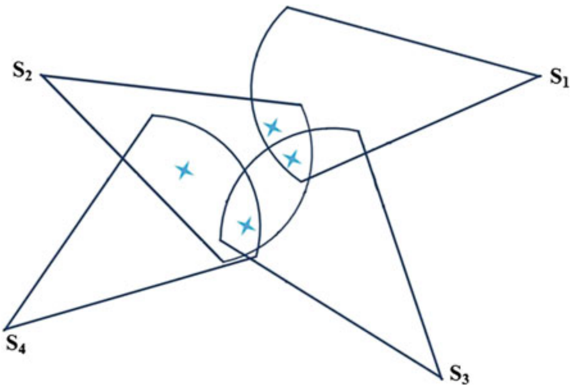


Fig. 5 Coverage overlapping occurs if a target is overlapped by more than one sensors

protection, and interloper discovery [41, 42]. In such kind of applications, the barrier coverage describes its ability to distinguish gatecrashers that endeavor to cross the area of intrigue. In the stationary sensor networks, when the number of sensors deployed is not sufficiently huge rather a few sensors utilized to make a barrier come up short on power, the barrier gaps may exist which enable the interlopers to pass through the undetected region. There are two exceptional methods to take care of this issue. One manner signifies to enlarge the number of stationary sensors, which brings approximately a ton of deployment expenses. The alternative manner implies deploying mobile sensors including thoroughly make the most sensor mobility to improve the barrier gaps, essentially outlined in Fig. 6.

If the number of stationary sensors is not able to make a barrier within the region of interest, then to improve the barrier coverage, the redeployment of the mobile sensors takes place. Subsequently, the deployment of mobile sensors for an assigned stationary sensor network is one of the major issues that demand to be resolved, which is characterized as a critical condition for mobile deployment (CCMD) issues by the

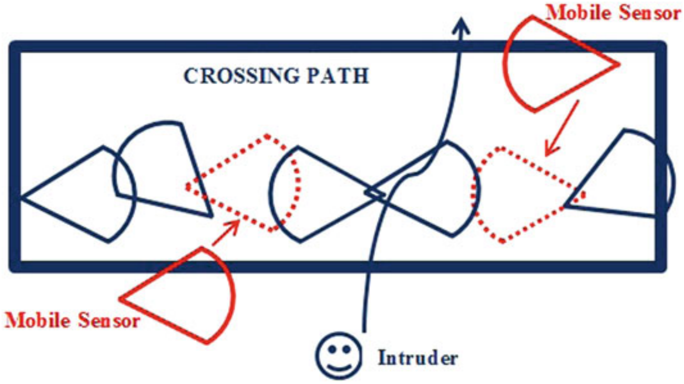


Fig. 6 Repairing of barrier gap by mobile directional sensors

analysts. The critical condition might be influenced by the deployment parameters of the DSN, for example, the density of deployment, the detecting radius of each sensor, and therefore the detecting angle of every sensor. Then again, most of the sensors have restricted sources of power, and because of some adversarial or inaccessible environments in lots of situations, the batteries of the sensors are difficult to interchange. Along these lines, building an energy-efficient barrier for DSNs is that the different issue needs to be resolved, which has been characterized as an energy-efficient barrier repair (EEBR) issue by the analysts. As appeared in Fig. 6, a barrier is shaped only when each mobile sensor is moving toward the preferred locations. The moving separation fundamentally decides to what extent the region of interest can be barrier covered. Hence, limiting the most separation travelled by any sensor can adjust the energy utilization among the sensors that extend the continuance of the network [44].

3.3.1 Weak Barrier and Strong Barrier

Weak barrier coverage simply desires to distinguish interlopers trying to move along the consistent crossing paths, whereas a strong barrier coverage has got to distinguish interlopers with discretionary moving paths, both are exhibited in Fig. 7a, b.

The main key issue concerning the weak barrier is to determine that the directional sensors is overlapping or not in horizontally. The key issue of the strong barrier is to decide overlapping at all of the directional sensors. Due to excess orientations and constraints, issues to directional sensors are difficult to justify than omnidirectional sensors. If Euclidean distance is less than or equivalent to $2r$, then two omnidirectional sensors are easily overlapped. If two or more directional sensors are closely one to another then they may not be overlap.

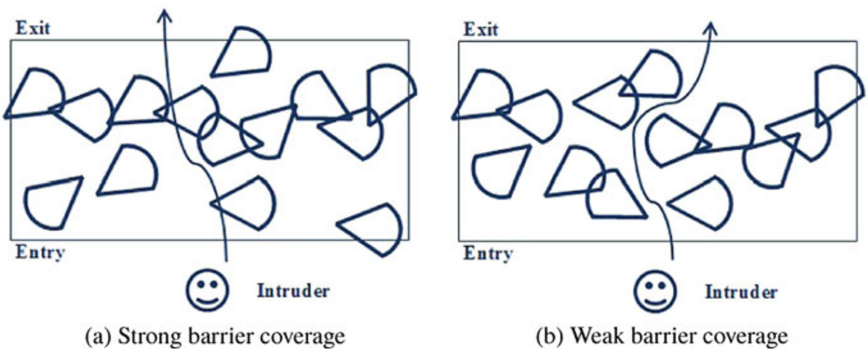


Fig. 7 Barrier coverage classification

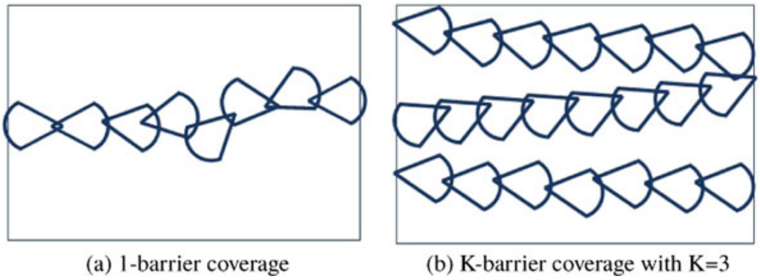


Fig. 8 1-barrier and k-barrier coverage

3.3.2 1-Barrier and K-Barrier

The capacities of DSNs border revealing method illustrate as k-barrier coverage property, and in this system, k designates a target infiltrating the network will be recognized with at least k distinct sensors before it leaves the observed region [43]. Clearly, 1-barrier denotes a unique instance concerning k-barrier while $k = 1$. Taking a look at the sensor deployment in Fig. 8, one can without much of a stretch find that the region is 1-barrier shrouded in Fig. 8a and region is 3-barrier canvassed in Fig. 8b.

4 Coverage Optimization

When we instate the network the sensors are deployed randomly, so the entire observing region is not constantly secured through that underlying deployment. Moreover, this remains pointless that all sensors are dynamic. The objective is to program the orientations to comprise the maximal region while actuating as few sensors as could reasonably be expected, calling as the optimal coverage issue in the directional sensor network. The aforementioned issue can be characterized as pursues.

Optimal coverage issues in DSNs: Given a predetermined region A, a set of directional sensors S, and every sensor with five parameters (X, Y, R, LOS, δ), discover a subset Z of Φ, with the limitation that at most one $\varphi_{i,\delta}$ can be picked for the same i (for example, a functioning sensor has just a single orientation), to expand the association of picked $\cup\varphi_{i,\delta}$ (for example, the covered region), while limiting the cardinality of $Z = \{\varphi_{i,\delta}|(i, \delta)$ is picked$\}$ (for example, the quantity of functioning directional sensors) where Φ is the arrangement of the coverage of all sensors, $(\Phi = \varphi_{i,\delta}|i = 1, 2, ..N, 0 \leq \delta \leq 2\pi)$, $\varphi_{i,\delta}$ is the coverage of S_i (i-th sensor) whose orientation is δ.

4.1 Particle Swarm Optimization (PSO) Algorithm for Area Coverage Issue

The deployment places of directional sensors fundamentally influence the coverage of the targets. The PSO is one of the multi-dimensional optimization algorithms [45]. It is a straightforward, efficient, and computationally effective optimization algorithm that is motivated by bird flocking and fish schooling and has high accuracy as well as a high convergence rate. The swarm of s potential arrangements is recognized as particles. A collection of particles moves in a search space wherever numerous conceivable solutions exist. The problem statement and the intended algorithm have been illustrated in the following section [46, 47].

4.1.1 Problem Definition

For a given region of interest (ROI) and the M number of targets $T = \{T_1, T_2, .., T_M\}$ monitored by the N number of sensors $S = \{S_1, S_2, .., S_N\}$, the primary goal is to discover the optimal locations from where the deployment of sensors S takes place in such a way that the coverage of targets should be maximum. In a situation, when the sensor S_i, $1 \leq i \leq N$ is at position (X_i, Y_i) and the target T_j, $1 \leq j \leq M$ is at position (X_j, Y_j), then S_i can sense the target T_j if and only if the gap between S_i and T_j is less than of the detecting range R, i.e., $d((X_i, Y_i)(X_j, Y_j)) \leq R$, and the angle between the $(X_i, Y_i)(X_j, Y_j)$ and LOS is within $[-\delta, \delta]$.

4.1.2 Proposed PSO Algorithm

The fundamental strides of the PSO algorithm have been portrayed in Algorithm 1. Every particle has a position vector x_{id} and a velocity vector v_{id} related to it. Notwithstanding this, every particle can in like manner remembers its very own best position distinguished up until this point and the best position that has been determined by the correspondence with its neighbors. The best value related to every particle is distinguished to be P_{best} and the best value covering every one of the particles is recognized to be G_{best}. A function that is utilized to assess every particle for confirming the nature of the arrangement is termed as the fitness function. To discover the positions of particles that outcome in the best assessment of the given fitness function is the objective concerning the PSO, every particle remains distributed by an arbitrary position and velocity to travel within the hunting space in the initialization procedure of PSO.

Algorithm 1: PSO Algorithm

initialization of particles;
while ($\overline{minimum\_error} || \overline{maximum\_iterations}$) **do**
  foreach *particles* **do**
    Determine the fitness value;
    (fitness value is better than P_{best}) P_{best} = fitness value;
  end
  Choose the particle by the best fitness value as the G_{best};
  foreach *particles* **do**
    Determine the velocity vector v_{id} of the particle as per velocity update equation (Eq. 4);
    Update the position vector x_{id} of the particle as per position update equation (Eq. 5);
  end
end

To achieve the global best arrangement, its personal best and global best are utilized to refresh the velocity v_{id} and position x_{id} by the accompanying equations.

$$v_{id}(t) = \omega.v_{id}(t-1) + c_1.r_1.(P_{best} - x_{id}(t-1)) + c_2.r_2.(G_{best} - x_{id}(t-1)) \quad (4)$$

$$x_{id}(t) = x_{id}(t-1) + v_{id}(t), 1 \leq i \leq n, 1 \leq d \leq D \quad (5)$$

where the parameters ω , $0 < \omega < 1$ are the inertia weight, $c_1, c_2, 0 \leq c_1, c_2 \leq 2$ are the acceleration coefficients, and r_1, r_2 , $0 \leq r_1, r_2 \leq 1$ are the randomly generated values. The refreshing procedure is rehashed until it is come to a satisfactory estimation of G_{best}. Subsequent to getting new refreshed position, the particle assesses the fitness function and refreshes P_{besti} just as G_{best} for the minimization issue as pursues (see Eqs. 6, 7, 8).

$$P_{besti}(t+1) = \begin{cases} P_{besti}(t) & f(x_i(t+1)) \leq f(P_{besti}(t)) \\ x_i(t+1) & f(x_i(t+1)) \geq f(P_{besti}(t)) \end{cases} \quad (6)$$

$$G_{best}(t) = \{P_{besti}(t) | f(P_{besti}(t))\} \quad (7)$$

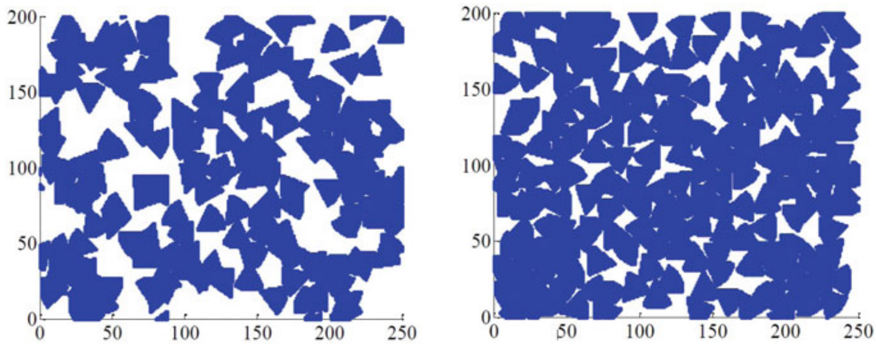
$$f(P_{besti}(t)) = max\{f(P_{best1}(t)), f(P_{best2}(t)), ..., f(P_{bestn}(t))\} \quad (8)$$

where $f(x)$ is the maximal fitness function. The global best is the association accumulation of personal best esteem every one of the particles in the swarm has encountered.

4.1.3 Experimental Result

The proposed calculation is executed utilizing MATLAB in various circumstances of DSN. The streamlining execution of PSO calculation including the impact of characteristic parameters will be checked through simulation analysis. A rectangular region of size $250 * 200$ is set to be the monitoring region and the quantity of the sensors is $n = 200$ have been taken in the simulation experiment. The sensing angle δ of the sensor is set to be $\delta = 30°$, and the detecting range R of sensor is $R = 20$, respectively. The locations of the sensor fulfill arbitrary normal distribution, moreover, the detecting direction of sensors is distributed randomly between $[0, \pi]$ initially. In the PSO calculation, the quantity of the particles is 100, the maximum angle of the molecule is $\epsilon_{\max} = 2\pi$, and the maximum iterations is $Maxnumber = 800$.

The complexity figures concerning directional sensor coverage, which is appeared in Fig. 9 has demonstrated the distinction of coverage before applying PSO calculation and in the wake of applying PSO calculation to optimize the coverage. The coverage percentage is low when the random deployment strategy of sensors takes place because of many overlapping regions and blind zones occur as demonstrated in Fig. 9a. Subsequent to utilizing PSO calculation to alter the detecting direction of sensors while the position of sensors does not transform, the region secured by directional sensors has expanded a great deal as displayed in Fig. 9b.



(a) Before optimization (b) After optimization

Fig. 9 PSO algorithm to optimize the coverage [46]

It is demonstrated by the simulation results that the utilization of PSO calculation can streamline the coverage of DSNs viably, so it is exceptionally effective to utilize the PSO calculation explained by the analysts to expand the coverage percentage of DSNs in a brief timeframe. The convergence rate of the PSO calculation is extremely great.

4.2 Memetic Algorithm for Target Coverage Issue

The number of directional sensors involved to screen every possible target within ROI is recognized as a cover set, including the condition that every directional sensor working in just a single detecting direction. A cover set may be characterized into a disjoint and non-disjoint cover set. A cover set is assumed to be disjoint if each sensor can be included in at the most a single cover set only, while a non-disjoint cover set overlooks the restriction and furnishes the chance to a sensor to have appeared into more than one cover set. A cover set to address the maximum set covers for DSNs (MSCD) issue in DSNs has been proposed by the analysts [49], which includes the expansion of network lifetime by allocating diverse times of execution to every cover set and the distinguishing proof of cover sets that may screen every possible target in an energy economical manner.

A memetic algorithm has been projected by the analysts for coverage optimization to solve the maximum set covers for DSNs (MSCD) issue [48]. The algorithm endeavors to discover maximum cover sets to drag out the continuance of the network. The answers for the issue are encoded as chromosomes and also the algorithm performs the simulation of the organic evolutionary process. The process of improvement, selection, and elimination is performed to enhance the arrangements until optimal solutions are acquired by the algorithm.

The number of solutions is encoded as chromosomes that may be demonstrated by various data structures as indicated by the features of the issue, described in line 5 of the algorithm. The initialization of N number of chromosomes is performed randomly as the preliminary population P. Line 6 of the algorithm is utilizing the fitness function to assess the number of solutions. The nearby improvement operations transform the chosen chromosome (lines 14–15). At that point, the algorithm attempts to locate a superior solution. In the wake of getting another chromosome, the worst solution in MP is wiped out. The process of improvement, selection, and elimination is repeated by the algorithm until the most extreme number of iterations is achieved (lines 10–19). Subsequent to achieving the most extreme amount of repetitions, the best solution in MP is taken as the final solution by the algorithm and the running time of the preferred sensors has been computed (lines 20–22). According to the solution, the energy of the sensors will be refreshed, and the number of sensors with no energy will be expelled from S (line 23). Line 4–24 is repeatedly performed by the algorithm until all the targets are not observed by the rest of the sensors at the same time.

Algorithm 2: Memetic Algorithm

Input: $DSN = (S, T)$ where S is set of sensors and T is set of targets, No. of detecting directions D;
Require: $S \neq \phi, T \neq \phi, D > 0$
$Runtime = 0, C = \phi$;
while *(every target T_i is covered by atleast one detecting direction of sensors)* **do**
  initial population of chromosomes in P;
  Calculate P;
  Repeat
  Choose the pareto-optimal chromosome in P and add it into the Mating Pool (MP)
  until the no. of chromosomes in MP reaches pts;
  while *(until termination()* **do**
    P_s = Select (MP);
    $P_t = \phi$;
    if *(!empty(P_s))* **then**
      P_t = Optimizer (P_s);
      P_t = Improver (P_s);
    end
    Calculate P_t;
    MP = Survival (MP, P_t);
  end
  P_{max} = MaxFitness(MP);
  $C = C \cup P_{max}$;
  $Runtime = Runtime + CalculateRuntime(P_{max})$;
  update sensor's lifetime in S;
end
return(C, $Runtime$);

The number of solutions is encoded as chromosomes, and every chromosome is outlined to a position in the solution space. The structure of a chromosome is described by a 1D array as appeared in Fig. 10 where N expresses the number of sensors in S (the set of sensors), $2N$ expresses the length of the 1D array, the odd places of the array expresses the sensor's ID, and the even places of the array expresses the detecting direction of the sensor individually. The positions of array 5 and 6 in Fig. 10 are signified as 4 and 3, individually, which demonstrates that sensor 4 works with detecting direction 3.

The number of cover sets held in a chromosome is determined by location 1 to N of the chromosome and joined by the targets that the sensors utilized currently cover. On position i, on the off chance that all targets in the set T are secured, at that point the sensors contain a cover set. At that point, the sensors that are utilized

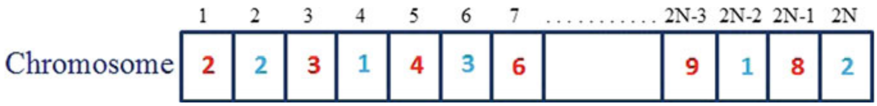


Fig. 10 Chromosome structure

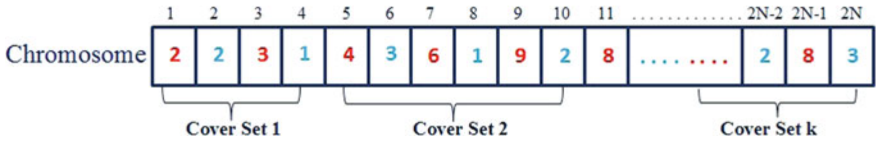


Fig. 11 Example of cover set

in the following cover set are recalculated from position $i+2$ and this procedure is iterated till $i > N$. As depicted in Fig. 11, a chromosome comprises k cover sets, and the sensors concerning any cover set can comprise all the targets.

The fitness function is essential to get an answer for assessment widespread of chromosomes. For the structure of a chromosome, as depicted in Fig. 11, the fitness function assesses a chromosome of the accompanying angles: the number of cover sets in the chromosome changes in the rest of the energy of sensors in a cover set and the number of unemployed sensors. The fitness function that is utilized in this investigation is demonstrated in Eq. 10.

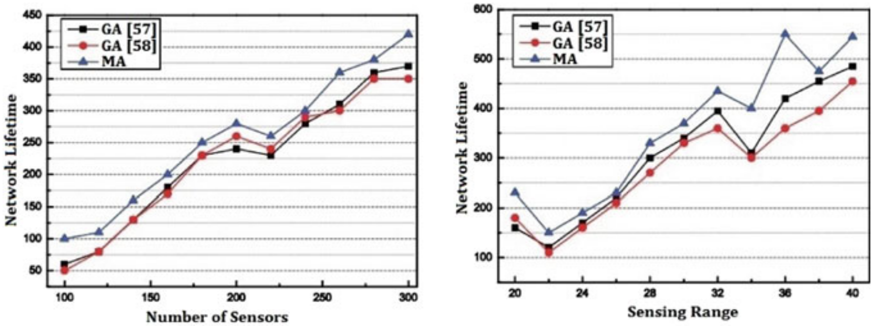
$$F(chromosome(i)) = \tau * CovNum + \epsilon * EVar + \varphi * UnSen, \tau + |\epsilon| + \varphi \quad (9)$$

$$= 1, (0 \leq \tau, \varphi \leq 1) \wedge (-1 \leq \epsilon \leq 0) \quad (10)$$

where CovNum is the number of cover sets, EVar is the change in the rest of the energy of sensors in a cover set, UnSen is the number of unemployed sensors, and τ, ϵ, and φ are constants.

4.2.1 Experimental Result

The proposed calculation has been researched utilizing MATLAB programming. It is assumed that the region of interest (ROI) to be observed is 100 m $*$ 100 m fixed region, where 30 targets are deployed randomly. It is assumed that all the directional sensors have three detecting directions, and every sensor detecting only one of these detecting directions. The detecting radius of all the directional sensors are 20 m. The algorithm is differentiated with the genetic algorithm (GA) [49, 50] for the characteristics of network lifetime. The average estimation of the trials has been taken as the consequence of every algorithm. The exploratory outcomes demonstrate that the suggested algorithm beats the GA [49, 50] as far as expanding the network lifetime.

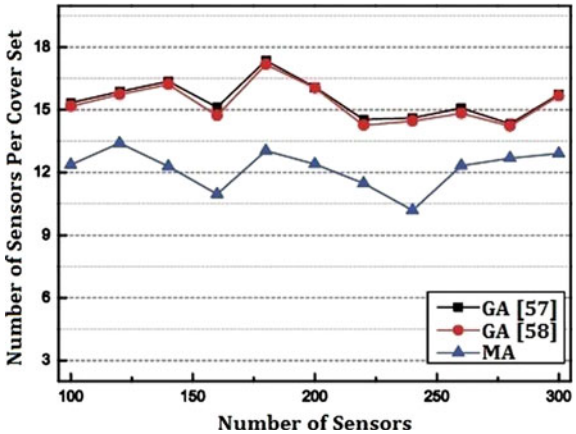


(a) When the no. of sensors varies from 200 to 300

(b) When detecting range varies from 20 m to 40 m

Fig. 12 Impact on network lifetime [48]

Fig. 13 Number of sensors per cover set [48]



When the number of sensors differs from 100 to 300, as described in Fig. 12a, the network lifetime acquired by these algorithms increases alongside the sensors deployed in the ROI. Memetic algorithm beats GA [49, 50] in broadening the network lifetime. At the point, when the number of sensors in ROI is thought to be 150 and the detecting range of sensors differs from 20 m to 40 m, as appeared in Fig. 12b, the acquired continuance of the network by MA is longer than that of GA [49, 50]. The number of sensors utilized by MA per cover set is lesser than that of the two alternative algorithms. This result demonstrates that the MA has a superior decision methodology. The average number of sensors utilized by these algorithms per cover set is demonstrated in Fig. 13 under the different number of sensors.

5 Conclusion

The chapter presents the effectiveness of DSNs and its optimization from a coverage point of view. Two proposals that are based on PSO and MA have been exhibited. In PSO-based approach, every particle can likewise remember its own best location distinguished up until this point, and furthermore the best location that has been determined by the correspondence with its neighbors. A fitness function is utilized to assess every particle for confirming the nature of the arrangement. To discover the positions of particles that outcome in the best assessment of the given fitness function is the objective of the PSO, every particle is distributed by an arbitrary location and swiftness to relocate in the hunting space within the initialization procedure of PSO. A mimetic algorithm has been projected by the analysts for coverage optimization to solve the maximum set covers for DSNs (MSCD) issue. The algorithm endeavors to discover maximum cover sets to drag out the continuance of the network. The answers for the issue are encoded as chromosomes and also the algorithm performs the simulation of the organic evolutionary process. The process of improvement, selection, and elimination is performed to enhance the arrangements until optimal solutions are acquired by the algorithm.

References

1. Zhang, G., You, S., Ren, J., Li, D., & Wang, L. (2016). Local coverage optimization strategy based on Voronoi for directional sensor networks. *Sensors*, *16*, 2183.
2. Akyildiz, I., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, *38*(4), 393–422.
3. Das, S. K., & Tripathi, S. (2018). Adaptive and intelligent energy efficient routing for transparent heterogeneous Ad-hoc network by fusion of game theory and linear programming. *Applied Intelligence*, *48*(7), 1825–1845.
4. De, D., Mukherjee, A., Kumar Das, S., & Dey, N. (2020). Wireless sensor network: Applications, challenges, and algorithms. In *Nature Inspired Computing for Wireless Sensor Networks* (pp. 1-18). Singapore: Springer.
5. Ahmad, H., & Kohli, N. (2015). A survey on various static and mobile base stations and wireless charging of sensor nodes in WSNs. In *2015 International Conference on Communication Networks (ICCN)*, Gwalior (pp. 16–22).
6. Yan, R., Si-Dong, Z. H., et al. (2006). Theories and algorithms of coverage control for wireless sensor networks. *Journal of Software*, *17*(3), 422–433.
7. Chen, U.-Re., Chiou, B.-S., Chen, J.-M., & Lin, W. (2009). An adjustable target coverage method in directional sensor networks. In *2008 IEEE Asia-Pacific Services Computing Conference*, Yilan (pp. 174–180).
8. Nur, F. N., Sharmin, S., Habib, M. A., Razzaque, M. A., & Islam, M. S. (2016). Collaborative neighbor discovery in directional wireless sensor networks: Algorithm and analysis. *EURASIP Journal on Wireless Communications and Networking, 119*. https://doi.org/10.1186/s13638-017-0903-6.
9. Güvensan, A., & Gökhan Yavuz, A. M. (2011). On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks, 9*(7), 1238–1255. https://doi.org/10.1016/j.adhoc.2011.02.003.

10. Ai, J., & Abouzeid, A. (2006). Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, *11*(1), 21–41.
11. Kezhong, L., & Ji, X. (2009). A fine-grained localization scheme using a mobile beacon node for wireless sensor networks. *2009 2nd International Conference on Computer Science and Its Applications* (pp. 1–6). Korea (South): Jeju.
12. Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer Networks*, *51*(4), 921–960.
13. Yu, Z., Yang, F., Teng, J., Champion, A., & Xuan, D. (2015). Local face-view barrier coverage in camera sensor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, Kowloon (pp. 684–692).
14. Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks*, *3*(3), 257–279.
15. Jiang, J., Han, G., Shu, L., Chan, S., & Wang, K. (2017). A trust model based on cloud theory in underwater acoustic sensor networks. *IEEE Transactions on Industrial Informatics*, *13*(1), 342–350.
16. Tao, S., Kudo, M., Pei, B. N., Nonaka, H., & Toyama, J. (2015). Multiperson locating and their soft tracking in a binary infrared sensor network. *IEEE Transactions on Human-Machine Systems*, *45*(5), 550–561.
17. Lin, T., Santoso, H. A., Wu, K., & Wang, G. (2017). Enhanced deployment algorithms for heterogeneous directional mobile sensors in a bounded monitoring area. *IEEE Transactions on Mobile Computing*, *16*(3), 744–758.
18. Wang, Z., Liao, J., Cao, Q., Qi, H., & Wang, Z. (2014). Achieving k-barrier coverage in hybrid directional sensor networks. *IEEE Transactions on Mobile Computing*, *13*(7), 1443–1455.
19. Yildiz, E., Akkaya, K., Sisikoglu, E., & Sir, M. Y. (2014). Optimal camera placement for providing angular coverage in wireless video sensor networks. *IEEE Transactions on Computers*, *63*(7), 1812–1825.
20. Akyildiz, I., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, *40*(8), 102–114.
21. Meguerdichian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2001). Coverage problems in wireless Ad Hoc sensor networks. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213),* Anchorage, AK, USA (Vol. 3, pp. 1380–1387).
22. Liu, C., & Cao, G. (2011). Spatial-temporal coverage optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, *10*(4), 465–478.
23. Zhu, C., Zheng, C., Shu, L., & Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. *Journal Network and Computer Applications*, *35*(2), 619–632.
24. Bang, W. (2011). Coverage problems in sensor networks: A survey. *ACM Computing Surveys, 43*(4), 32:1–32:53. https://doi.org/10.1145/1978802.1978811.
25. Islam, M. M., Ahasanuzzaman, M., Razzaque, M. A., Hassan, M., Alelaiwi, A., & Xiang, Y. (2015). Target coverage through distributed clustering in directional sensor networks. *EURASIP Journal on Wireless Communications and Networking, 167* (2015). https://doi.org/10.1186/s13638-015-0394-2.
26. Wang, J., Niu, C., & Shen, R. (2009). Priority-based target coverage in directional sensor networks using a genetic algorithm. *Computers & Mathematics with Applications*, *57*(11–12), 1915–1922.
27. Cheng, W., Li, S., Liao, X., Changxiang, S., & Chen, H. (2007). Maximal coverage scheduling in randomly deployed directional sensor networks. In *2007 International Conference on Parallel Processing Workshops (ICPPW 2007)*, Xian (pp. 68–68).
28. Varposhti, M., Saleh, P., Afzal, S., & Dehghan, M. (2016). Distributed area coverage in mobile directional sensor networks. In *2016 8th International Symposium on Telecommunications (IST),* Tehran (pp. 18–23).
29. Peng, S., Xiong, Y., Wu, M., & She, J. (2017). A new method of deploying nodes for area coverage rate maximization in directional sensor network. In *IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing (pp. 8452–8457).

30. Zhang, J. W., Li, N., Wang, N. W, Y., Shi, J. (2016). A coverage algorithm based on D-S theory for directional sensor networks. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1177/1550147716669623.
31. Mishra, S., Sharma, R., & Saxena, S. (2016). The issue of coverage in directional sensor network. *International Journal of Computer Application*, *115*, 17–20.
32. Huang, C. F., & Tseng, Y. C. (2005). The coverage problem in a Wireless Sensor Network. *Mobile Networks and Applications*, *10*, 519–528.
33. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. (2001). Coverage problems in Wireless Ad-Hoc sensor networks. In *Proceedings of IEEE (INFOCOM'01). Conference on Computer Communications* (Vol. 3, pp. 1380–1387).
34. Li, J., Wang, R. C., Huang, H. P., Sun, L. J. (2009). Voronoi based area coverage optimization for directional sensor networks. In *2009 Second International Symposium on Electronic Commerce and Security*, Nanchang (pp. 488–493).
35. Ma, H., Zhang, X., & Ming, A. (2009). A coverage-enhancing method for 3d directional sensor networks. In *IEEE INFOCOM 2009*, Rio de Janeiro (pp. 2791–2795).
36. Zhao, J., & Zeng, J. C. (2009). An electrostatic field-based coverage-enhancing algorithm for wireless multimedia sensor networks. *2009 5th International Conference on Wireless Communications* (pp. 1–5). Beijing: Networking and Mobile Computing.
37. Kandoth, C., & Chellappan, S. (2009). Angular mobility assisted coverage in directional sensor networks. *2009 International Conference on Network-Based Information Systems* (pp. 376–379). IN: Indianapolis.
38. Tao, D., Ma, H., Liu, L. (2006). Coverage-enhancing algorithm for directional sensor networks. In J. Cao, I. Stojmenovic, X. Jia, S. K. Das (Eds.) *Mobile Ad-hoc and Sensor Networks. MSN 2006. Lecture Notes in Computer Science* (Vol. 4325). Berlin: Springer. https://doi.org/10.1007/11943952_22.
39. Cardei, M., Thai, M., Li, Y., & Wu, W. (2005). Energy-efficient target coverage in wireless sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL (Vol. 3, pp. 1976–1984).
40. Wen, J., Fang, L., Jiang, J., & Dou, W. (2008). Coverage optimizing and node scheduling in directional wireless sensor networks. *2008 4th International Conference on Wireless Communications* (pp. 1–4). Dalian: Networking and Mobile Computing.
41. Balister, P., Bolloba's, B., & Sarkar, A. (2016). Barrier coverage. *Random Structures & Algorithms*, *49*(3), 429–478.
42. Tao, D., & Wu, T. Y. (2015). A survey on barrier coverage problem in directional sensor networks. *IEEE Sensors Journal*, *15*(2), 876–885.
43. Ssu, K., Wang, W., Wu, F., & Wu, T. (2009). K-barrier coverage with a directional sensing model. *International Journal on Smart Sensing and Intelligent Systems*, *2*(1), 75–93.
44. Zhao, L., Bai, G., Jiang, Y., Shen, H., & Tang, Z. (2018). Strong barrier coverage of directional sensor networks with mobile sensors. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1177/1550147718761582
45. Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics & Computation, 274*, 292–305.
46. Peng, S., Xiong, Y., Wu, M., & She, J. (2017). A new method of deploying nodes for area coverage rate maximization in directional sensor network. In *IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing (pp. 8452–8457).
47. Singh, P., Mini, S., & Sabale, K. (2016). Particle swarm optimization for the deployment of directional sensors. In B. Panigrahi, P. Suganthan, S. Das, S. Satapathy (Eds.), *Swarm, Evolutionary, and Memetic Computing*. SEMCCO. Lecture Notes in Computer Science (Vol. 9873). Cham: Springer.
48. Wang, A., et al. (2016). A novel multi-objective coverage optimization memetic algorithm for directional sensor networks. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1177/1550147716657923.
49. Gil, J. M., & Han, Y. H. (2011). A target coverage scheduling scheme based on genetic algorithms in directional sensor networks. *Sensors (Basel, Switzerland)*, *11*(2), 1888–1906.

50. Mohamadi, H., Salleh, S., Ismail, A. S., et al. (2015). Scheduling algorithms for extending directional sensor network lifetime. *Wireless Network*, *21*, 611–623.
51. Das, S. K., Samanta, S., Dey, N., & Kumar, R. (2020). *Design frameworks for wireless networks*. Berlin: Springer.
52. De, D., Mukherjee, A., Kumar Das, S., & Dey, N. (2020). *Nature inspired computing for wireless sensor networks*. Berlin: Springer.
53. Binh, H. T. T., Hanh, N. T., Van Quan, L., et al. (2018). Improved Cuckoo Search and Chaotic Flower Pollination optimization algorithm for maximizing area coverage in Wireless Sensor Networks. *Neural Computing and Applications*, *30*(7), 2305–2317.

Security and Management

Flower Pollination Optimization-Based Security Enhancement Technique for Wireless Sensor Network

Ranjit Kumar, Sachin Tripathi, and Rajeev Agrawal

1 Introduction

WSNs are quickly increasing popularity because of the fact that they are almost certainly affordable strategies to a selection of real-world complications [1]. It is provided by low price to rearrange excellent sensor component arrays for completing within a selection of scenario adept for the job of every military and civilian. You will find restricted information storage space as well as strength that make it accountable to begin stringent learning resource limits within this sensor system. Each of these matches to chief hindrances toward the setup of typical protection ways of personal computer with this system. The protection on the system turns into harder because of the unforeseen correspondence channel as well as unattended process. In reality [2], wireless receptors routinely have the processing qualities of devices which are many decades outdated (or maybe longer), and also the manufacturing inclination is diminishing the cost of wireless receptors while keeping associated computing energy. By bearing in mind the idea of improving the system features of processing together with the big energy usage, a great deal of scientists had labored on it while simultaneously safeguarding them against unauthenticated people. Most parts on

R. Kumar (✉) · S. Tripathi
Indian Institute of Technology (Indian School of Mines) Dhanbad, Dhanbad, Jharkhand 826004, India
e-mail: ranjit_kumaruitbu@yahoo.co.in

S. Tripathi
e-mail: var_char1285@yahoo.com

R. Kumar
GCET, Greater Noida 201306, India

R. Agrawal
GL, BAJAJ Institute of Technology and Management, Greater Noida 201306, India
e-mail: rajkecd@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_10

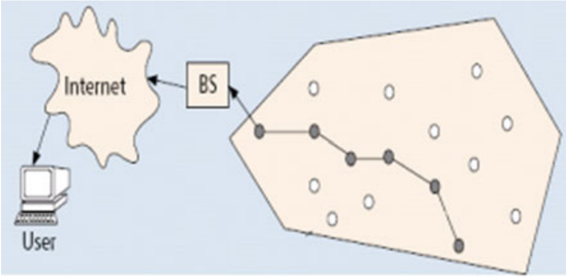


Fig. 1 Wireless sensor network

the WSN will be inspected, which includes provided as well as able routing, group arrangement, data aggregation and so forth.

Along with all that standard protection applies to, we look at that a number of general-purpose sensor system methods assumed that almost all nodes are trustworthy and cooperative. This is not true for many, or perhaps a lot of, real-world wireless sensor social networking programs, and they necessitate a certain volume of loyalty with inside the application program consequently 1 may keep appropriate gadget practical use. Experts within this method began focusing on setting up a sensor loyalty design to deal with the problems past the cap of cryptographic protection. Furthermore, at this time there are numerous assaults meant to misuse the untrustworthy corresponding routes as well as unattended functioning of WSNs. Moreover, due to the organic unattended typical for WSNs, we put up which actual physical assaults to receptors believe a big component within the functioning of WSNs (Fig. 1).

As a result, with this paper, it contains found-level conversation of actual physical strikes as well as their corresponding defenses, subjects usually dismissed in the majority of the present exploration on sensor safeguards. We arrange the basic periods of WSN protection directly into 4 basic principle classes: the problems for sensor manage, protected, the essentials of a secure WSN, assaults and guarded steps. The connection next just captures after this particular grouping [3].

A set of sensor nodes launched particularly environment entails a WSN. These are for probably the most component used as a part of an assortment of army programs what about buildup within therapeutic programs. Therefore, protection is a basic principle problem in these kinds of a fine selection.

Cryptography is an incredibly important exercise for info protection of WSN. Based on crucially used, this particular method is purchased in 2 sorts: asymmetric encryption and symmetric encryption. The 2 devices enjoy a couple of reasons for fascination as well as furthermore a couple of weak points. Symmetric crucial cryptography continues to be fast around procedure still when the equivalent key element must be discussed amongst beneficiary and sender so protection on this key element is a difficult job, whereas asymmetric crucial cryptography resolves the problem of protected exchange of key element, the way it is almost average compared to symmetric crucial cryptography. Symmetric crucial cryptography continues to be fast around procedure just yet as identical key element is must be discussed amongst sender as well as receiver, and therefore protection on this key element is a difficult job, whereas asymmetric crucial cryptography resolves the trouble of protected alter of key element, though it is comparatively easy compared to symmetric crucial cryptography [4].

2 Related Work

In last few decades, several works are proposed in the context of wireless network along with wireless sensor network, cloud computing and Web applications with network [5–12]. Hsueh et al. [13] in this particular newspaper propose a two-level protected transmission pattern. The hash chain is utilized by this plan to create the powerful period key element that could be used for mutual authentication together with the symmetric encryption key element. The primary computations of powerful time key element will be the hash features, for instance, SHA-1 or MD5, that happen to be incredibly fundamental as well as fast. By matching with MAC process, at this time there are not any extra package contrasted and also the present MAC programs. The two-level strategy is able to verify as well as interrupt about the episodes during several examination concentrates. The blend of lower multifaceted dynamics protection progression as well as countless inspection concentrate setup would be safeguard in a position to form strikes as well as transmit the sensor nodes to rest method about the earliest ability. The protection evaluation shows this program is able to fight the replay episode as well as forge encounter, as well as the big energy exploration shows this electricity strategy is able too. The entire power division of power exploration in addition shows yet another possible choice guideline to negotiation the requirements in between power preservation and protection pattern. Bhave et al. [14] this particular undertaking largely concentrated about the improvement on the crossbreed encryption program that merges asymmetric and symmetric encryption algorithms for protected Key interchange plus more proper cipher textual content safety measures This newspaper reviews on the evaluation of total functionality of keyword phrases of little mistake cost for symmetric, Hybrid and asymmetric encryption systems used within WSNs. Test results suggest reduced with little mistake price through the use of crossbreed encryption pattern as in comparison with asymmetric and symmetric systems on its own. Increased volume of receptors additional brings down little errors fee as well as prior general performance. Alamouti codes with space precious time obstruct codes are very broadly used transmission mechanism, for space was extended by WSN, and precious time block codes (ECBSTBC) have much better signal-to-sound ratio alter when contrasted as well as sensor having determination conspire. Suggested product makes use of ECBSTBC codes for transmission. Abuhelaleh et al. [15], within this particular paper, focus on exactly how to reach the highest prospective degree of safeguard by making use of novel crucial managing device which could be utilized during WSNs marketing and sales communications. For the proposal of ours to become applicable and efficient more to a great level of wireless sensor system programs, we work with a unique sort of structure which was recommended toward the bunch hierarchy of WSNs so we choose just about the most fascinating protocols which have been recommended for this particular kind of preparation, that is, LEACH. This suggestion is part of an entire plan that we have produced covering each among the components of WSNs correspondence that is architectural for WSN (SOOAWSN). Lai et al. [16], BROSK is recommended by the creators (communicate Session Key negotiation protocol). With BROSK, every node declares an idea alongside the nonce of it. Thus, every single neighboring node which listens one another can actually compute a typical element that is an attribute of the nonces of theirs. Neighboring nodes authenticate themselves using a predeployed element and that is said to be unavailable within the situation the node is restricted. Liu et al. suggest in [17] location-based keys (LBKs) that depends on place information to achieve crucial managing. The secrets are put in place as per the geographical place of sensor nodes. Be that here as it might, understanding the geological part of nodes is not guaranteed with arbitrary usage. Eschenauer as well as Gligor [18] suggest a method started within an arbitrary element pre-appropriation. With this strategy, every sensor arbitrarily picks an amassing of secrets as well as the identifiers of theirs originating from a vital swimming pool just before transmitting. Next, a shared key detection phase is released exactly where 2 friends and neighbors exchange and also assess a summary of identities of secrets within the key chains of theirs. Really, every single sensor node broadcasts an email and also gets a single idea coming from each node within its stereo assortment the best place communications offer crucial identification prospect lists. And so, any kind of set of nodes features a special opportunity to at a minimum discuss the main typical element. The task on this system is finding a great trade-off in between the dimensions of crucial swimming pool as well as how many secrets stashed by nodes to achieve the right likelihood. The leading drawback of this method is the fact that when the volume of jeopardized nodes improves, the part of affected back links likewise improves. When it comes to [19], the experts focus on producing cost-saving approaches while weakening the risk shown. Key Infection is proposed by them, a light weight protection process ideal for appointing within noncritical product sensor networks the best place, an assailant is able to filter easiest a lasting percent of chat stations. Typical, present symmetric cryptographic ways for WSNs target chiefly over the usefulness of organization that is key following the deployment on the system. Nevertheless, they do not cope with crucial refresh, and that tends to make crucial business dynamic and also gives another intricacy on the process of assailants. In addition, symmetric fixes do not scale nicely if the quantity of sensor nodes goes up, as well as overlook the result of shot node strikes. Making use of symmetric cryptographies in deep, a software application setup is difficult. Since they are not offering an excellent trade-off between performance and resilience, along with aggressive dynamics locations in which sensor nodes are deployed cause it to be at the mercy of an assortment of strikes. When it comes to [20], Gura et al. article that each elliptic curve cryptography and RSA are practical for small devices with no gear to enhance velocity. With 8-bit CPUs, ECC exhibits a delivery edge over RSA. Another convenient job is the fact that ECC's 160-bit element end result within smaller mail messages amid transmission contrasted together with the 1024-bit RSA secrets. Particularly, Gura et al. show which ECC stage duplication on small devices is very similar for delivery to RSA receptive crucial activities along with a petition of dimension speedier compared to RSA private key activities. When it comes to [21], Watro et al. screen which slice on the RSA cryptosystem can easily additionally be very just connected to certain wireless receptors. The TinyPK framework depicted by [42] is meant to permit verification as well as understanding that is key between advantage-constrained receptors. The process is used along with the present symmetric encryption, advantage for node methods, for instance, TinySec. Particularly, they actualized the RSA receptive activities on the receptors and also the RSA personal activities to an exterior gathering, for instance, a tablet. When it comes to [22], Malan et al. showcase a working utilization of Diffie Hellman found perspective of the ECDLP. Nevertheless, they show which public secrets could be manufactured in 34 s, which secrets that are shared could be conveyed amongst nodes of a sensor system within the exact same, making use of few a lot more crucial compared to one kilobyte of SRAM as well as 34 kilobytes of ROM. Thus, the public crucial base is doable about the MICA2 for uncommon blood circulation of shared insider specifics. Wang et al. when it comes to [23] propose a public major for WSNs. They constructed an ECC-based entry that includes a pairwise crucial base, nearby gain access to management, along with remote entry management. They have played away a comparability examination by actualizing both symmetric key and public crucial primitives on Hp and MICAz nodes iPAQ. The contextual analysis of theirs shows the common element program tends to be more lucrative compared to symmetric crucial as much while the mind utilizes and notes intricacy, security and strength. Liu and Ning [24] in addition underline which ECC is a standout among the best types of public major cryptography of WSNs. The way of plan, assessment and execution of TinyECC, an adaptable and configurable library for ECC functions in WSNs, are shown. The library provides a variety of development changes which may be joined up with through the engineer's demands for a particular program, taking concerning a variety of delivery situations as well as source utilizations. The TinyECC library was the same evaluated on several sensor stages, such as MICAz, Tmote Sky and also Imotel, to find essentially the most computationally successful and additionally probably the most storage space successful styles.

3 Cryptography in WSNs

Cryptography is definitely the science whenever using the arithmetic to encode as well as unscramble info. Cryptography, study and craftsmanship of obtaining all set coded or even made certain interchanges likely to become intelligible simply toward the person setting up the key [25]. Cryptography refers both with the process or maybe capability of imparting inside or even interpreting secretes functions and also toward the utilization of coder to alter more than mechanized info therefore only a specific beneficiary will likely have the capability to peruse it making use of key element. Figure 2 shows the traditional cryptography framework.

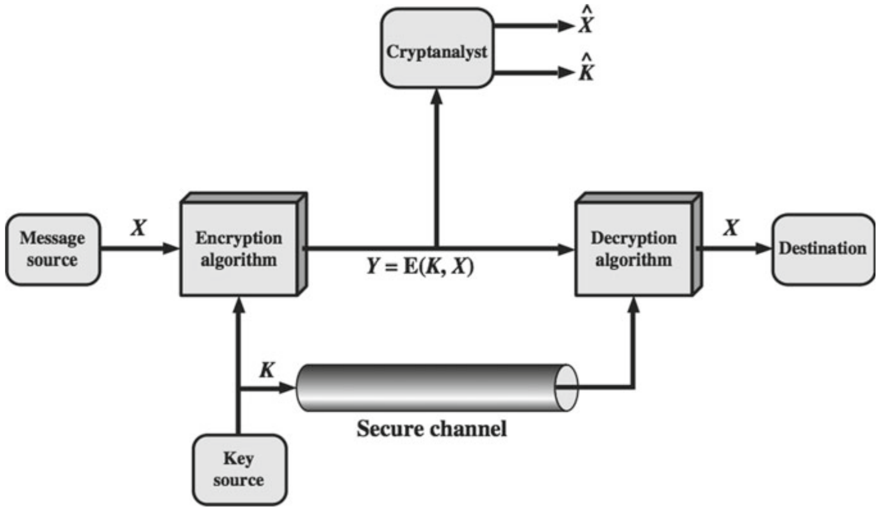


Fig. 2 Cryptography

A distinctive correspondence of the apparent textual content or maybe basic textual content is contacted by cryptographers. When the revolutionary correspondence has become scrambled or even enciphered, the final result is determined when the cipher textual content or even cryptogram [26]. Cryptography is vital for one apart from secrecy; however, cryptography shields the planet's conserving cash frameworks too. WSN is utilized in numerous severe uses such as army and habitat monitoring. Least degree of protection as integrity as well as authentication is necessary for many programs, thanks to the sensitive nature of theirs of WSN. The beneath figure exhibits the type of cryptography.

1. **Cryptanalysis**

It is the craft of evaluation cipher textual content to sort the key or the plaintext. Through the day's conclusion, cryptanalysis would be the complete opposite of cryptography. It is the breaking up of cipher, and comprehending the procedure of code pauses is really important when developing the encryption phone. The 3 cryptographic styles are:

2. **Secret key cryptography or symmetric key cryptography**

A singular element for equal encryption as well as decryption relies on by symmetric key cryptography [27]. It may be meant which creates a top price of information throughput. Secrets of symmetric figure are extremely scant. These figures are designed to create powerful ciphers. The drawbacks of secret crucial cryptography will be in 2 collecting correspondence frameworks, and the main element should be discussed through the sender as well as the receiver.

Desirable earlier and also requires one and hidden secret key element for equally encryption and also decryption of the information packets inside a communicating system that is maintained as and hidden secret within a system [28].

The additional purchase of symmetric crucial algorithms is

1. Block figures for settling modifications.
2. Stream figures for period of different changes.

These 2 subdivisions are used to consider encryption computations on plain mail messages in an assortment of amounts, as an example, remarkable info kinds, electric battery balance utilization details, and sizes are blocked by distinctive data, for just a range of primary measurements as well as various encryption/decryption rates of speed (Figs. 3, 4 and 5).

3. **Public key cryptography or asymmetric cryptography**

It runs on a particular secret for encryption as well as a different secret for unscrambling. Distinctiveness, the personal element, has to be secret. Within a big community, the amount of secrets in essence may be much less compared to within the symmetric crucial circumstance [29]. For wide daytime crucial cryptography depending on the technique for using, it remains unaltered for substantial precious time frame. The drawback is essential sizes are ordinarily drastically larger compared to all those necessary for symmetric crucial encryption.

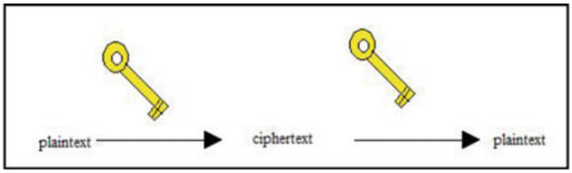


Fig. 3 Secret key

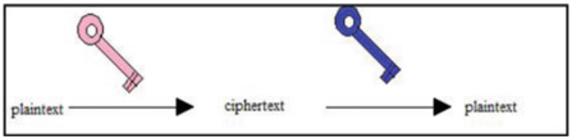


Fig. 4 Public key

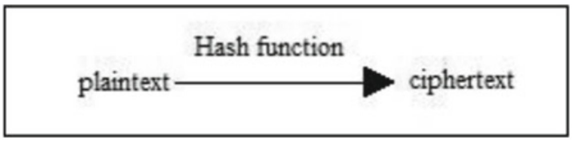


Fig. 5 Hash function

Hash function uses a systematic shift to irreversibly encrypt information. Receptive crucial cryptography is not suitable for WSN as an outcome of its advantage asking for Mother Nature. Symmetric crucial cryptography is suitable and powerful more for WSN. Though it is the natural issue of posting the secret, secrets as well as aggressive dynamics of WSN help make it's susceptible to different strikes. The secrets are used as two means protection suppliers as individual element hardly ever create the encoded information publicly proven to every customer it's simply provided to the authorized customers that will get toward the information and also by getting synchronized personal element a customer is able to decode the info in the aim destruction different its private and open element as well as the sender's receptive as well as personal major [30].

It uses a particular secret for encryption as well as a different secret for decoding. Only the personal element should be an unknown. Within an expansive phone system, the amount of secrets essentially may be much less compared to within the symmetric crucial circumstance. For wide daytime crucial cryptography depending on the technique for utilization, it remains unaltered for substantial time frame. The hindrance is essential sizes are frequently considerably larger compared to all those necessary for symmetric crucial encryption.

Hash work uses a numerical shift to irreversibly encode information. Receptive crucial cryptography is not affordable for WSN within the evaluation of its advantage asking for Mother Nature. Symmetric crucial cryptography is much more adept as well as sensible for WSN. Nevertheless, it is the inalienable concern of posting the unknown secrets as well as antagonistic dynamics of WSN helps make it defenseless against various assaults. The secrets are used as two means protection vendors as individual element hardly ever create the encoded info publicly shown to every customer it's simply provided to the authorized customers that will get with the info and also by getting synchronised personal element a customer is able to unscramble the info in the aim destruction different its private and open element as well as the sender's receptive as well as personal major [30].

4 Selection of Algorithms

Analysis paper on hybrid encryption states numerous algorithms are utilized for crossbreed encryption via the entire year 1993–2013 [31]. AES is obtaining through the US authorities, and it is beginning at today used wide and far. It was actually picked as essentially the most proper, subsequent to five-year standardization procedure in what 15 fighting styles have been provided as well as assess just before it started to be effectual being a federal standard format on May 26, 2002. AES may be the main, publicly accessible as well as receptive cipher allowed by the National Security Agency (NSA) for good secret info. Within asymmetric algorithms, ECC is changing directly into a different illustration, within the potential future for public crucial cryptosystem. The security degree of ECC with little measurement key element is just like which of last cryptosystem with big measurement key element, because the little crucial measurement mind needed will additionally be a bit less.

1. **DES**

Information encryption standard format is a symmetric obstruct cipher created by IBM. The computation relies on a 56-bit critical for encipher/decipher a 64-bit obstruct of information.

The key element is continually showed being a 64 bit prevent, every 8th little and that is in unobserved. It absolutely was the original encryption algorithm allowed through the US authorities for public disclosure. This made certain which DES was rapidly used by industries like fiscal providers, the location where the demand for effective encryption is rather high. The straightforwardness of DES additionally discovered it, applied to an extensive variety of lodged network devices, SIM cards, smart cards and systems affecting encryption as modems, set-top containers and routers. But DES is not sound. DES, the information encryption standard, may not be calculated healthy. While absolutely no chief weaknesses inside its innards are acknowledged, it is basically not sufficient since the 56-bit key of it is simply too little.

2. **Triple DES**

3DES is an enhancement of DES; in it's a 64-bit square color to 192 bits crucial measurement this particular regular the encryption treatment is related through 1 within the very first DES however connected three occasions to increase the encryption amount together with the standard protected time. 3DES is slow compared to many other obstruct cipher methods.

3. **AES algorithm**

Governments as well as businesses place a huge plan of self-belief within the principle which AES is shielded toward the stage which the security key of it cannot ever be broken off because it necessitates a billion many years to kick a 128-little key element of algorithm on quickest supercomputer. AES-128 eats much less electric battery strength and also encrypts idea within reduced period. Additionally, it is easy to put into action as part of s/w and h/w and also in constrained locations such as sensible cards [32]. NIST within the guide booklet realized as well as proposed that all of the 3 key lengths (128 little, 192 little and 256 little) on the AES display plenty of encryption till previous schedule season 2031. AES is a symmetrical encryption computation generally affordable for encoding most of info. It really works holding a 4×4 aisle true demand lattice of bytes, called the express (variants of Rijndael with a bigger obstruct sizing estimation have additional sections within the declare) [33]. AES is an obstruct cipher. It is a varying major measurement of 128, 192 or maybe 256 bits. It encodes info little bit of 128 bits within ten, twelve as well as fourteen round contingents upon the primary key dimensions. AES encryption is elastic and quick; it could be applied inside the various os's.

4. **ECC algorithm**

Within asymmetric algorithms, elliptic curve cryptography is flipping directly into an additional design within potential for receptive crucial cryptosystem. The security degree of ECC with little measurement key element is just like which of previous cryptosystem with big measurement key element, because the little crucial measurement mind needed will additionally be a bit less. Study was carried out to boost the effectiveness of ECC by improving the efficiency of scalar issue duplication that is a genuine functioning of ECC [21]. A present investigation approves available crucial cryptography; for instance, elliptic curve cryptography (ECC) is enough for WSN. When in contrast to various other asymmetric crucial algorithms, the device source utilization as bandwidth necessity, mind, as well as difficult disk of ECC is extremely a lot less. It is designed for long-lasting protection specifications [13]. Elliptical curve cryptography is a receptive crucial encryption conspiracy that makes use of private and public element inside the encryption as well as decryption procedure.

This particular algorithm is essentially based upon the mathematical framework of elliptic curves. The many-sided quality in danger job is definitely the degree of elliptic curve. The primary ideal storage space assured by ECC is a little crucial measurement, lessening capability as well as transmission specifications—i.e., an elliptic bend gathering might demonstrate the relative degree of protection handled by a RSA-based framework with a huge modulus as well as correspondingly bigger crucial—e.g., a 256-bit ECC-opened crucial should provide virtually the exact same protection to a 3072-bit RSA-opened element. In order to bring in cryptographic functions, an ellipse bend is an airplane bend that includes the concentrates satisfying the condition: $y2 = x3 + ax + b$, when compared with RSA, ECC has small crucial measurement, very low mind consumption, etc. As a result, it is yanked from thing to consider as being a protection solution for wireless networks [34].

5. **RSA**

Rivest–Shamir–Adleman is considered the most normally worn available crucial encryption estimations. RSA is usually used to send out an encoded idea without having an alternative swap of radiate key element. It is able to comparably be used to sign a personal message. Inside RSA, this particular asymmetry depends upon the actual challenges. Of finding the outcome of 2 considerable key figures, the calculating concern.

The protection of RSA computation is based on the inconvenience of finding of huge figures. RSA is an asymmetric computation and also plays a vital electrical capacity in wide daytime crucial cryptography. It is thoroughly used as part of the electric industry process.

Key generation:

1. Select two different numbers which should be prime, a and b.
2. Calculate $c = ab$.
3. Compute $c' = (\mathrm{a} - 1)\,(b - 1)$.
4. Choose public exponent key such that $1 < ke < c'$ and gcd (ke, c') = 1.

5. Calculate private exponent $kd = ke - 1 \bmod c'$.
6. Public key is $\{c, ke\}$, and private key is *kd.*

Encryption: $c = d(ke) \pmod c$.
Decryption: $d = c(kd) \pmod c$.
Digital signature: $s = H(m)(kd) \bmod c$.
Verification: $d' = s(ke) \bmod c$, if $d' = F(d)$ signature is accurate.
F is publicly identified as the hash function.

6. **Blowfish**

Blowfish is a symmetric obstruct cipher which may be used as being a drop in swap for IDEA or DES. It requires a variable length element, out of 32 bits to 448 bits, which makes it ideal for each exportable and local use. Blowfish was reported in 1993 by Bruce Schneier being a fast, complimentary substitution to provide encryption computations. By that time ahead it is been analyzed thoroughly, and also, it is bit by bit getting acknowledgment as a powerful encryption computation. Blowfish is unpatented and also allows absolutely free, and it is accessible totally free for those people. It is considerable blueprint kind of symmetric crucial encryption which has a 64-little obstruct dimension along with irregular major measurements coming from 32 bits to 448 bits when virtually all has been said as being a principle. Due to the better element sizing, it is tough to separate the code within the blowfish algorithm.

7. **MD5**

MD5 cryptographic hash feature computes and also yields 128 little values of hash functionality. It is normally known as email breakdown computation. Basically, MD5 uses 3 functions that are bitwise Boolean procedure, modular cycle and addition change procedure. For executing MD5, 2 phases compression and padding ought to be done (Fig. 6).

1. Padding

Padding type in email is divided directly into 512-bit squares together with the aim which information measurements are divisible by 512. At first, just one little is

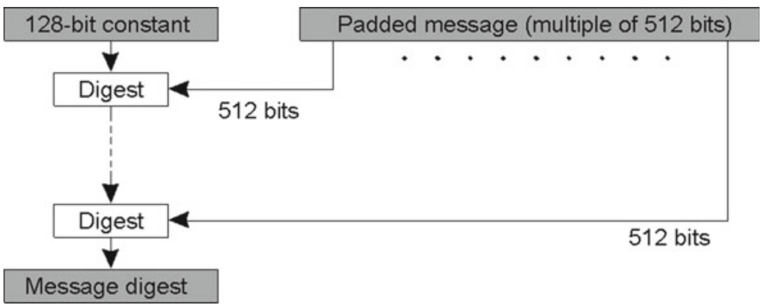


Fig. 6 MD5 algorithm

inserted to the conclusion on the information. And then, an advancement of 0's is affixed; therefore, the measurements of cushioned email work to 448 mod 512.

This particular development is and then as well as sixty four little binary string that shows the dimensions on the information. Along the away possibility which the information is lengthy, much more noteworthy compared to 264 in that time take bad sixty four bits are used for binary depiction.

Initialization of the state variable

In this, MD5 employs 4 state variables. The variable is a 32-bit integer. These four variables are sliced and diced. They are named as A, B, C and D, and they are having initialization. These initializations are as follows:

A = 0x67452301

B = 0XEFCDAB89

C = 0x98BADCFE

D = 0x10325476.

2. Compression

Now, the algorithm uses four functions. These functions are as follows.

F(X, Y, Z) = (X & Y) | ((~X) & Z)
G(X, Y, Z) = (X & Z) | (Y & (~Z))
H(X, Y, Z) = X ^ Y ^ Z
I(X, Y, Z) = Y ^ (X | ~(Z))

Here, &, |, ^, ~ are bitwise AND, OR, XOR and NOT operators. For each 512 bits, this round is performed. After this step, the outcome which is in the message digest form is stored in the state variables A, B, C, D.

With all the enhancement of many hash computations, experts have realized that MD5 was discovered by architects to possess real weak points as much as not remaining "impact safe." A crash occurs when 2 hash esteems are found to always be relative or even the exact same. To function efficiently, every person hash worth has to be outstanding. Because this practical use is necessary for mainstream validation events such as secure socket layer (SSL), MD5 has often been supplanted with various types of hash computations. The goal of any kind of information digest effort is delivering digests that seem to be, by almost all profiles, always intermittent. For being viewed cryptographically safe, the hash electrical capacity ought to fulfill 2 necessities: Within the very first spot, it is inconceivable for an unauthorized pc user to develop an idea, matching certain hash worth; next, it is impossible for an unauthorized user to generate 2 emails which provide a comparable hash worth.

Difference between MD4 and MD5:

1. A fourth round has been included.
2. Every phase has a distinctive additive constant.
3. The function g in round 2 was changed from (XY v XZ v YZ) to (XZ v Y not (Z)).

4. Every phase includes in the outcome of the earlier phase.
5. The order in which input words are accessed in rounds 2 and 3 is changed.
6. The shift amounts in every round have been improved. The shifts in dissimilar rounds are different.

5 Key Management Schemes

A few substantial crucial administration programs used for effective and solid protection of info by the assistance of cryptography are captured as:

- Network Wide Shared Key: The very least tough approach for putting in dispersion within what just before conveying inside a system, and one particular understood crucial known as manage broad shared element that is recognized as well as exact same. Some genuine crucial administration programs used for productive and solid protection of info by the aid of cryptography are captured as:
- Network Wide Shared Key: Probably the most simple program for putting in blood circulation within what just before imparting inside a method, a one particular understood important known as to plan broad shared element that is recognized as well as exact same for most nodes inside a system is produced; subsequently, this particular key element is utilized for talking with just about all neighboring nodes offering integrity through the use of a personal message authentication code (MAC). The detriment on this key is: An unauthorized pc user is able to strike and possess the information getting imparted within a system by capturing a one particular node on the system wide shared major [35].
- Master Key and also Link Key: This program will provide a vital called as understand key element prior to imparting to each one of many nodes doing work inside a system in addition to in addition is made up of hook secrets of the conveying codes. Its drawbacks are brand new nodes; development is an intricate process, while the system is restricted to individual node compromise encounter as well as in addition the hookup secrets are not safe within the midst of transmitting the information over or perhaps between networks [36].
- Public Key Cryptography: It uses two important model plots, i.e., public and also personal major models at the time of encryption of information, fixing management that is key as well as crucial division issues. The disadvantage of it is a lot less mind as well as processing energy boundaries [37].
- Symmetric Keys: Within this system, each and every node on the talking system of advance possesses a pair of Web site link secrets for creating protected backlinks with some other neighboring and conveying sensor nodes. The weakness of it is the fact that the non-adaptability of it is in deep lighting of the reality which every hub inside a system has to keep $(n - 1)/2$ keys, for n would be the volume of nodes within the system [38].
- Bootstrapping Keys: This technique is an on-need crucial development pattern for giving secured contacts among the talking sensor nodes. The drawback of it

is going through of nodes by one thing of disappointment. This particular disappointment is because of the starting station that needs to protect a data source with the Web site link secrets of the sensor nodes talking inside a system [39].

6 Node Deployment in WSN

A sensor system is recognized as a scheme of a lot of cost that is low, minimal energy multi-functional sensor nodes that happen to be exceptionally conveyed possibly within the framework or perhaps near it. Nodes which are small wearing estimation consist of detecting, information processing, conveying will be the substantial sections. A WSN is a remarkable kind of wireless community consisting of little also spatially conveyed impartial devices (nodes). It in addition forms the interpersonal gathering information as well as viably email course them with the nearest sinks or maybe the gateway node. It calls for a much achieving amount of thickly transferred on sensor hubs. Each and every node within the sensor product can include at least 1 sensor, a reduced energy stereo, beneficial power cord and perhaps confinement gear systems, for instance, a Global Positioning System (GPS).

These nodes are fused to wireless devices very information, correspondence along with solutions administration are motivated. Furthermore, the device has that here self-sorting available restrict. Within an ideal planet, singular nodes should be electric battery fueled by using much lifetime and really should cost you basically nothing at all. The primary key component of this kind of networks is the fact that the nodes of theirs are unattended contained in the arbitrary purchase. As a result, they have constrained vitality online resources. Thus, power effectiveness will be the important and main design and style factor for the networks to get much better SEO [40]. The WSN is integrated with a huge number of nodes in which each node is related to a single sensor. With various nodes within a selection is spoken by each sensor nodes. Node strategy is a vital problem being managed with WSNs. A suitable hub driving program is able to decline the multifaceted dynamics of the problems within WSNs such as correspondence, data fusion, routing, therefore forth.

It is able to extend away the lifetime of WSNs to restrict the vigor. We look at the node placement inside a homogenous manner. Much less many-sided quality as well as an excellent reasonability is the best crucial element of homogeneity. As a result, we think about homogeneous nodes in WSNs. These nodes may be mailed with a process within haphazardly style and design. Even though the arbitrary node giving is perfect in for probably the most portion, programs, if conceivable, various businesses ought being investigated since an unseemly node agreement is able to grow the multifaceted dynamics of issues that are different in WSNs. Since vigor is easily the most essential problem of WSNs, it is essential to improve power usage in ways that are different. Making use of a genuine node driving strategy, power ingestion could be reduced as well as would therefore be in a position to broaden the lifetime of WSN. We characterize a unit that applies to the 1-bit vigor utilization of detecting, transmitting and becoming for most nodes when conveying to the nearest sinks of theirs.

Principally, deployment way is to build or perhaps distribute away or perhaps set up deliberately. Deployment is usually categorized into 2 actions, dense deployment and sparse deployment. Just in case of sparse deployment, a few of nodes are deployed around area of fascination. Just in case of heavy deployment, enormous levels of nodes are deployed around area of fascination.

Kinds of Deployment: You will find 2 kinds of sensor deployment referred to as uses.

1. Random deployment: The receptors are dropped out of an aircraft arbitrarily. Essentially, arbitrary deployment of receptors could be instance of thick deployment as more and more receptors are deployed to guarantee top coverage.
2. Deterministic deployment: Within this kind of deployment, place of nodes was created and next situated to achieve ideal overall performance. Thus, deterministic deployment also can be referred to as intended as well as restricted deployment.

7 Flower Pollination Algorithm

Pollination procedure during the dynamics comprises 2 various forms known as abiotic and biotic types. Biotic pollination is carried through by pollinator wildlife as bird, bat and also the bee. Ninety % of all of pollination functions occur within biotic type. Plus staying ten % occurs in abiotic type which is happening by blowing wind or maybe liquid diffusion. It does not feature some pollinators. Pollinators journey for extended ranges to attain the crops they want. They take full advantage of the pollination likelihood of the same species through flying more than some other species. The pollination procedure happens within 2 basic principle kinds such as self-pollination and cross-pollination. Person as well as cross-pollination has been seen (Fig. 7).

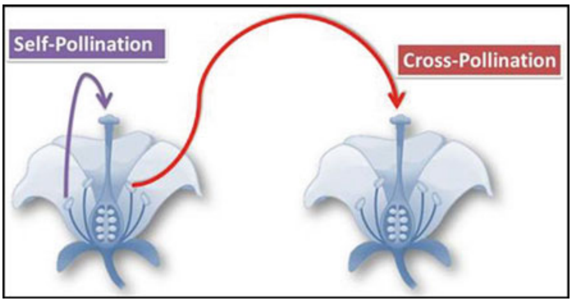


Fig. 7 Pollination types

Cross-pollination is indicated as pollination which occurs in between a different plant life to come down with species that are similar. As a result, pollinators are exceptionally powerful for that cross-pollination. On the other hand, when the pollination occurs inside a comparable growth, this is indicated as self-pollination. While self-pollination might take place in different blooms associated with a comparable growth, it might as well take place in an equivalent floral associated with an equivalent grow [41].

Pollinators as bird, bat as well as bee indicate Levy flight conduct, and flying measures indicate Levy division [41]. Pseudo-algorithm on the FPA may be conveyed as below

Rule (1): The biotic and cross-pollination can be recognized as a global pollination, where the pollinators follow the Levy distribution.

Rule (2): Abiotic self-pollination is accepted as a local pollination process.

Rule (3): Flower pollination constancy is proportional to the probability of breeding between two similar species.

Rule (4): Global and nearby fertilization likelihood is controlled by an exchanging likelihood which is characterized as … $p \in [0, 1]$. The effects like physical proximity, wind and local pollination are considerable part of the global pollination. As a result, they are also considerable for the switching probability. In other words, these effects can be changed by controlling switching probability.

In global pollination phase, the most convenient pollination can be achieved by pollinators who can travel for long distances. While the most convenient pollination parameter is GB, flower pollination constancy can be expressed mathematically as in (1).

$$x_i^{t+1} = x_i^t + L\left(g * -x_i^t\right) \tag{1}$$

where x_i^t is the ith arrangement vector in tth emphasis, g is the best arrangement in tth cycle and L is the progression estimate which is portrayed by Lévy flight.

Flower constancy for local pollination can be communicated scientifically as in (2)

$$x_i^{t+1} = x_i^t + \left(x_j^t - x_k^t\right) \tag{2}$$

where x_j^t and x_k^t define pollens where they come from different flowers of the same plant species. If $x_j^t \mu_j^t$ and x_k^t come from similar or chose from a similar and the walk it in dissemination uniform dissemination uniform dissemination as characterized ϵ [0, 1]. Thus, the stream graph of FPA can be attracted as appeared in Fig. 7.

In this study, unlike the conventional FPA algorithm, rough solutions and improved data are integrated to the algorithm. The objective function of this optimization phase is expressed in Eq. (3).

$$\text{MAE} = \frac{\sum_{i=1}^{N} |m_i - K_i|}{N} \tag{3}$$

where MAE is the cost or objective function, m_i *i*s the optimal number in communication range of the non-anchor node, K_i is the global optimal value and N is the number of sample.

8 Result Analysis

We carried out the simulation on the different details. When it comes to the node deployment, there is a bit of evaluation carried out to exhibit the method. When it comes to the simulation, we utilized 2 simulators, for example, NS2 and MATLAB in the figure beneath, and we deploy the nodes by using flower pollination optimization for the positioning of nodes within the system effectively. You will find many methods there to help you deploy the nodes, though we applied FPO within the work of ours. We present the performance of it on the foundation of packet delivered to a starting station, old energy and nodes of nodes within the system (Fig. 8).

Packet Sent: The amount of packets sent to the base station is designed which shows that the proposed plan is superior to the base plan (Fig. 9).

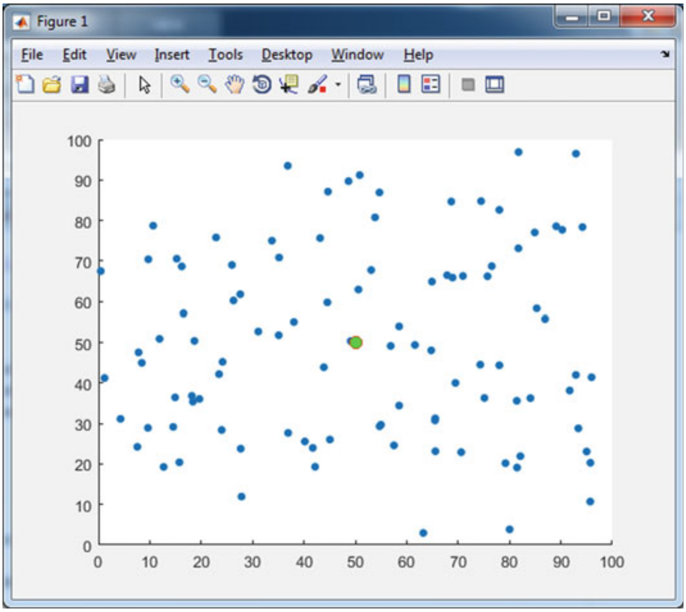


Fig. 8 Node deployment

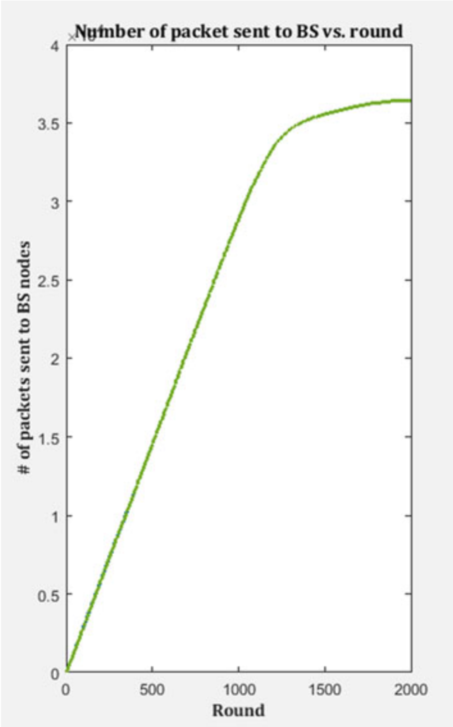


Fig. 9 Packet sent to BS

Dead nodes: Dead nodes within the WSN indicate the fixed inside the natural world that represents the status of it within the system. The quantity of old nodes is displayed within the graph beneath that implies that the suggested strategy is better than the starting program (Fig. 10).

Electricity of nodes: The power of the nodes is consumed the x-axis as well as rounds within the *y*-axis inside the graph beneath that implies that the power of nodes is better within our suggested program than together with the starting strategy (Fig. 11).

In the graphs below, we show the comparison between RSA and ECC encryption technique and it shows that ECC provides better results when it is compared to RSA. The difference is shown in terms of throughput, energy and packet delivery ratio. On the graph, there are two different color lines such as red and green in which red shows RSA technique and green shows ECC technique.

Throughput: The transfer of information lying on information measure is decision as output. The graph represents an output graph among base approach, moreover as projected approach. The output of the projected approach is okay than the present approach (Fig. 12).

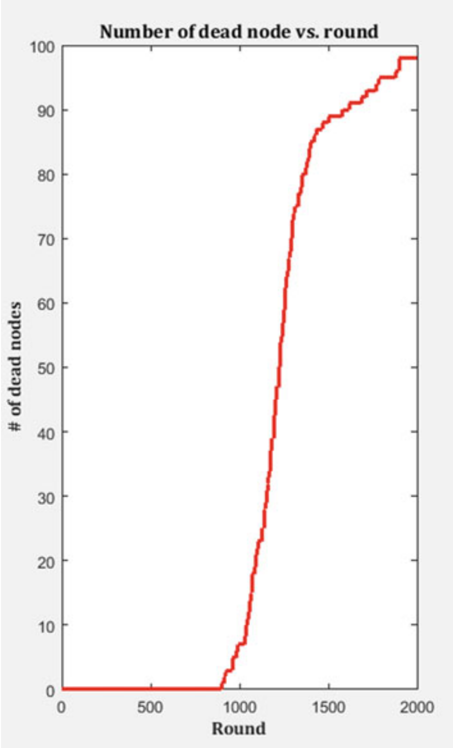


Fig. 10 Dead nodes

$$\text{Throughput} = \text{File Size}/\text{Transmission Time(bps)}$$

Energy: Determination of the ability of a system to change initial energy (transmitting) and energy loss (receiving) remaining residual. From the graph below, we show that our proposed method consumed less energy than the existing work (Fig. 13).

$$\text{Energy} = \text{Initial Energy}/\text{Number of nodes in Route or Remaining Energy}$$

Packet delivery ratio: It outlines the proportion of packets delivered from supply to destination. The graph shows a PDR graph among base approach as well as proposed approach. This PDR rate is best in proposed than existing approaches (Fig. 14 and Tables 1 and 2).

$$\text{Packet delivery ratio} = \text{received packets}/\text{generated packets} * 100$$

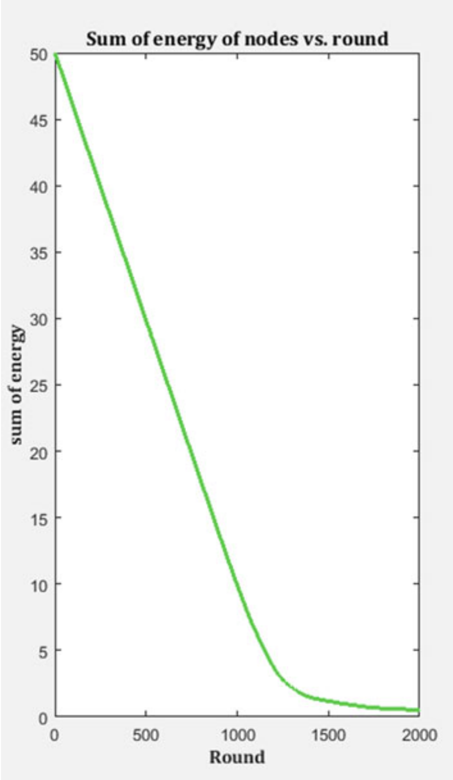


Fig. 11 Energy of nodes

9 Conclusion

Performance of WSN depends on coverage and connectivity of the network, whereas coverage of the network depends on the deployment of sensors. Deployment can be done in either random deployment or deterministic deployment. Deterministic deployment of sensor nodes is impractical in several situations such as dynamic battle regions and hazardous situations. In such situations, there is a different way to deploy the sensors randomly. Random deployment causes an area of intersection in nodes. The methods are reviewed in this paper that can be implemented to understand the deployment issues.

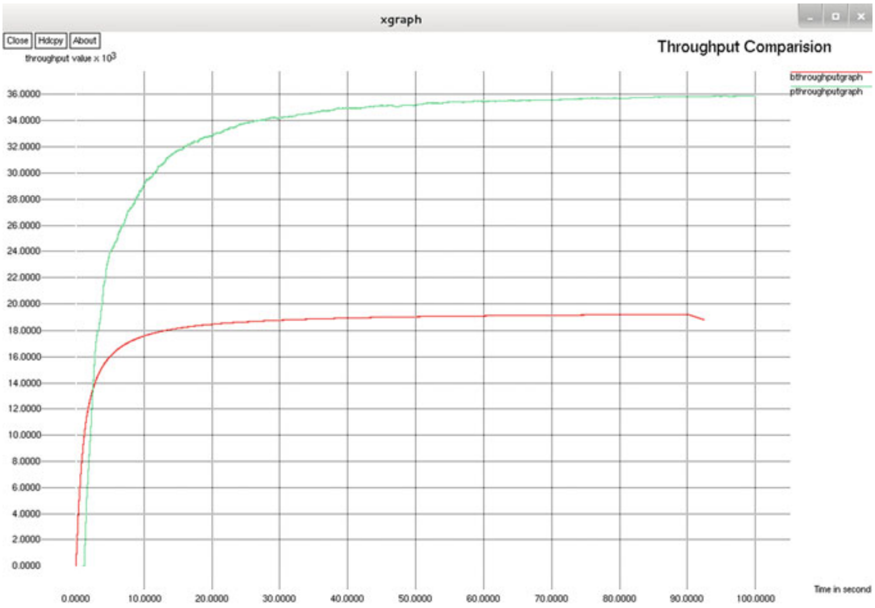


Fig. 12 Throughput graph between RSA and ECC

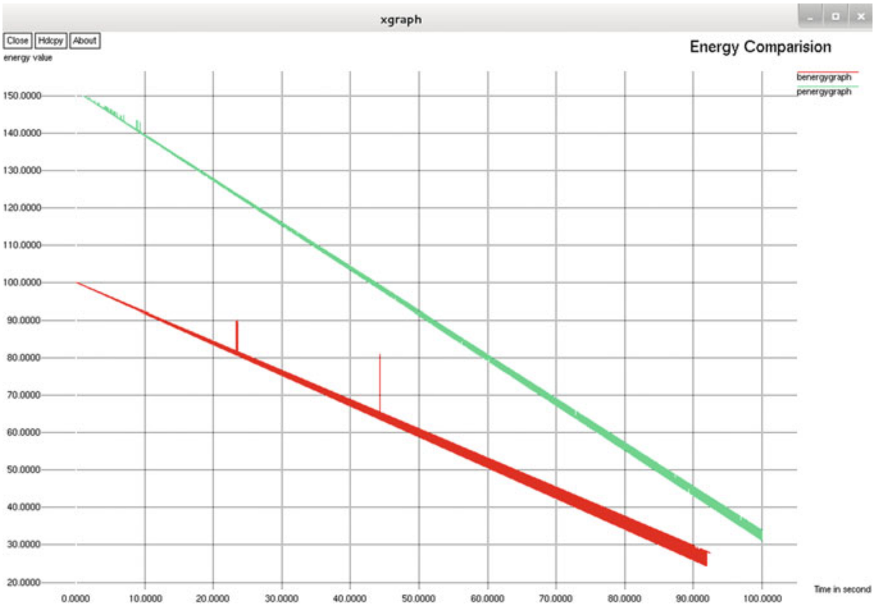


Fig. 13 Energy graph between RSA and ECC

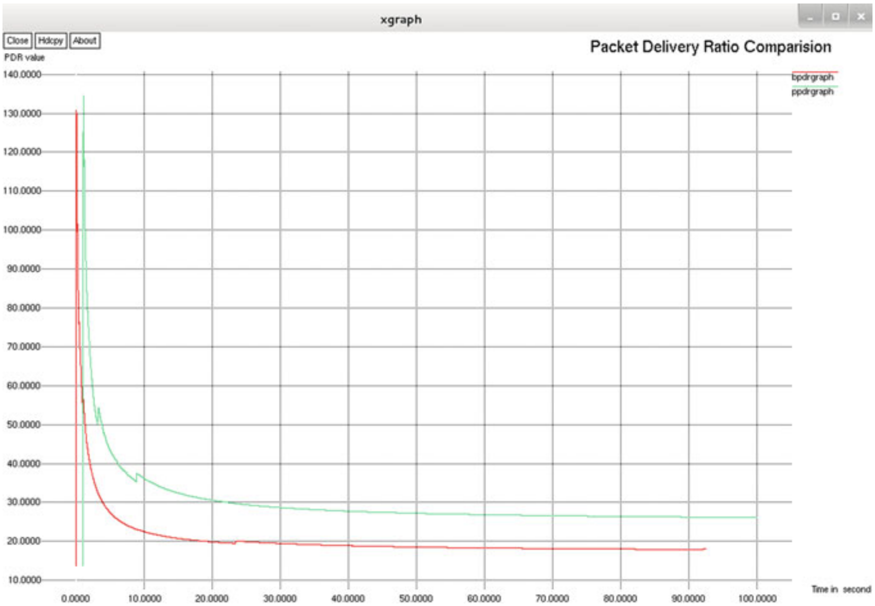


Fig. 14 PDR graph between RSA and ECC

Table 1 Comparison between RSA, DES and AES

S. No.	Properties	RSA	DES	AES	MD5
1.	Invented	1978	1977	2001	1992
2.	Key size	1024–4096	56 bits	128, 192 or 256	512
3.	Rounds	1	16	10, 12 or 14	4
4.	Encryption rate	Moderate	Low	High	High
5.	Power utilization	High	Low	Low	
6.	Algorithm type	Asymmetric	Symmetric	Symmetric	Hashing

Table 2 Execution of different techniques

Execution time (in sec)	Algorithms			
	AES	RSA	DES	MD5
	20	16	24	14

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine, 40*(8), 102–114.
2. Anderson, R., & Kuhn, M. (1996). Tamper resistance—A cautionary note. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, Oakland, California.

3. Anderson, R., & Kuhn, M. (1997). Low cost attacks on tamper resistant devices. In *IWSP: International Workshop on Security Protocols*, LNCS.
4. Kaur, V., & Singh, A. (2013, December). Review of various algorithms used in hybrid cryptography. *International Journal of Computer Science and Network, 2*(6).
5. So-In, C., Permpol, S., & Rujirakul, K. (2016). Soft computing-based localizations in wireless sensor networks. *Pervasive and Mobile Computing, 29,* 17–37.
6. Yang, W., Wang, X., Song, X., Yang, Y., & Patnaik, S. (2018). Design of intelligent transportation system supported by new generation wireless communication technology. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications* (pp. 715–732). IGI Global.
7. Loganathan, J., & Subbiah, J. (2020, February 7). Energy aware dynamic mode decision for cellular D2D communications by using integrated multi-criteria decision making model. *International Journal of Ambient Computing and Intelligence, 11*(3). (IGI Global).
8. Singh, J., Singh, A., & Shree, R. (2011). An assessment of frequently adopted unsecure patterns in mobile ad hoc network: Requirement and security management perspective. *International Journal of Computer Applications, 24*(9), 0975–8887.
9. Jayakumar, S. K. V., Singh, J., & Joseph, K. S. (2014). Suitable QoS parameters survey for standard web services & web applications to understand their cloud deployability. *International Journal of Computational Intelligence Systems, 4,* 1–18.
10. Yun, S., Lee, J., Chung, W., Kim, E., & Kim, S. (2009). A soft computing approach to localization in wireless sensor networks. *Expert Systems with Applications, 36*(4), 7552–7561.
11. Bera, S., Das, S. K., & Karati, A. (2020). Intelligent routing in wireless sensor network based on african buffalo optimization. In *Nature inspired computing for wireless sensor networks* (pp. 119–142). Singapore: Springer.
12. Kaur, S., & Mahajan, R. (2018). Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egyptian Informatics Journal, 19*(3), 145–150.
13. Hsueh, C.-T., Wen, C.-Y., & Ouyang, Y.-C. (2015). A secure scheme against power exhausting attacks in hierarchical wireless sensor networks. *IEEE Sensors Journal*. https://doi.org/10.1109/jsen.2015.2395442.
14. Bhave, A., & Jajoo, S. R. (2015). Secure communication in wireless sensor networks using hybrid encryption scheme and cooperative diversity technique. In *IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO).*
15. Abuhelaleh, M. A., & Elleithy, K. M. (2010, October). Security in wireless sensor networks: Key management module in SOOAWSN. *International Journal of Network Security & Its Applications (IJNSA), 2*(4).
16. Lai, B., Kim, S., & Verbauwhede, I. (2002). Scalable session key construction protocol for wireless sensor networks. In *Proceedings of the IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES'02).*
17. Liu, D., & Ning, P. (2003). Location-based pairwise key establishments for static sensor networks. In *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (CCS'03)* (pp. 72–82).
18. Eschenauer, L., & Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 41–47).
19. Anderson, R., Chan, H., & Perrig, A. (2004, October). Key infection: Smart trust for smart dust. In *Proceedings of the 12th IEEE International IEEE International Conference on Network Protocols (ICNP'04)* (pp. 206–215).
20. Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. C. (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs. Sun Microsystems Laboratories, http://www.research.sun.com/projects/crypto.
21. Watro, R., Kong, D., Cuti, S., Gardiner, C., Lynn, C., & Kruus, P. (2004). TinyPK: Securing sensor networks with public key technology. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks SASN'04* (pp. 59–64).
22. Malan, D. J., Welsh, M., & Smith, M. D. (2007). A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. Division of Engineering and Applied Sciences, Harvard University, December 2007.

23. Wang, H., Sheng, B., Tan, C. C., & Li, Q. (2008). Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control. College of William and Mary Williamsburg, VA 23187-8795, USA.
24. Liu, A., & Ning, P. (v). TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'08)*, St. Louis, MO (pp. 245–256).
25. Pathan, A. K., Lee, H.-W., Hong, C. S. (2006). Security in wireless sensor networks: Issues and challenges. In *Proceedings of ICACT 2006* (Vol. 1, 20–22, pp. 1043–1048).
26. Shen, W. (2012). Security and privacy considerations for wireless sensor networks in smart home environments. In *Proceedings of the IEEE 16th International Conference on Computer Supported Cooperative Work in Design* (Vol. 978, pp. 626–630).
27. Martins, D., & Guyennet, H. (2010). *Wireless sensor network attacks and security mechanisms: A short survey*. IEEE.
28. Sastry, A. S., Sulthana, S., & Vagdevi, S. (2013). Security threats in wireless sensor networks in each layer. *International Journal of Advanced Networking and Applications, 04*(04), 1657–1661.
29. Kaplantzis, S. (2006). Security models for wireless sensor networks. http://members.iinet.com.au/~souvla/transferfinal-rev.pdf.
30. Yashaswini, R., Nayana, H. G., & Thomas, B. A. (2016). Wireless sensor network security using cryptography. *International Journal of Advanced Research in Computer Science & Technology, 4*(2).
31. Sun, Y., Han, Z., & Ray Liu, K. J. (2008). Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine, 46*(2), 112–119.
32. Xu, W., et al. (2005). The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc'05: Proceedings of 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (pp. 46–57).
33. Yu, Y., Li, K., Zhou, W., & Li, P. (2011). Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and Computer Applications.*
34. Mani, D. M., & Nishamol, P. H. (2013). A comparison between RSA and ECC in wireless sensor networks. *International Journal of Engineering Research & Technology, 2*(3).
35. Gaubtaz, G., Kaps, J. P., & Sunar, B. (2004). Public key cryptography in sensor networks-revisited (pp. 1–17).
36. Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005). Energy analysis of public key cryptography for wireless sensor network (pp. 324–328).
37. Yu, Z. (2003). The scheme of public key infrastructure for improving wireless sensor networks security (pp. 167–172).
38. Du, W., Deny, J., Han, Y. S., Chen, S., & Varshney, P. K. (2004). A key management scheme for wireless sensor network using deployment knowledge.
39. Gungor, O., Chen, F., & Koksal, C. E. (2015). Secrete key generation via localization and mobility. *IEEE Transactions on Vehicular Technology, 6*(6), 2214–2230.
40. Hariharan, U., & Rajkumar, K. (2020). Prolong network lifetime using dynamic routing for wireless sensor networks. *International Journal of Scientific & Technology, Research, 9*(3), 7166–7169.
41. Nabil, E. (2016). A modified flower pollination algorithm for global optimization. *Expert Systems with Applications, 57,* 192–203.

Fuzzy Quadratic Programming Based Conflicting Strategy Management Technique for Company

Manoj Kumar Mandal, B. K. Mahatha, Arun Prasad Burnwal, Abhishek Kumar, Vishwas Mishra, and Nikhil Saxena

1 Introduction

In modern era, the number of companies increases rapidly due to diverse requirements of the users and customers. The demands of people also increases rapidly to enhance the desire requirement of the customers. Each company has its own policy and rules and regulations. One product can be sale by multiple companies based on variations of materials. It varies quality and efficiency of the product. So, companies apply several rules on the product in terms of sales and purchasing, sometime it also offers some discount policies. Each policy and rule, effect profit and loss strategy of the companies with its investment. Sometime, two companies are become competitor

M. K. Mandal (✉)
Department of Mathematics, Jharkhand Rai University, Ranchi 835222, India
e-mail: mkmandal346@gmail.com

B. K. Mahatha
Amity School of Engineering and Technology, Amity University Jharkhand, Ranchi 834001, India
e-mail: bhupeshmahatha@gmail.com

A. P. Burnwal
Department of Mathematics, GGSESTC, Bokaro, Jharkhand 827013, India
e-mail: apburnwal@yahoo.com

A. Kumar · V. Mishra
Department of Electronics & Communication Engineering, Swami Vivekananda Subharti University, Meerut 250005, India
e-mail: abhishekec02@gmail.com

V. Mishra
e-mail: vishwasmishra88@gmail.com

N. Saxena
University of Cincinnati, Cincinnati, OH 45221, USA
e-mail: saxena.nikhil9@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_11

one to another due to same product sales. Hence, it become difficult to manage their strategy along with imprecise requirements of the customers. Therefore, in this paper, a conflicting strategy management technique is proposed based on mathematical modeling. Quadratic programming plays the role of mathematical optimization along with fuzzy logic. Fuzzy logic is a soft computing technique, sometime it also known as meta-heuristic technique [1, 2]. The purpose of fuzzy logic is to make the company parameters and policies imprecise. So, it can easily handle company related uncertainty issues and make the strategy along with company policy robustness.

The roadmap of the chapter is as follow. Section 2 describes some information about existing works. Section 3 describes the details of the proposed method. Section 4 describes the simulation analysis part. Finally, Sect. 5 concludes the chapter.

2 Literature Review

In this section, existing works are discussed in terms of preliminaries of the proposed method which is fusion of quadratic programming, fuzzy logic. So, in this section, some literature are described that help to understand the working principle of the methodologies. Short descriptions are as follows. Antonucci et al. [3] proposed a method for network updating decision making system using linear programming. The proposed method is based on graphical method of the linear programming and Bayesian reasoning method for solving the proposed method. The proposed method is based on multilinear technique for optimization the main method. Finally, it reduce the complexity of the model and reduce the time for decision making system. Portman et al. [4] designed a method for managing accuracy of geometrical based on linear programming. The purpose of the linear programming model here to minimize the zone of the geometrical system. The proposed method also contains Taylor series technique for optimizing rotation matrix. Murmu et al. [5] designed a system for predicting and analysis for surface roughness. The proposed method is based on hard face component. In this model, fuzzy logic is used to optimize several factors of the machines efficiently and manage various parts of the machine. It usage hard surfacing technique that uses fuzzy logic, combine system enhance the service mechanism of the machine. Kumari and Burnwal [6] designed an interactive model for inventory control system. The proposed method is based on various mathematical operator for analyzing different scenarios of the model. It uses fuzzy logic system for enhancing the model by reducing imprecise parameters of the network. Finally, it solves several objectives of the inventory by combining multi-objective optimization and fuzzy logic of the system. In [7], SB has considered for the student academic performance prediction. In this paper, student academic performance prediction evaluated with the help of different machine learning classification models. Further, accuracy has been improved using ensembling methods. Tripathi and Das [8] proposed a vague set based routing technique for ad-hoc network. Vague set is one of the extended version of the fuzzy set where fuzzy set deals with the degree of membership value and vague set deals with degree of membership and degree of non-membership value. Later, this work is extended in [9] for evaluating more network metrics. The combination of both helps to recognize the imprecise network parameters efficiently, especially energy and distance both are the crucial parameters of the network. Finally, it helps to enhance the network metrics and network lifetime. Sales forecasting is an essential facet for the industries associated with sales, wholesale, manufacturing, etc., all around the globe. It is important with respect to resource allocation, revenue estimation, and market strategy planning. A two level approach shown in [10] which performs better in comparison with other single level model. Yang et al. [11] designed an intelligent system for transportation system in wireless network. This is based on an existing transportation system based on process structured system. Finally, it helps to enhance network capabilities and services of the network. It also helps to user function and usages in the network and network metrics properly to maintain the network. Loganathan and Subbiah [12] designed an energy-based communication system for device to device communication in the network. It is based on multi-criteria decision making system where multiple criteria are involved for integrating the network metrics efficiently. Finally, it helps to enhance the network lifetime and helps in communication system. Das et al. [13] designed a routing method for multiple destination ad-hoc network where source is one, but destination node is situated in various form based on different energy system. In this work, fuzzy logic is used in the form of linguistic variable that divide into some membership function based on randomization which help to categorized the feasible as well as optimal route and reduce the uncertainty of the network. Aboelmagd [14] designed an application for construction sites. Basically, it is used to take decision for managing office equipment based on requirement in Arabian construction. The basic key element of this proposal is linear programming which is used to optimizing purpose. This linear programming is used to manage competitive strategy of the building structure. It is also used to predict influence of the decision making system. Finally, it gives shortest computation structure which is affordable and less time consuming. Rodias [15] designed a combined technique for fertilizer application using linear programming. The proposed method used linear programming for optimizing several strategies of fertilizer such as harvesting handling operation, organic fertilization, mineral, and other several usage. These strategies are mapped into linear programming for solving different issues of the system. Kumari et al. [16] designed a model for routing technique in wireless network using geometric optimization technique. The proposed method is based on fusion of two intelligent techniques based on mathematical modeling such as game theory and geometric programming. Game theory is used to model the conflicting strategies of the network efficiently, and geometric programming is used to estimate non-linear parameters of the network efficiently. The combination of both efficiently enhance the network lifetime of the wireless network. Mishra et al. [17] proposed a model for grinding process based on fuzzy logic. The proposed method is based on an intelligent operation which is named as compensatory operator. It is based on weighted factoring technique. The proposed method basically optimized several process such as speed, density, sectional area which is cross, in this model, all these parameters are imprecise which are compact and model by membership function of the fuzzy logic. Tripathi and Das [18] designed a robustness method for routing in ad-hoc network.

Here, network is based on transparent system and heterogeneous in nature. Basic key elements of this network is fusion of linear programing and game theory. It consists of two objective functions for two players where both players are competitive to each other. Finally, it helps to reduce conflicting nature and pave the complexity of the network and find optimal route. Das et al. [19] designed an efficient routing method for ad-hoc network where the nature of the ad hoc is wireless. The complete work is based on two stages, first stage graph initiation phase for nodes with source and destination nodes and second stage contains route decision system named as reward calculation by the help of input parameters. Finally, it helps to reduce the complexity as well as uncertainty of the network. In WSN, data is gathered from multiple homogeneous or heterogeneous sources because, real life data is connected with different IoT, IoV, or cloud environment. So, it is difficult to keep the natures of the data in same structure. Information retrieval [20] is very important part in modern research areas which indicates collect information that are stored in unstructured form based on multiple local languages and process it in particular pattern after observing. Hao et al. [21] designed an evaluation system for big data analysis. This data is based on IoV where it means internet of vehicle. This proposal is based on K-means algorithm that is used here as a clustering. In this work, different behavior of the driving are involve for controlling vehicle. Finally, it helps in reducing fuel consumption and helps in transportation globally. Lin et al. [22] designed a load identification technique for residential system using quadratic programming. The proposed method is based on power managing system which is manage by modeling quadratic programming efficiently. The proposed method efficiently manage and control non-intrusive load of the residential system. Nazemi [23] proposed a neural network-based convex problem using quadratic programming. The main aim of this method is to solve convex quadratic model based on some real life applications. In this method, strictly convex problem is solved by modeling linear constraints of the quadratic programming. Lin et al. [24] proposed a method for quadratic programming for solving dual algorithm. The proposed method is based on strictly convex model based on two constraints such as box constraints and equality constraints. It is suitable for distributed system for optimizing large applications. The proposed method finally solve the two issues such as power flow and power state system of the model efficiently.

3 Proposed Method

In this section, the proposed method is illustrated efficiently. The proposed method is based on strategy management technique of two company named as Company A and Company B. Each company has several strategy named as Strategy 1, Strategy 2, …, Strategy n. These strategies may be anything based on company policies shown in Table 1. In this model, four strategies are taken for mathematical modeling purpose shown in Table 1, whose labeled marked in third column, where i varies 1–4.

Table 1 Names of company strategies

S. No.	Description	Label
1	Discount	a_i
2	Free item "X," if you buy item "Y"	b_i
3	Gift Voucher	c_i
4	Free Delivery	d_i

Table 2 Membership functions of company strategy

Linguistic variable	Range	Notation
Low	(0–100)	S_{L}
Medium	(80–150)	S_{M}
High	(130–250)	S_{H}
Very High	(240–400)	S_{VH}

The above-mentioned strategies or offers are linguistic in terms of real life applications where it varies customer to customer. So, the above-mentioned strategies are distributed in terms of fuzzy variables where highest unit is considered as 400 unit. The membership functions of fuzzy variables shown in Table 2.

The proposed method uses game theory technique for maintaining and managing different conflicting strategies of the company. Let in this model both companies are players in the context of game theory where Company A is Player A means first player, and Company B is Player B means second player. Both company are opponent of each and other. So, both are want to maximize their profit and minimize their losses. Let here, Company A is the winner and Company B is the looser. In indicates with context of game theory, Company A want to increases its own profits and Company B want to decreases its own losses. Let consider Company A apply the mentioned strategies of Table 2 as forward direction means "Low" to "Very High" and Company B apply the mentioned strategies of Table 2 as backward direction. So, applying strategies of both companies are reverse to each and other in the market. It indicates, when Company A apply the strategy S_{L} for selling the product then Company B apply the strategy S_{VH} for selling the same product. The mathematical model for quadratic programming of both company shown in Eqs. (1) and (2), where Z_1 and Z_2 are objective function of both model and p_1 to p_4 are different profit by using different strategies of both companies. Company A uses a_1 to a_5 for discount, b_1 to b_5 for free item 'X', if you buy item 'Y', c_1 to c_5 for gift voucher, and d_1 to d_5 for free delivery. Company B uses a_6 to a_{10} for discount, b_6 to b_{10} for free item 'X', if you buy item 'Y', c_6 to c_{10} for gift voucher, and d_6 to d_{10} for free delivery. Both models uses maximization method for maximizing profit of the companies as p_1 to p_4 by applying four different strategies. As Company A is the winner, so it maximizes its profit and Company B is looser, so it minimizes its losses by maximizing its profits. The optimize values along with objective value and their decision variables along with linguistic variables taken shown in Tables 3, 4, 5, 6, 7, 8, 9 and 10 for Company A and Tables 11, 12, 13, 14, 15, 16, 17 and 18 for Company B. Both optimization of

Table 3 Dataset of company A under **"Low"** linguistic variable under 1000 investment

a_i = Low (0–100)	b_i = Low (0–100)	c_i = Low (0–100)	d_i = Low (0–100)
$a_1 = 100$	$b_1 = 90$	$c_1 = 15$	$d_1 = 30$
$a_2 = 90$	$b_2 = 40$	$c_2 = 75$	$d_2 = 40$
$a_3 = 80$	$b_3 = 20$	$c_3 = 20$	$d_3 = 70$
$a_4 = 60$	$b_4 = 80$	$c_4 = 25$	$d_4 = 90$
$a_5 = 20$	$b_5 = 25$	$c_5 = 50$	$d_5 = 100$

Objective value $(Z_1) = 161.2797$ with $p_1 = 0.000000$, $p_2 = 9.756098$, $p_3 = 8.130081$, $p_4 = 0.000000$

Table 4 Dataset of company A under **"Medium"** linguistic variable under 1000 investment

a_i = Medium (80–150)	b_i = Medium (80–150)	c_i = Medium (80–150)	d_i = Medium (80–150)
$a_1 = 100$	$b_1 = 90$	$c_1 = 150$	$d_1 = 130$
$a_2 = 90$	$b_2 = 100$	$c_2 = 95$	$d_2 = 140$
$a_3 = 80$	$b_3 = 120$	$c3 = 110$	$d_3 = 150$
$a_4 = 100$	$b_4 = 80$	$c_4 = 125$	$d_4 = 90$
$a_5 = 80$	$b_5 = 125$	$c_5 = 150$	$d_5 = 80$

Objective value $(Z_1) = 100.00000$ with $p_1 = 10.000000$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 0.000000$

Table 5 Dataset of company A under **"High"** linguistic variable under 1000 investment

a_i = High (130–250)	b_i = High (130–250)	c_i = High (130–250)	d_i = High (130–250)
$a_1 = 130$	$b_1 = 180$	$c_1 = 160$	$d_1 = 170$
$a_2 = 150$	$b_2 = 170$	$c_2 = 145$	$d_2 = 250$
$a_3 = 170$	$b_3 = 60$	$c_3 = 190$	$d_3 = 110$
$a_4 = 140$	$b_4 = 190$	$c_4 = 175$	$d_4 = 170$
$a_5 = 200$	$b_5 = 225$	$c_5 = 150$	$d_5 = 160$

Objective value $(Z_1) = 27.70083$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 5.263158$, $p_4 = 0.000000$

Table 6 Dataset of company A under **"Very High"** linguistic variable under 1000 investment

a_i = Very High (240–400)	b_i = Very High (240–400)	c_i = Very High (240–400)	d_i = Very High (240–400)
$a_1 = 240$	$b_1 = 370$	$c_1 = 350$	$d_1 = 360$
$a_2 = 280$	$b_2 = 300$	$c_2 = 310$	$d_2 = 340$
$a_3 = 260$	$b_3 = 290$	$c_3 = 380$	$d_3 = 310$
$a_4 = 360$	$b_4 = 270$	$c_4 = 260$	$d_4 = 270$
$a_5 = 370$	$b_5 = 355$	$c_5 = 390$	$d_5 = 290$

Objective value $(Z_1) = 7.716049$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 0.0000$, $p_4 = 2.77778$

Table 7 Dataset of company B under **"Very High"** linguistic variable under 1000 investment

a_i = Very High (240–400)	b_i = Very High (240–400)	c_i = Very High (240–400)	d_i = Very High (240–400)
$a_6 = 240$	$b_6 = 290$	$c_6 = 250$	$d_6 = 300$
$a_7 = 290$	$b_7 = 400$	$c_7 = 375$	$d_7 = 280$
$a_8 = 280$	$b_8 = 380$	$c_8 = 320$	$d_8 = 350$
$a_9 = 360$	$b_9 = 280$	$c_9 = 250$	$d_9 = 280$
$a_{10} = 270$	$b_{10} = 255$	$c_{10} = 350$	$d_{10} = 250$

Objective value $(Z_2) = 7.111111$ with $p_1 = 0.000000, p_2 = 0.000000, p_3 = 2.666667, p_4 = 0.00000$

Table 8 Dataset of company B under **"High"** linguistic variable under 1000 investment

a_i = High (130–250)	b_i = High (130–250)	c_i = High (130–250)	d_i = High (130–250)
$a_6 = 130$	$b_6 = 190$	$c_6 = 150$	$d_6 = 130$
$a_7 = 140$	$b_7 = 150$	$c_7 = 195$	$d_7 = 240$
$a_8 = 180$	$b_8 = 160$	$c_8 = 170$	$d_8 = 150$
$a_9 = 160$	$b_9 = 180$	$c_9 = 145$	$d_9 = 190$
$a_{10} = 190$	$b_{10} = 225$	$c_{10} = 250$	$d_{10} = 180$

Objective value $(Z_2) = 17.36111$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 4.166667$

Table 9 Dataset of company B under **"Medium"** linguistic variable under 1000 investment

a_i = Medium (80–150)	b_i = Medium (80–150)	c_i = Medium (80–150)	d_i = Medium (80–150)
$a_6 = 80$	$b_6 = 90$	$c_6 = 130$	$d_6 = 120$
$a_7 = 100$	$b_7 = 110$	$c_7 = 105$	$d_7 = 130$
$a_8 = 80$	$b_8 = 100$	$c_8 = 130$	$d_8 = 140$
$a_9 = 110$	$b_9 = 90$	$c_9 = 115$	$d_9 = 90$
$a_{10} = 100$	$b_{10} = 145$	$c_{10} = 140$	$d_{10} = 110$

Objective value $(Z_2) = 82.644463$ with $p_1 = 9.090909$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 0.00000$

Table 10 Dataset of company B under **"Low"** linguistic variable under 1000 investment

a_i = Low (0–100)	b_i = Low (0–100)	c_i = Low (0–100)	d_i = Low (0–100)
$a_6 = 10$	$b_6 = 40$	$c_6 = 95$	$d_6 = 90$
$a_7 = 70$	$b_7 = 90$	$c_7 = 15$	$d_7 = 100$
$a_8 = 30$	$b_8 = 80$	$c_8 = 90$	$d_8 = 80$
$a_9 = 20$	$b_9 = 50$	$c_9 = 65$	$d_9 = 30$
$a_{10} = 90$	$b_{10} = 85$	$c_{10} = 70$	$d_{10} = 90$

Objective value $(Z_2) = 111.1111$ with $p_1 = 3.333333, p_2 = 0.000000, p_3 = 0.000000, p_4 = 0.00000$

Table 11 Dataset of company A under **"Low"** linguistic variable under 2000 investment

a_i = Low (0–100)	b_i = Low (0–100)	c_i = Low (0–100)	d_i = Low (0–100)
$a_1 = 10$	$b_1 = 50$	$c_1 = 65$	$d_1 = 40$
$a_2 = 30$	$b_2 = 10$	$c_2 = 55$	$d_2 = 10$
$a_3 = 70$	$b_3 = 50$	$c_3 = 40$	$d_3 = 20$
$a_4 = 60$	$b_4 = 90$	$c_4 = 55$	$d_4 = 60$
$a_5 = 80$	$b_5 = 45$	$c_5 = 30$	$d_5 = 90$

Objective value $(Z_1) = 981.2745$ with $p_1 = 12.04819$, $p_2 = 0.000000$, $p_3 = 28.91566$, $p_4 = 0.000000$

Table 12 Dataset of company A under **"Medium"** linguistic variable under 2000 investment

a_i = Medium (80–150)	b_i = Medium (80–150	c_i = Medium (80–150)	d_i = Medium (80–150)
$a_1 = 120$	$b_1 = 90$	$c_1 = 130$	$d_1 = 120$
$a_2 = 80$	$b_2 = 100$	$c_2 = 90$	$d_2 = 110$
$a_3 = 100$	$b_3 = 120$	$c_3 = 100$	$d_3 = 140$
$a_4 = 130$	$b_4 = 90$	$c_4 = 105$	$d_4 = 80$
$a_5 = 140$	$b_5 = 145$	$c_5 = 130$	$d_5 = 90$

Objective value $(Z_1) = 204.0816$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 14.28571$

Table 13 Dataset of company A under **"High"** linguistic variable under 2000 investment

a_i = High (130–250)	b_i = High (130–250)	c_i = High (130–250)	d_i = High (130–250)
$a_1 = 130$	$b_1 = 140$	$c_1 = 190$	$d_1 = 130$
$a_2 = 190$	$b_2 = 190$	$c_2 = 165$	$d_2 = 240$
$a_3 = 200$	$b_3 = 130$	$c_3 = 140$	$d_3 = 150$
$a_4 = 240$	$b_4 = 150$	$c_4 = 145$	$d_4 = 190$
$a_5 = 130$	$b_5 = 205$	$c_5 = 190$	$d_5 = 200$

Objective value $(Z_1) = 110.8033$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 10.52632$, $p_4 = 0.000000$

Table 14 Dataset of company A under **"Very High"** linguistic variable under 2000 investment

a_i = Very High (240–400)	b_i = Very High (240–400)	c_i = Very High (240–400)	d_i = Very High (240–400)
$a_1 = 280$	$b_1 = 310$	$c_1 = 250$	$d_1 = 360$
$a_2 = 290$	$b_2 = 320$	$c_2 = 350$	$d_2 = 340$
$a_3 = 360$	$b_3 = 390$	$c_3 = 330$	$d_3 = 310$
$a_4 = 290$	$b_4 = 370$	$c_4 = 360$	$d_4 = 270$
$a_5 = 300$	$b_5 = 305$	$c_5 = 390$	$d_5 = 260$

Objective value $(Z_1) = 30.86420$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 5.555556$

Table 15 Dataset of company B under **"Very High"** linguistic variable under 2000 investment

a_i = Very High (240–400)	b_i = Very High (240–400)	c_i = Very High (240–400)	d_i = Very High (240–400)
$a_6 = 260$	$b_6 = 310$	$c_6 = 250$	$d_6 = 380$
$a_7 = 390$	$b_7 = 350$	$c_7 = 325$	$d_7 = 260$
$a_8 = 250$	$b_8 = 320$	$c_8 = 380$	$d_8 = 300$
$a_9 = 320$	$b_9 = 380$	$c_9 = 290$	$d_9 = 250$
$a_{10} = 250$	$b_{10} = 245$	$c_{10} = 310$	$d_{10} = 290$

Objective value $(Z_2) = 27.70083$ with $p_1 = 0.000000$, $p_2 = 0.000000$, $p_3 = 0.0000000$, $p_4 = 5.26158$

Table 16 Dataset of company B under **"High"** linguistic variable under 2000 investment

a_i = High (130–250)	b_i = High (130–250)	c_i = High (130–250)	d_i = High (130–250)
$a_6 = 140$	$b_6 = 200$	$c_6 = 180$	$d_6 = 180$
$a_7 = 190$	$b_7 = 180$	$c_7 = 185$	$d_7 = 230$
$a_8 = 130$	$b_8 = 170$	$c_8 = 160$	$d_8 = 200$
$a_9 = 150$	$b_9 = 210$	$c_9 = 135$	$d_9 = 160$
$a_{10} = 170$	$b_{10} = 205$	$c_{10} = 140$	$d_{10} = 170$

Objective value $(Z_2) = 90.70295$ with $p_1 = 0.000000, p_2 = 9.523810, p_3 = 0.000000, p_4 = 0.00000$

Table 17 Dataset of company B under **"Medium"** linguistic variable under 2000 investment

a_i = Medium (80–150)	b_i = Medium (80–150)	c_i = Medium (80–150)	d_i = Medium (80–150)
$a_6 = 90$	$b_6 = 120$	$c_6 = 130$	$d_6 = 130$
$a_7 = 100$	$b_7 = 80$	$c_7 = 145$	$d_7 = 120$
$a_8 = 90$	$b_8 = 150$	$c_8 = 140$	$d_8 = 130$
$a_9 = 120$	$b_9 = 90$	$c_9 = 95$	$d_9 = 90$
$a_{10} = 130$	$b_{10} = 125$	$c_{10} = 120$	$d_{10} = 80$

Objective value $(Z_2) = 173.1302$ with $p_1 = 10.52632$, $p_2 = 0.000000$, $p_3 = 0.000000$, $p_4 = 7.894737$

Table 18 Dataset of company B under **"Low"** linguistic variable under 2000 investment

a_i = Low (0–100)	b_i = Low (0–100)	c_i = Low (0–100)	d_i = Low (0–100)
$a_6 = 60$	$b_6 = 90$	$c_6 = 45$	$d_6 = 40$
$a_7 = 40$	$b_7 = 50$	$c_7 = 65$	$d_7 = 90$
$a_8 = 50$	$b_8 = 70$	$c_8 = 60$	$d_8 = 70$
$a_9 = 20$	$b_9 = 90$	$c_9 = 15$	$d_9 = 60$
$a_{10} = 100$	$b_{10} = 35$	$c_{10} = 40$	$d_{10} = 20$

Objective value $(Z_2) = 946.7456$ with $p_1 = 0.000000, p_2 = 0.000000, p_3 = 30.76923, p_4 = 0.00000$

the mathematical model is analyzed based on two investments of the company first is 1000 and second is 2000. It reflects the effectiveness of the models in terms of quadratic programming and fuzzy logic optimization models.

$$\begin{aligned} &\text{Maximize}: \ Z_1 = (p_1)^2 + (p_2)^{2+} + (p_3)^2 + (p_4)^2 \\ &\text{Subject to constraints}: a_1 p_1 + b_1 p_2 + c_1 p_3 + d_1 p_4 \le 1000 \\ &\qquad a_2 p_1 + b_2 p_2 + c_2 p_3 + d_2 p_4 \le 1000 \\ &\qquad a_3 p_1 + b_3 p_2 + c_3 p_3 + d_3 p_4 \le 1000 \\ &\qquad a_4 p_1 + b_4 p_2 + c_4 p_4 + d_4 p_4 \le 1000 \\ &\qquad a_5 p_1 + b_5 p_2 + c_5 p_3 + d_5 p_4 \le 1000 \\ &\qquad p_i \ge 0, \text{ where } i \text{ varies } 1 - 4 \end{aligned} \tag{1}$$

$$\begin{aligned} &\text{Maximize } Z_2 = (p_1)^2 + (p_2)^{2+} + (p_3)^2 + (p_4)^2 \\ &\text{Subject to constraints}: a_6 p_1 + b_6 p_2 + c_6 p_3 + d_6 p_4 \le 2000 \\ &\qquad a_7 p_1 + b_7 p_2 + c_7 p_3 + d_7 p_4 \le 2000 \\ &\qquad a_8 p_1 + b_8 p_2 + c_8 p_3 + d_8 p_4 \le 2000 \\ &\qquad a_9 p_1 + b_9 p_2 + c_9 p_3 + d_9 p_4 \le 2000 \\ &\qquad a_{10} p_1 + b_{10} p_2 + c_{10} p_3 + d_{10} p_4 \le 2000 \\ &\qquad p_i \ge 0, \text{ where } i \text{ varies } 1 - 4. \end{aligned} \tag{2}$$

The above-mentioned Tables 3, 4, 5, 6, 7, 8, 9, 10 and Tables 11, 12, 13, 14, 15, 16, 17, 18 shows data analysis of both companies as Company A and Company B based on four strategies as a_i, b_i, c_i, and d_i where i varies 1–5 for Company A and 6–10 for Company B. The summarize dataset is shown in Tables 19 and 20. Finally, shows

Table 19 Objective values based on different linguistic variables of Company A

Under 1000 investment			
Low	Medium	High	Very High
161.2797	100.0000	27.70083	7.716049
Under 2000 investment			
Low	Medium	High	Very High
981.2745	204.0816	110.8033	30.86420

Table 20 Objective values based on different linguistic variables of Company B

Under 1000 investment			
Low	Medium	High	Very High
111.1111	82.64463	17.36111	7.111111
Under 2000 investment			
Low	Medium	High	Very High
946.7456	173.1302	90.70295	27.70083

Company A profit is more than Company B because, Company A is the winner and Company B is the loser. Company A follows sequential strategy of the market, it enhance and chooses strategy "Low" to "Very High." Profit of Company B is lesser than Company A, but it minimize its losses.

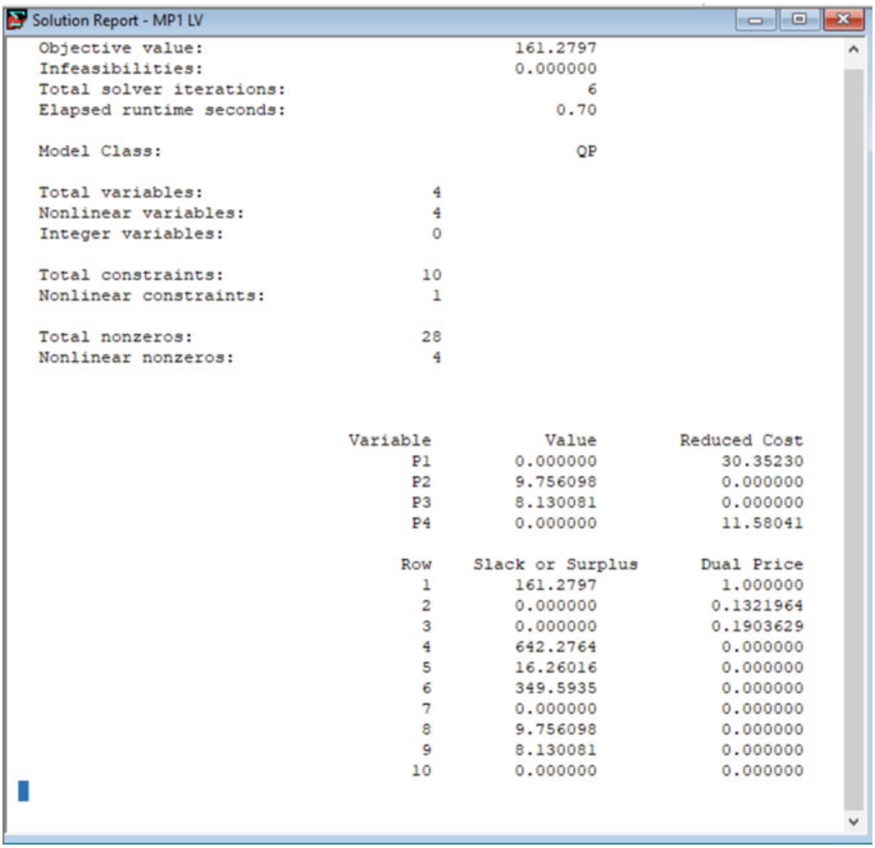
4 Simulation and Analysis

In this section, details of simulation and analysis are illustrated. The proposed method is simulated in LINGO optimization software which is based on non-linear formulation of quadratic programming. The basic parameters are shown in Table 21. In this simulation, Windows 10 pro is used along with MS Office 2013 and optimization software. In this method, two optimization models are used for Company A and Company B where both are competitors of one to another. In this work, game theory technique is used for maximizing the profit of the company. The company invest is two times first is 1000, and second time is 2000. The nature of the objective functions are non-linear where the nature of the constraints are linear of both company's models. Each model has five linear constraints with four linguistic variables such as "Low," "Medium," "High," and "Very High."

Figures 1, 2, 3 and 4 shows profits of Company A under 1000 investment for linguistic variables "Low," "Medium," "High," and "Very High." In these figure, it have been reflect as investment is increases bases on linguistic variables then profit of the company is decreases as 161.2797, 100.0000, 27.70083, and 7.716049.

Table 21 Simulation parameter details

Parameter	Description
Windows	Windows 10 pro
MS Office	2013
Optimization software	LINGO
Optimization models	2
Model names	Company A, Company B
Total investment	1000–2000
No. of times of investment	2
Constraints for first model	5
Constraints for second model	5
Nature of objectives	Non-linear
Nature of constraints	Linear
Linguistic variables in first model	4
Linguistic variables in second model	4
Name of the linguistic variables	Low, Medium, High, and Very High



Solution Report - MP1 LV

Objective value:	161.2797
Infeasibilities:	0.000000
Total solver iterations:	6
Elapsed runtime seconds:	0.70
Model Class:	QP
Total variables:	4
Nonlinear variables:	4
Integer variables:	0
Total constraints:	10
Nonlinear constraints:	1
Total nonzeros:	28
Nonlinear nonzeros:	4

Variable	Value	Reduced Cost
P1	0.000000	30.35230
P2	9.756098	0.000000
P3	8.130081	0.000000
P4	0.000000	11.58041

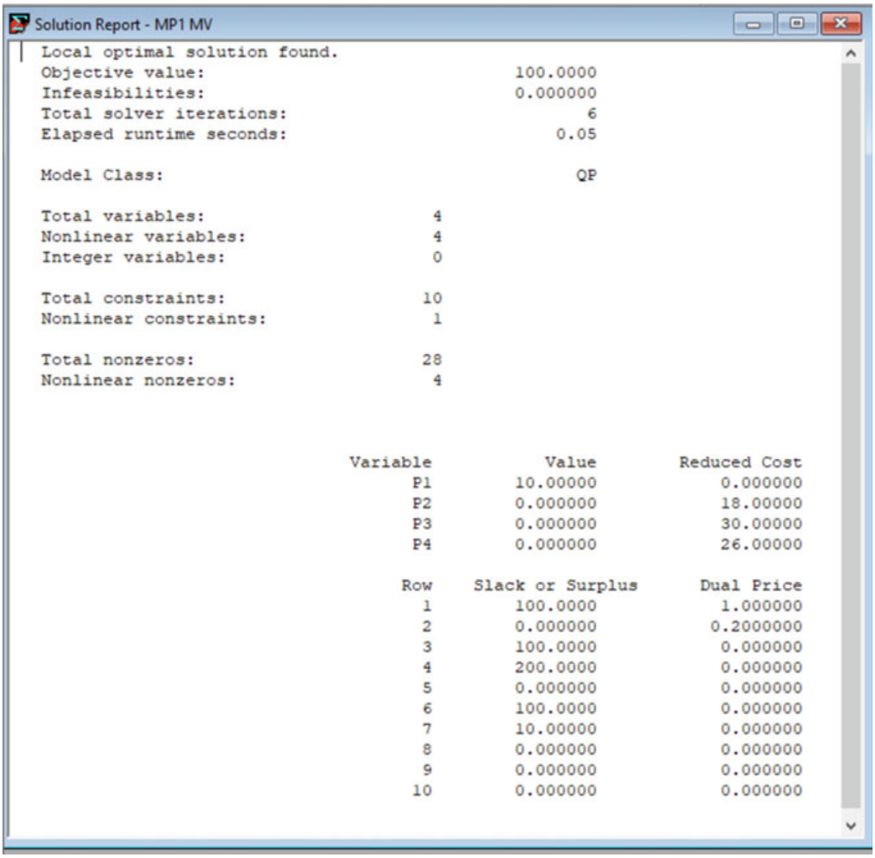
Row	Slack or Surplus	Dual Price
1	161.2797	1.000000
2	0.000000	0.1321964
3	0.000000	0.1903629
4	642.2764	0.000000
5	16.26016	0.000000
6	349.5935	0.000000
7	0.000000	0.000000
8	9.756098	0.000000
9	8.130081	0.000000
10	0.000000	0.000000

Fig. 1 Profit of Company A under 1000 investment for "Low" linguistic variable

Figures 5, 6, 7 and 8 shows profits of Company A under 2000 investment for linguistic variables "Low", "Medium," "High," and "Very High." In these figure, it have been reflect as investment is increases bases on linguistic variables then profit of the company is decreases as 981.2745, 204.0816, 110.8033 and 30.86420.

Figures 9, 10, 11 and 12 shows profits of Company B under 1000 investment for linguistic variables "Low," "Medium," "High," and "Very High." In these figure, it have been reflect as investment is increases bases on linguistic variables then profit of the company is decreases as 111.1111, 82.64463, 17.36111, and 7.111111.

Figures 13, 14, 15 and 16 shows profits of Company B under 2000 investment for linguistic variables "Low," "Medium," "High," and "Very High." In these figure, it have been reflect as investment is increases bases on linguistic variables then profit of the company is decreases as 946.7456, 173.1302, 90.70295, and 27.70083.



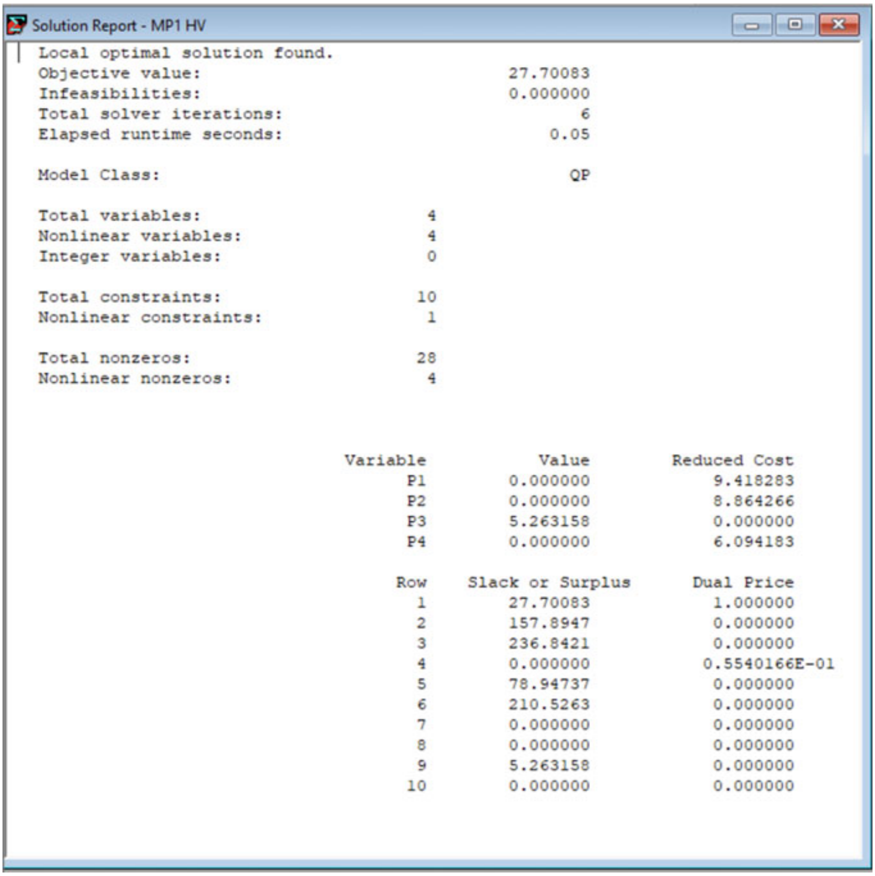
Solution Report - MP1 MV

```
Local optimal solution found.
Objective value:                              100.0000
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4

                     Variable           Value        Reduced Cost
                           P1        10.00000            0.000000
                           P2        0.000000            18.00000
                           P3        0.000000            30.00000
                           P4        0.000000            26.00000

                          Row    Slack or Surplus      Dual Price
                            1        100.0000            1.000000
                            2        0.000000           0.2000000
                            3        100.0000            0.000000
                            4        200.0000            0.000000
                            5        0.000000            0.000000
                            6        100.0000            0.000000
                            7        10.00000            0.000000
                            8        0.000000            0.000000
                            9        0.000000            0.000000
                           10        0.000000            0.000000
```

Fig. 2 Profit of Company A under 1000 investment for “Medium” linguistic variable

5 Conclusions

The proposed method is a strategy management technique for two company based on the fusion of fuzzy logic and game theory optimization. The combination of both efficiently helps to manage the conflicting strategy of the company. It shows winner company always follow right strategy instead of wrong strategy. The proposed method uses fuzzy logic for reducing uncertainty related information by estimating imprecise strategy of the customers. Day by day variations are raises in the market due to conflicting desire of the customers. Desire of each customer vary based on features and quality of the product. Hence, the proposed method efficient solve the proposed goal of the both companies.



Solution Report - MP1 HV

Local optimal solution found.
Objective value: 27.70083
Infeasibilities: 0.000000
Total solver iterations: 6
Elapsed runtime seconds: 0.05

Model Class: QP

Total variables: 4
Nonlinear variables: 4
Integer variables: 0

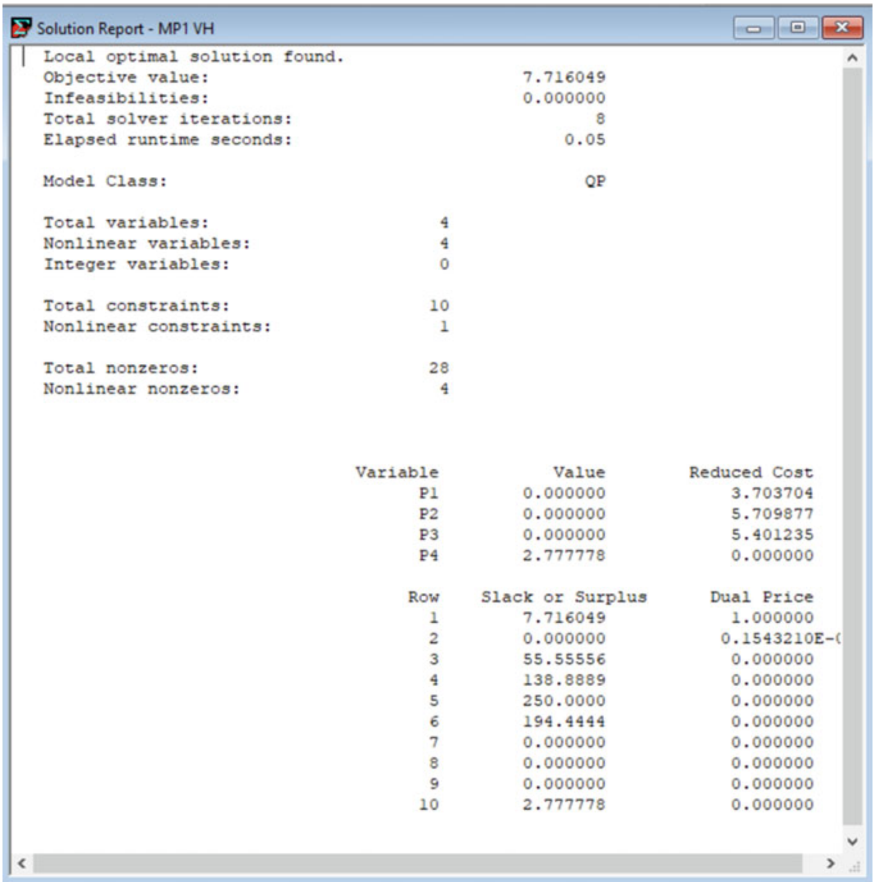
Total constraints: 10
Nonlinear constraints: 1

Total nonzeros: 28
Nonlinear nonzeros: 4

Variable	Value	Reduced Cost
P1	0.000000	9.418283
P2	0.000000	8.864266
P3	5.263158	0.000000
P4	0.000000	6.094183

Row	Slack or Surplus	Dual Price
1	27.70083	1.000000
2	157.8947	0.000000
3	236.8421	0.000000
4	0.000000	0.5540166E-01
5	78.94737	0.000000
6	210.5263	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	5.263158	0.000000
10	0.000000	0.000000

Fig. 3 Profit of Company A under 1000 investment for "High" linguistic variable



Solution Report - MP1 VH

Local optimal solution found.
Objective value: 7.716049
Infeasibilities: 0.000000
Total solver iterations: 8
Elapsed runtime seconds: 0.05

Model Class: QP

Total variables: 4
Nonlinear variables: 4
Integer variables: 0

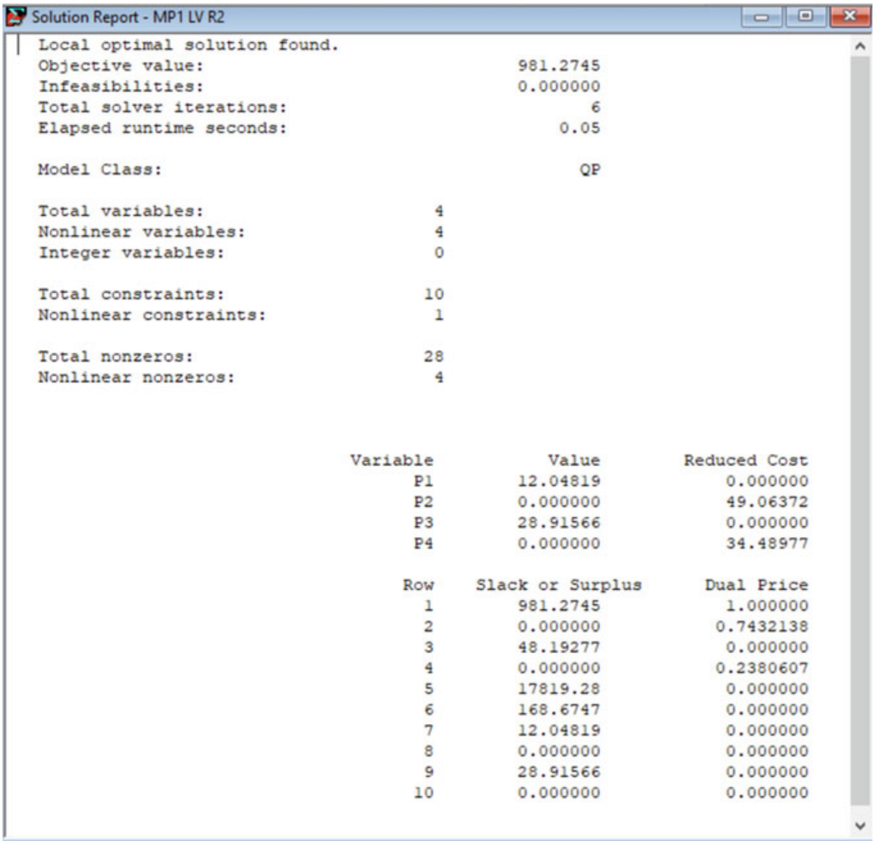
Total constraints: 10
Nonlinear constraints: 1

Total nonzeros: 28
Nonlinear nonzeros: 4

Variable	Value	Reduced Cost
P1	0.000000	3.703704
P2	0.000000	5.709877
P3	0.000000	5.401235
P4	2.777778	0.000000

Row	Slack or Surplus	Dual Price
1	7.716049	1.000000
2	0.000000	0.1543210E-(
3	55.55556	0.000000
4	138.8889	0.000000
5	250.0000	0.000000
6	194.4444	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	0.000000	0.000000
10	2.777778	0.000000

Fig. 4 Profit of Company A under 1000 investment for "Very High" linguistic variable



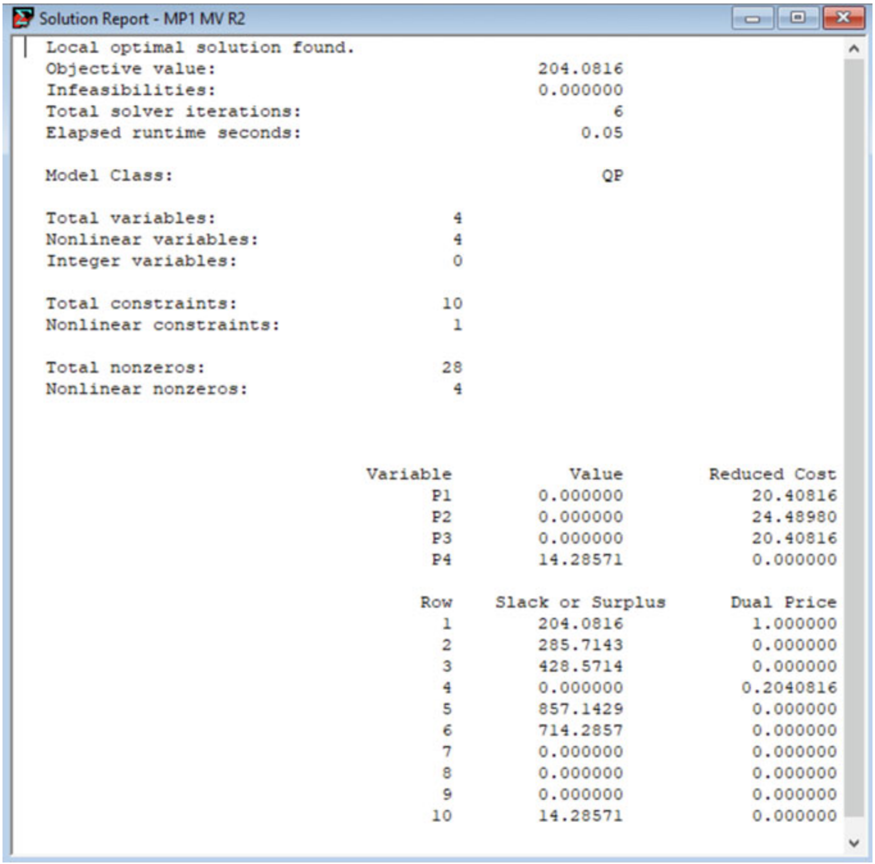
Solution Report - MP1 LV R2

Local optimal solution found.	
Objective value:	981.2745
Infeasibilities:	0.000000
Total solver iterations:	6
Elapsed runtime seconds:	0.05
Model Class:	QP
Total variables:	4
Nonlinear variables:	4
Integer variables:	0
Total constraints:	10
Nonlinear constraints:	1
Total nonzeros:	28
Nonlinear nonzeros:	4

Variable	Value	Reduced Cost
P1	12.04819	0.000000
P2	0.000000	49.06372
P3	28.91566	0.000000
P4	0.000000	34.48977

Row	Slack or Surplus	Dual Price
1	981.2745	1.000000
2	0.000000	0.7432138
3	48.19277	0.000000
4	0.000000	0.2380607
5	17819.28	0.000000
6	168.6747	0.000000
7	12.04819	0.000000
8	0.000000	0.000000
9	28.91566	0.000000
10	0.000000	0.000000

Fig. 5 Profit of Company A under 2000 investment for “Low” linguistic variable



Solution Report - MP1 MV R2

```
Local optimal solution found.
Objective value:                              204.0816
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4


                                Variable           Value        Reduced Cost
                                      P1        0.000000            20.40816
                                      P2        0.000000            24.48980
                                      P3        0.000000            20.40816
                                      P4        14.28571            0.000000

                                     Row    Slack or Surplus      Dual Price
                                       1        204.0816            1.000000
                                       2        285.7143            0.000000
                                       3        428.5714            0.000000
                                       4        0.000000           0.2040816
                                       5        857.1429            0.000000
                                       6        714.2857            0.000000
                                       7        0.000000            0.000000
                                       8        0.000000            0.000000
                                       9        0.000000            0.000000
                                      10        14.28571            0.000000
```

Fig. 6 Profit of Company A under 2000 investment for “Medium” linguistic variable

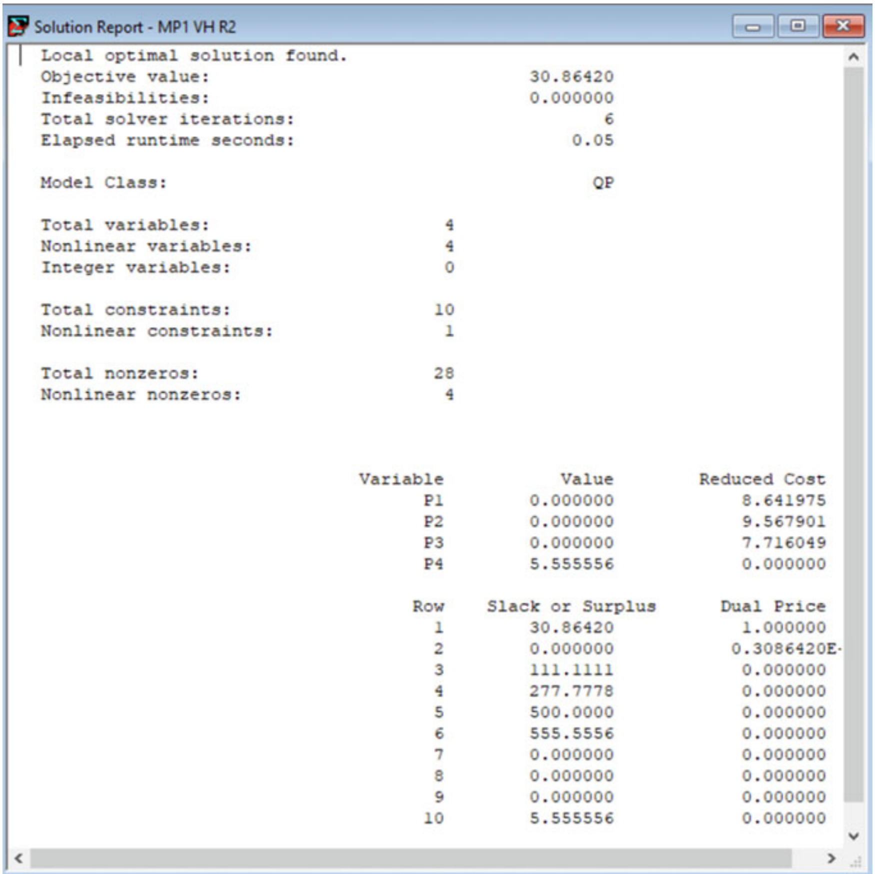
Solution Report - MP1 HV R2

Objective value:	110.8033
Infeasibilities:	0.000000
Total solver iterations:	6
Elapsed runtime seconds:	0.05
Model Class:	QP
Total variables:	4
Nonlinear variables:	4
Integer variables:	0
Total constraints:	10
Nonlinear constraints:	1
Total nonzeros:	28
Nonlinear nonzeros:	4

Variable	Value	Reduced Cost
P1	0.000000	14.40443
P2	0.000000	22.71468
P3	10.52632	0.000000
P4	0.000000	22.16066

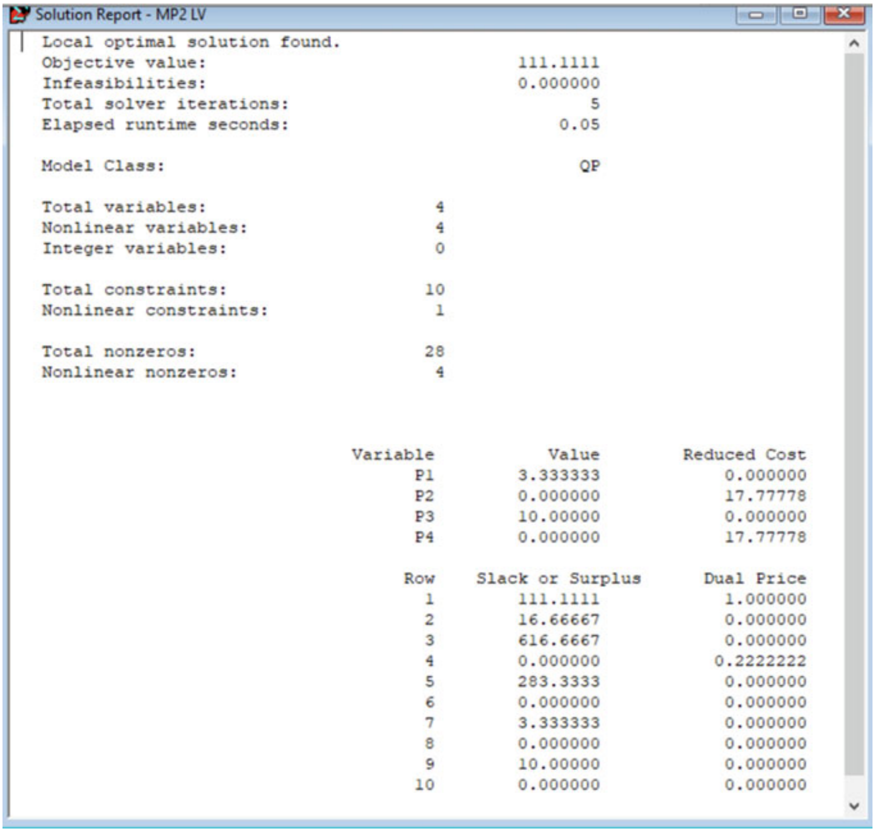
Row	Slack or Surplus	Dual Price
1	110.8033	1.000000
2	0.000000	0.000000
3	263.1579	0.000000
4	526.3158	0.000000
5	473.6842	0.000000
6	0.000000	0.1108033
7	0.000000	0.000000
8	0.000000	0.000000
9	10.52632	0.000000
10	0.000000	0.000000

Fig. 7 Profit of Company A under 2000 investment for "High" linguistic variable



```
Solution Report - MP1 VH R2

Local optimal solution found.
Objective value:                              30.86420
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4


                              Variable           Value        Reduced Cost
                                    P1        0.000000            8.641975
                                    P2        0.000000            9.567901
                                    P3        0.000000            7.716049
                                    P4        5.555556            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        30.86420            1.000000
                                     2        0.000000           0.3086420E-
                                     3        111.1111            0.000000
                                     4        277.7778            0.000000
                                     5        500.0000            0.000000
                                     6        555.5556            0.000000
                                     7        0.000000            0.000000
                                     8        0.000000            0.000000
                                     9        0.000000            0.000000
                                    10        5.555556            0.000000
```

Fig. 8 Profit of Company A under 2000 investment for "Very High" linguistic variable



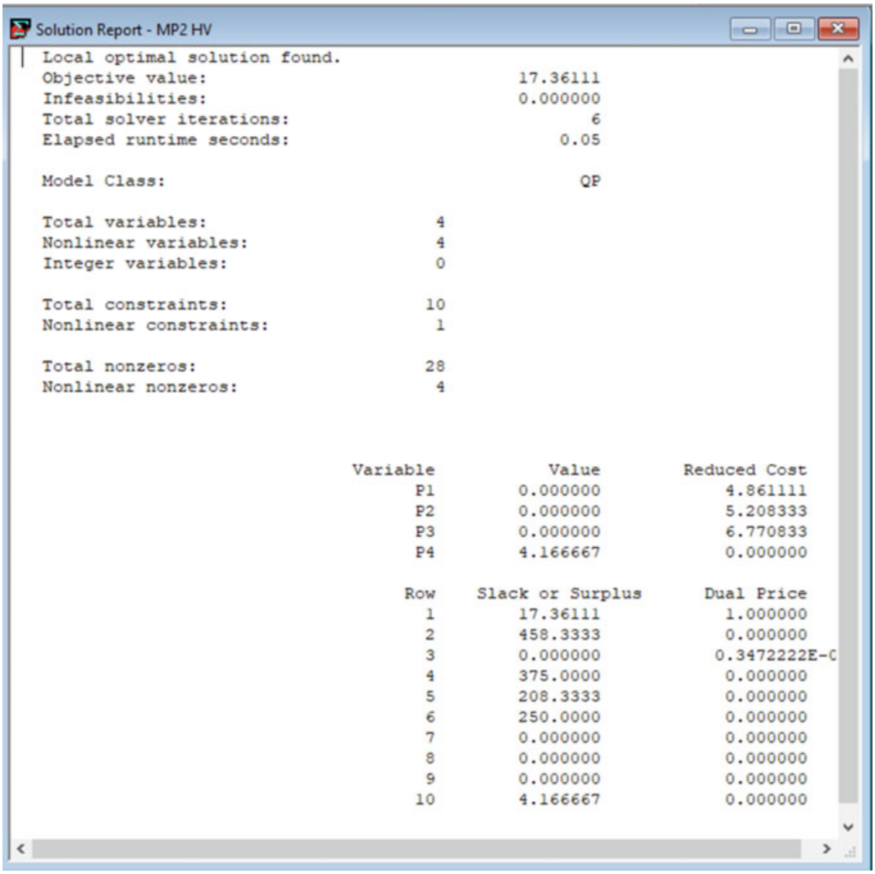
Solution Report - MP2 LV

```
Local optimal solution found.
Objective value:                              111.1111
Infeasibilities:                              0.000000
Total solver iterations:                             5
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                      4
Nonlinear variables:                  4
Integer variables:                    0

Total constraints:                   10
Nonlinear constraints:                1

Total nonzeros:                      28
Nonlinear nonzeros:                   4



                          Variable           Value        Reduced Cost
                                P1        3.333333            0.000000
                                P2        0.000000            17.77778
                                P3        10.00000            0.000000
                                P4        0.000000            17.77778

                               Row    Slack or Surplus      Dual Price
                                 1        111.1111            1.000000
                                 2        16.66667            0.000000
                                 3        616.6667            0.000000
                                 4        0.000000           0.2222222
                                 5        283.3333            0.000000
                                 6        0.000000            0.000000
                                 7        3.333333            0.000000
                                 8        0.000000            0.000000
                                 9        10.00000            0.000000
                                10        0.000000            0.000000
```

Fig. 9 Profit of Company B under 1000 investment for "Low" linguistic variable

```
Solution Report - MP2 MV

Local optimal solution found.
Objective value:                              82.64463
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4


                          Variable           Value        Reduced Cost
                                P1        9.090909            0.000000
                                P2        0.000000            14.87603
                                P3        0.000000            19.00826
                                P4        0.000000            14.87603

                               Row    Slack or Surplus      Dual Price
                                 1        82.64463            1.000000
                                 2        272.7273            0.000000
                                 3        90.90909            0.000000
                                 4        272.7273            0.000000
                                 5        0.000000           0.1652893
                                 6        90.90909            0.000000
                                 7        9.090909            0.000000
                                 8        0.000000            0.000000
                                 9        0.000000            0.000000
                                10        0.000000            0.000000
```

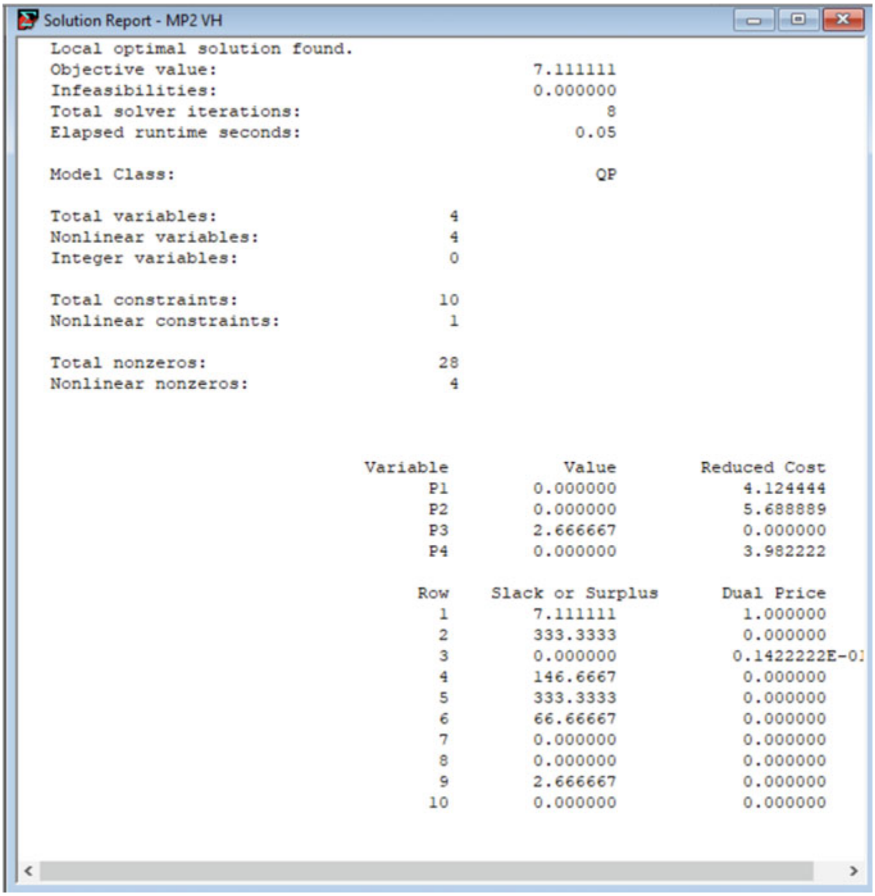
Fig. 10 Profit of Company B under 1000 investment for "Medium" linguistic variable



Solution Report - MP2 HV

```
Local optimal solution found.
Objective value:                              17.36111
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                      4
Nonlinear variables:                  4
Integer variables:                    0

Total constraints:                   10
Nonlinear constraints:                1

Total nonzeros:                      28
Nonlinear nonzeros:                   4


                              Variable           Value        Reduced Cost
                                    P1        0.000000            4.861111
                                    P2        0.000000            5.208333
                                    P3        0.000000            6.770833
                                    P4        4.166667            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        17.36111            1.000000
                                     2        458.3333            0.000000
                                     3        0.000000          0.3472222E-0
                                     4        375.0000            0.000000
                                     5        208.3333            0.000000
                                     6        250.0000            0.000000
                                     7        0.000000            0.000000
                                     8        0.000000            0.000000
                                     9        0.000000            0.000000
                                    10        4.166667            0.000000
```

Fig. 11 Profit of Company B under 1000 investment for “High” linguistic variable



```
Solution Report - MP2 VH
Local optimal solution found.
Objective value:                              7.111111
Infeasibilities:                              0.000000
Total solver iterations:                             8
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                      4
Nonlinear variables:                  4
Integer variables:                    0

Total constraints:                   10
Nonlinear constraints:                1

Total nonzeros:                      28
Nonlinear nonzeros:                   4


                              Variable           Value        Reduced Cost
                                    P1        0.000000            4.124444
                                    P2        0.000000            5.688889
                                    P3        2.666667            0.000000
                                    P4        0.000000            3.982222

                                   Row    Slack or Surplus      Dual Price
                                     1        7.111111            1.000000
                                     2        333.3333            0.000000
                                     3        0.000000          0.1422222E-01
                                     4        146.6667            0.000000
                                     5        333.3333            0.000000
                                     6        66.66667            0.000000
                                     7        0.000000            0.000000
                                     8        0.000000            0.000000
                                     9        2.666667            0.000000
                                    10        0.000000            0.000000
```

Fig. 12 Profit of Company B under 1000 investment for "Very High" linguistic variable

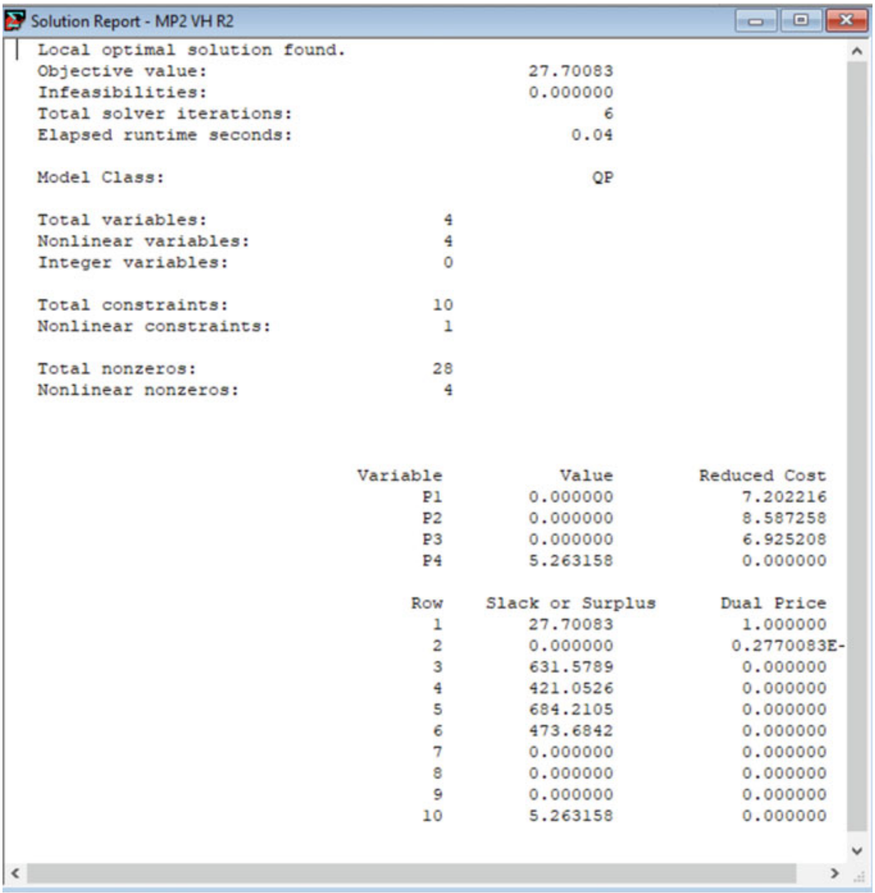
```
Solution Report - MP2 LV R2

Local optimal solution found.
Objective value:                              946.7456
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.04

Model Class:                                        QP

Total variables:                     4
Nonlinear variables:                 4
Integer variables:                   0

Total constraints:                  10
Nonlinear constraints:               1

Total nonzeros:                     28
Nonlinear nonzeros:                  4


                            Variable           Value        Reduced Cost
                                  P1        0.000000            37.86982
                                  P2        0.000000            47.33728
                                  P3        30.76923            0.000000
                                  P4        0.000000            85.20710

                                 Row    Slack or Surplus      Dual Price
                                   1        946.7456            1.000000
                                   2        615.3846            0.000000
                                   3        0.000000           0.9467456
                                   4        153.8462            0.000000
                                   5        1538.462            0.000000
                                   6        769.2308            0.000000
                                   7        0.000000            0.000000
                                   8        0.000000            0.000000
                                   9        30.76923            0.000000
                                  10        0.000000            0.000000
```

Fig. 13 Profit of Company B under 2000 investment for "Low" linguistic variable

Solution Report - MP2 MV R2

```
Local optimal solution found.
Objective value:                              173.1302
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.05

Model Class:                                        QP

Total variables:                      4
Nonlinear variables:                  4
Integer variables:                    0

Total constraints:                   10
Nonlinear constraints:                1

Total nonzeros:                      28
Nonlinear nonzeros:                   4



                              Variable           Value        Reduced Cost
                                    P1        10.52632            0.000000
                                    P2        0.000000            20.70637
                                    P3        0.000000            21.98753
                                    P4        7.894737            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        173.1302            1.000000
                                     2        26.31579            0.000000
                                     3        0.000000           0.4847645E-01
                                     4        26.31579            0.000000
                                     5        26.31579            0.000000
                                     6        0.000000           0.1246537
                                     7        10.52632            0.000000
                                     8        0.000000            0.000000
                                     9        0.000000            0.000000
                                    10        7.894737            0.000000
```

Fig. 14 Profit of Company B under 2000 investment for “Medium” linguistic variable



Solution Report - MP2 VH R2

```
Local optimal solution found.
Objective value:                              27.70083
Infeasibilities:                              0.000000
Total solver iterations:                             6
Elapsed runtime seconds:                          0.04

Model Class:                                        QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4


                              Variable           Value        Reduced Cost
                                    P1        0.000000            7.202216
                                    P2        0.000000            8.587258
                                    P3        0.000000            6.925208
                                    P4        5.263158            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        27.70083            1.000000
                                     2        0.000000           0.2770083E-
                                     3        631.5789            0.000000
                                     4        421.0526            0.000000
                                     5        684.2105            0.000000
                                     6        473.6842            0.000000
                                     7        0.000000            0.000000
                                     8        0.000000            0.000000
                                     9        0.000000            0.000000
                                    10        5.263158            0.000000
```

Fig. 15 Profit of Company B under 2000 investment for "High" linguistic variable

```
Solution Report - MP2 VH R2
Local optimal solution found.
Objective value:                          27.70083
Infeasibilities:                          0.000000
Total solver iterations:                         6
Elapsed runtime seconds:                      0.04

Model Class:                                    QP

Total variables:                    4
Nonlinear variables:                4
Integer variables:                  0

Total constraints:                 10
Nonlinear constraints:              1

Total nonzeros:                    28
Nonlinear nonzeros:                 4



                         Variable           Value        Reduced Cost
                               P1        0.000000            7.202216
                               P2        0.000000            8.587258
                               P3        0.000000            6.925208
                               P4        5.263158            0.000000

                              Row    Slack or Surplus      Dual Price
                                1        27.70083            1.000000
                                2        0.000000           0.2770083E-
                                3        631.5789            0.000000
                                4        421.0526            0.000000
                                5        684.2105            0.000000
                                6        473.6842            0.000000
                                7        0.000000            0.000000
                                8        0.000000            0.000000
                                9        0.000000            0.000000
                               10        5.263158            0.000000
```

Fig. 16 Profit of Company B under 2000 investment for "Very High" linguistic variable

References

1. Binh, H. T. T., & Dey, N. (Eds.). (2018). *Soft computing in wireless sensor networks.* CRC Press.
2. Binh, H. T. T., Hanh, N. T., Nghia, N. D., & Dey, N. (2020). Metaheuristics for maximization of obstacles constrained area coverage in heterogeneous wireless sensor networks. *Applied Soft Computing, 86,* 105939.
3. Antonucci, A., de Campos, C. P., Huber, D., & Zaffalon, M. (2015). Approximate credal network updating by linear programming with applications to decision making. *International Journal of Approximate Reasoning, 58,* 25–38.
4. Portman, V. T., Weill, R. D., Shuster, V. G., & Rubenchik, Y. L. (2003). Linear-programming-based assessments of geometrical accuracy: Standard presentation and application area. *International Journal of Machine Tools and Manufacture, 43*(10), 1023–1033.
5. Murmu, S., Jha, S. K., Burnwal, A. P., & Kumar, V. (2015). A proposed fuzzy logic based system for predicting surface roughness when turning hard faced components. *International Journal of Computer Applications, 125*(4).

6. Kumari, N., & Burnwal, A. P. (2017). Interactive fuzzy programming model in multi-objective inventory control problem using various operators. *International Journal of Students' Research in Technology & Management, 5*(4), 18–26.
7. Kumari, P., Jain, P. K., & Pamula, R. (2018, March). An efficient use of ensemble methods to predict students academic performance. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)* (pp. 1–6). IEEE.
8. Das, S. K., & Tripathi, S. (2015). Energy efficient routing protocol for manet based on vague set measurement technique. *Procedia Computer Science, 58,* 348–355.
9. Das, S. K., & Tripathi, S. (2016). Energy efficient routing protocol for MANET using vague set. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving* (pp. 235–245). Springer, Singapore.
10. Punam, K., Pamula, R., & Jain, P. K. (2018, September). A two-level statistical model for big mart sales prediction. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 617–620). IEEE.
11. Yang, W., Wang, X., Song, X., Yang, Y., & Patnaik, S. (2018). Design of intelligent transportation system supported by new generation wireless communication technology. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications* (pp. 715–732). IGI Global.
12. Jayakumar Loganathan, J., & Subbiah, J. (2020). Energy aware dynamic mode decision for cellular D2D communications by using integrated multi-criteria decision making model. *International Journal of Ambient Computing and Intelligence, 11*(3), 7 February 2020, IGI Global.
13. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015, February). Fuzzy based energy efficient multicast routing for ad-hoc network. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (pp. 1–5). IEEE.
14. Aboelmagd, Y. M. (2018). Linear programming applications in construction sites. *Alexandria Engineering Journal, 57*(4), 4177–4187.
15. Rodias, E. C., Sopegno, A., Berruto, R., Bochtis, D. D., Cavallo, E., & Busato, P. (2019). A combined simulation and linear programming method for scheduling organic fertiliser application. *Biosystems Engineering, 178,* 233–243.
16. Kumari, N., Kumar, A., & Burnwal, A. P. (2018, March). An application of geometric programming for designing optimal routing technique in wireless network. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)* (pp. 1–6). IEEE.
17. Mishra, B. K., Yadav, B., Jha, S. K., & Burnwal, A. P. (2015). Fuzzy set theory approach to model super abrasive grinding process using weighted compensatory operator. *International Journal of Research in Computer Applications and Robotics (IJRCAR), 3,* 62–68.
18. Das, S. K., & Tripathi, S. (2018). Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming. *Applied Intelligence, 48*(7), 1825–1845.
19. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015, February). Design of fuzzy based intelligent energy efficient routing protocol for wanet. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (pp. 1–4). IEEE.
20. Rasheed, I., & Banka, H. (2018, March). Query expansion in information retrieval for urdu language. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)* (pp. 1–6). IEEE.
21. Hao, R., Yang, H., & Zhou, Z. (2019). Driving behavior evaluation model base on big data from internet of vehicles. *International Journal of Ambient Computing and Intelligence (IJACI), 10*(4), 78–95.
22. Lin, S., Zhao, L., Li, F., Liu, Q., Li, D., & Fu, Y. (2016). A nonintrusive load identification method for residential applications based on quadratic programming. *Electric Power Systems Research, 133,* 241–248.

23. Nazemi, A. (2014). A neural network model for solving convex quadratic programming problems with some applications. *Engineering Applications of Artificial Intelligence, 32,* 54–62.
24. Lin, S. S., & Horng, S. C. (2009). A parallel dual-type algorithm for a class of quadratic programming problems and applications. *Expert Systems with Applications, 36*(3), 5190–5199.

A Novel Multilevel Classifier Hybrid Model for Intrusion Detection Using Machine Learning

Sunil Gautam, Hari Om, and Kumar Dixit

1 Introduction

Nowadays, the network security has become a challenging issue due to openness in Internet technology that has resulted in rapid growth of intrusion activities. The main problem to identify an attacker (insider or outsider attacker) that has entered in the system database with the intention to access or modify the information in a system. Researchers have designed some intrusion detection systems (IDSs) to handle this type of problem. In a single or network system, an event or group of events compromising of confidentiality, availability, and integrity of the system/source information is termed as an intrusion. The objective of IDS is to provide the protection to a single system or network from malicious actions. The IDSs store the pattern of activities and analyze the system activities. Any activity deviating from the pattern of the stored activities is assumed to be suspicious activity. The intrusion detection systems may be categorized into two groups: host-based intrusion detection systems (HIDSs) and network-based intrusion detection systems (NIDSs). An HIDS monitors and analyzes incoming and outgoing data packets of a single system, whereas a NIDS monitors and analyzes the inbound and outbound data packets of an entire network, for malicious activities [1, 2].

S. Gautam (✉)
Department Engineering and Computing, Institute of Advanced Research, Gandhinagar, India
e-mail: gautamsunil.cmri@gmail.com

H. Om · K. Dixit
Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India
e-mail: hariom4india@gmail.com

K. Dixit
e-mail: kumardixit88@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_12

IDS uses two types of approaches for detection of malicious activities: misuse and anomaly-based detection approaches [3]. In misuse-based approach, only known attacks or threats are found. It is also known as knowledge-based misuse detection because it uses the accumulated knowledge for identifying the suspicious activities [4]. The misuse detection uses the rule-based approach for detecting the network attacks [5]. The limitation of misuse detection is that it is applicable for recognized attacks only. The disadvantage of misuse detection is that it is not capable to identify un-recognized attacks [6].

The anomaly detection approach is better than the misuse-based approach because it can determine both the recognized and un-recongnized attacks in a system. The anomaly detection approach is also identified as behavior-based detection because it compares the normal behavior with the abnormal behavior [7]. It monitors the regular behavior as well as novel behavior that occurs in a single system or a network system for intrusive activities. Anomaly detection is a major concern for researchers because its presence in several applications, namely fraud recognition for credit cards, cyber-security, insurance or healthcare, and fault detection in safety critical systems [8]. An intrusive activity occurs in the network whenever an anomalous traffic pattern of data is sent by an unauthorized host [9]. Anomaly detection-based intrusion detection model has the ability to find out the intrusive activity without specifying attack models. Anomaly-based intrusion detection approach is more reliable as it is not easily befooled by small variation in patterns [10], and it is applicable for both HIDS and NIDS models. The HIDS model examines the intrusive activity in normal audit records (logs), whereas an NIDS model analyzes the network traffic records to detect anomalies. Both models measure an anomaly threshold and raise alarm when some specific attribute value of a novel record is found to be higher than the threshold [3]. This approach overcomes the limitations of the misuse detection approach by concerning the system activities rather than the attack behavior. The mechanism of this approach is carried out in two stages: training stage and detection stage. In training stage, the behavior of the system is analyzed in the absence of attacks that can be done by using a machine learning algorithm. In detection phase, this profile (normal behavior) is compared with the current behavior of the system, and any deviation is considered as a latent attack [11].

The reliability of an IDS depends on its detection accuracy of an attack because most of the IDSs face higher false alarm rate problem. It is a major concern for the researchers that require novel techniques to address. Data mining is an important approach for intrusion detection because it can easily analyze and retrieve the usable information from a large volume of dataset [12]. The data mining approach is generally used in a classification that can be helpful extracting similar patterns in a dataset, which can provide better understanding the behavior of a given large dataset. They are several classification approaches, namely as decision tree, neural network, naive Bayes, fuzzy set, etc. that can be helpful in designing the IDSs [13].

The rest of this book chapter structured as follows. In Sect. 2, we provide the related works, and Sect. 3 focuses on the theoretic aspect of the techniques. Section 4 introduces a hybrid model for IDS, and Sect. 5 provides the results and discussions. Finally, Sect. 6 concludes this chapter.

2 Related Work

IDS provides an alternate way to detect various kinds of attacks that occur in a network. The IDSs use various kinds of approaches for detecting the suspicious activities and also improving the detection accuracy. Nowadays, the researchers are exploring the data mining techniques for designing the IDSs to detect the attacks. In this techniques, generally use the kdd cup'99 dataset for evaluating the performance of IDSs. These techniques use classification model for determining the attacks in a dataset. The classification model train and test on the datasets that help predicating the normal and abnormal classes [14].

Zhang et al. discuss a NIDS random forest data mining technique for intrusion detection. It combines both anomaly and misuse detection approaches for uncertain behavior of a dataset [15]. Dhakar et al. discuss an intrusion detection framework for intrusion detection using the tree augmented naive Bayes (TAN) and reduced error pruning (REP) data mining technique. This model uses TAN and REP breed classification rules for detecting the intrusive activities [16]. Golmah discusses a method with C5.0 and support vector machine (SVM) that first generates C5.0 node information using kdd cup'99 dataset, and then, the generated node information is along with the original dataset information is provided to the SVM classifier for final output [17]. Morris et al. discuss IDS for reducing intrusive problems in power system by using data mining techniques for detecting the disturbance, normal control operations, and malicious activities in a power system [18].

Li discusses a hybrid evolutionary neural network IDS for detecting complex anomalous activities using the genetic algorithm. It evaluates the neural with crossover and weights adjustment mutation. Whenever previous mutation fails in training, then another mutation is selected randomly. This process repeats until there is no improvement in the system accuracy [19]. Zhang et al. discuss an intrusion detection system for network intrusive activities by combining the anomaly and misuse detection approaches. It applies the random forest technique on the kdd 99 dataset, which is a data mining algorithm specially designed for huge datasets. It generates several classification trees, every constructed by a different bootstrap. Every single tree gives a vote to specify the tree's decision about the class of the object. In this technique, no requirement for cross-validation or a test set an balanced estimate of analysis error. Malik et al. discuss an IDS determine probe attacks in a network [20]. They use the particle swarm optimization technique for attribute selection on kdd'99 dataset and then apply random forest technique for malicious activities detection. Panda et al. discuss a hybrid intelligent method by using principal component analysis and decision tree techniques for filtering the kdd'99 dataset and then apply intelligent decision technique radial basis function (RBF) for refining the detection accuracy and reducing the false alarm rate [21]. Govindarajan et al. discuss a neural network-based hybrid classification intrusion detection system that uses the ensemble-based multilayer perceptron and ensemble-based radial basis function for intrusion detection, which is superior to the base classifiers for intrusion detection in terms of prediction accuracy [4]. Powers et al. discuss an intrusion detection system by combining the advantages of anomaly and misuse detection approaches that is used to detect the denial of service and user to root attack in kdd cup'99 dataset with low false positive rate and high detection accuracy [22]. Nalini et al. discuss network ids using the genetic algorithm and principal component analysis for anomalous activities [23]. Zhou et al. discuss ids using the artificial immune system and principal component analysis neural network that provides high detection accuracy with low false alarm rate [24]. The principal component analysis is used for classification, and the artificial immune system is used to detect the suspicious network connections. There are mainly two demanding jobs for the misuse and anomaly detection methods: low detection rate and high false positive rate. There are very few hybrid approaches that discuss these problems simultaneously.

In our proposed model, we combine the anomaly and misuse detection methods for reducing the false positive rates and improved detection rate by using the data mining techniques.

3 Theoretic Aspects of Techniques

Here, we discuss about the multilevel classifier, which is a hybrid model of random tree and instance-based learning on K (IBK) technique. The main purpose of these multilevel classifier hybrid ids is to combine together anomaly and misuse detection technique for monitoring and analyzing the anomalous activities in a network. This model uses the particle swarm optimization (PSO), random tree forest, and instance-based learning on K (IBK) techniques to continuously monitor the network traffic and analyze the data packets.

3.1 Particle Swarm Optimization Techniques

Eberhart et al. introduced the particle swarm optimization (PSO) technique based on metaphor social communication such as human social behavior [25]. It is used in several applications such as neural network training, function optimization, classification problem, etc. A huge dataset contains several features, all of which are not useful for classification. The PSO removes the irrelevant features that help an intrusion detection system to improve performance and accuracy. Feature selection is a challenging task due to the multifarious relationship between the features, and the search space may increase or decrease with respect to available features in a dataset [26, 27]. Each attribute or element changes according to its velocity and updates its position and velocity according to its own experience as well as its neighbors. In search space problem is considered as a solution space, where particles cooperate to find the best position in it [28]. PSO algorithm begins with a set of particles that are randomly scattered in the problematic space. Every particle is allocated a random velocity and also declares a fitness function in particle's location. The PSO searches the best position for particles by minimizing the fitness function. It checks the fitness function of each particle in each iteration; each particle updated its velocity and computes its novel position. The new velocity is computed using its current velocity, its distance from its own best position, and its distance from the population's best position. Every element is updated by two "best" values in each iteration. The first best value, called as *pbest*, stores the fitness function value, and other best value, called as *gbest* (*global best*), is obtained by the swarm optimizer. If a particle takes part of the population as its topological neighbors, then the best value is known as *lbest* value [29].

3.2 Instance-Based Learning on K (IBK)

The instance-based learning on K (IBK) method uses the k-nearest neighbor algorithm. Its main goal is to classify a new object based on attributes, training, and testing samples. It uses the neighborhood classification as the predication value of a new query instance [30]. It assigns weights to the neighbors in such a way that the nearer neighbors contribute more to the average than the distant ones. A common weighting scheme assigns each neighbor a weight of 1/d, where d is the distance to a neighbor. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known [31, 32].

3.3 Random Tree

The instance-based learning on K (IBK) method uses the k-nearest neighbor algorithm. Its main goal is to classify a new object based on attributes, training, and testing samples. It uses the neighborhood classification as the predication value of a new query instance [30]. It assigns weights to the neighbors in such a way that the nearer neighbors contribute more to the average than the distant ones. A common weighting scheme assigns each neighbor a weight of 1/d, where d is the distance to a neighbor. The neighbors are taken from a set of objects for which the class (for k-NN classification), or the object property value (for k-NN regression) is known [31, 32].

3.4 KDD Cup'99 Dataset

Knowledge discovery and data mining tools are a dataset that is used for examining the malicious activities as it is a benchmark pf assessment of an intrusion detection system. Stolfo et al. developed at Defense Advanced Research Projects Agency (DARPA) for intrusion detection, assessment program that was managed MIT Lincoln Laboratory at MIT in 1998. The goal of this assessment program was

Table 1 Description of attacks

Attack	
Category	Sub-categories
DoS	back, land, pod, smurf, neptune, teardrop
U2R	buffer_overflow, rootkit, perl, loadmodule
R2L	ftp_write, guess_passwd,I map, multihop, phf, spy, warezclient, warezmaster
Probe	Satan, ipsweep, nmap, portsweep

to evaluate a research program in the area of intrusion detection. This dataset is used as training as well as testing dataset for detecting the attacks. The DARPA'98 dataset contains seven weeks training data and two weeks of testing data. The improved version of DARPA'98 dataset is KDD CUP' 99 dataset that contains about 4 gigabytes of compressed raw tcpdump data in which 5 million connection records in training dataset and 2 million records in test dataset. The KDD cup'99 contains approximately 4,900,000 records and 41 features, labeled as either normal or an attack. The main deficiency of the KDD dataset is data redundancy, i.e., 78% and 75% data records are repeated in train and test dataset, respectively [33].

Table 1 contains the attacks falling into four major attacks: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and probe attacks [34].

1. **DoS**: An attacker makes machine occupied or jam packed that it cannot assist a genuine request then machine denies one to entrance the machine.
2. **U2R**: An attacker has no authenticated account in the network; he takes the user's password by predicting and exploits the victim's local machine.
3. **R2L**: An attacker sends the packets to a mechanism over a network to which he is not authorized to access the machine.
4. **Probe**: An attacker scans the user machine and collects information about the networking devices for the drive of circumventing its safety panels.

In our work, we use the KDDD'99 dataset available on kdd.ics.uci.edu//databases/kddcup99/kddcup99.html. This dataset contains 7 symbolic and 34 continuous features that are grouped into four different categories. The first category contains 1–9 features labeled that is individual of TCP connections. The second category includes 10–22 content 9 features labeled. The third category includes 23–31 labeled features that are traffic features computed using a two second time window from destination to host [35]. The fourth category includes 32–42 features labeled.

4 Proposed Single-Level Hybrid Intrusion Detection Model

In this work, we use the KDD'99 dataset, a huge dataset, that contains 490,000 records and 41 features. In a huge dataset, the feature selection is a challenging issue due to the multifarious relationship between the features. In features selection phase, a set of network traffic attributes or features that are most effective attributes is extracted to construct the intrusion detection model by using the particle swarm optimization technique. We first make each category of dataset to contain only numeric values. The PSO removes less significant data and reduces the dataset dimensionality. Then, the random tree classifier is applied for identifying the known attacks and categorizing the dataset into classified and unclassified dataset. The classified data consists of known type of attacks, and the random tree classifier gives about 98.01% detection accuracy. The remaining part (unclassified data) is again categorized for the attacks by using the IBK classifier that uses the anomaly detection approach. The IBK classifier classifies the intrusive activity with 98.96% accuracy the remaining dataset. Figure 1, shows our proposed hybrid intrusion detection model that integrates the random tree

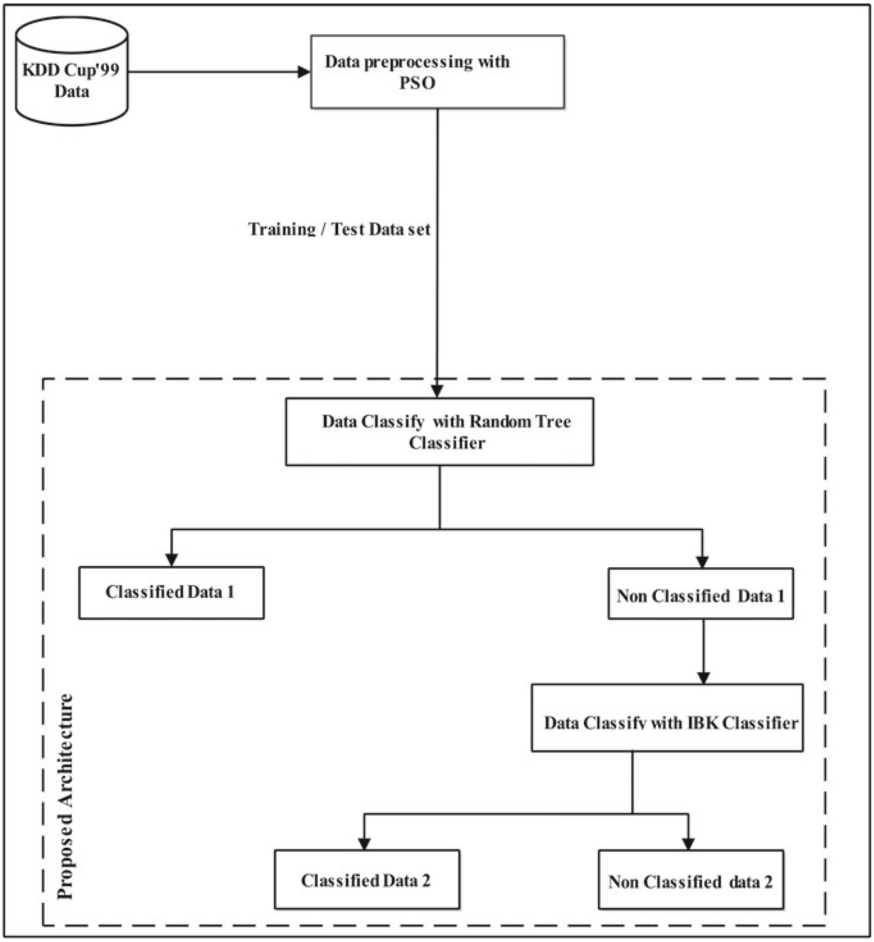


Fig. 1 Hybrid IDS model

and instance-based technique to make a hybrid classifier for detecting intrusion. The hybrid intrusion detection model provides the intrusion detection with 99.81% accuracy, which is significant for intrusion detection model.

4.1 A Multilevel Hybrid Classifier IDS Model

Here, we discuss about our proposed multilevel classifier hybrid intrusion detection system model for classification of attacks. In this model, we have combined two different data mining classifiers for detecting the known and unknown attacks. It works in two phases using anomaly and misuse detection approaches. The anomaly-based approach has higher false positive rate, which is not good for an ideal intrusion detection system; that's why we have employed the misuse detection approach. It has led us to adopt some modification adopts in the proposed model. The structure of a multilevel hybrid classifier model works in four stages as shown in Fig. 2.

In the first stage, the model classifies the known and unknown attacks by using the misuse detection approach. The second approach classifies the novel attacks, and the remaining dataset is classified by using the anomaly detection approach. In the third stage, the attacks are classified into DoS, U2R, R2L, and probe attacks. In the final stage, we have further classified the attacks into more specific attacks types. The DoS attacks are classified into subclasses that include back, land, pod, smurf, neptune, teardrop; the U2R attacks have been classified into subclasses that include buffer_overflow, rootkit, perl, loadmodule; the R2L are have been classified into subclasses such as ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster; and the probe attacks have been classified into subclasses attacks such as satan, ipsweep, nmap, portsweep.

The primary aim of stage 1 is minimize the data dimensionality of use experimental dataset because it is a huge dataset. The particle swarm optimization technique removes the irrelevant features from datasets, which reduce the processing time in this stage. These irrelevant features increased the false alarm rate and will produce hinder the process of identify intrusive. Some of these features may also be useless information. Now, the random tree classifier technique, apply on filtered dataset for detection of known attacks. At this stage, only 15 attributes, namely protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, logged_in, count, same_srv_rate, diff_srv_rate, dst_host_same_src_port_rate, is_guest_login, dst_host_count, num_file_creations, are selected for building the random tree for classification of attacks.

The stage 2 is essential for classification of remaining datasets because the random tree classifier classified the known attacks. In this stage, we have received the classified and unclassified dataset. In classified dataset contains both normal as well as abnormal connection records which are defined as known attacks and correctly classified by the random tree classifier. The unclassified dataset contains the novel connections, record which is not determined by random tree. The IBK classifier

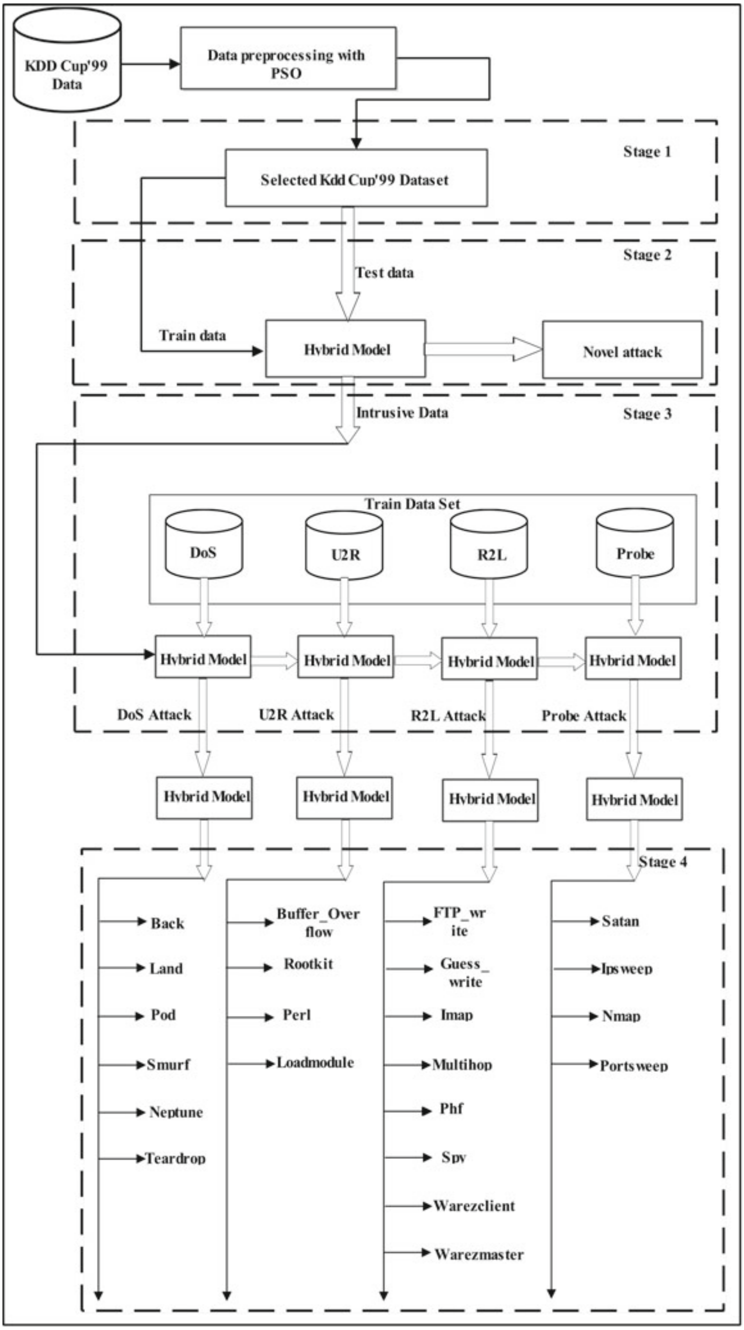


Fig. 2 Proposed multilevel hybrid IDS model

applied on these unclassified datasets for using the anomaly detection approach. IBK classifier takes very less time for processing due to the small size of datasets.

In stage 3, the unclassified datasets are separate out the DoS, U2R, R2L, and probe attacks, which are difficult steps for IBK classifier. Thus, this hybrid model coined the hybrid classifier: Both supervised learning (random tree) and unsupervised learning (k-nearest neighbor's algorithm) are involved. Though random tree was initially implemented in stage 1, unfortunately classification of attack's result was away from the satisfactory. Therefore, we have used the k-nearest neighbor algorithm, which classified the attacks separately and given better performance which is shown in Table 7.

The final fourth stage further classifies all four attack class into particular attack based on giving training dataset. The k-nearest neighbor algorithm determines which kind of attacks offensive the network. Thus, this classification more effective for classification of all sub-categories attacks and administrator easily understand all details of the attacks.

The algorithm of the proposed model is given below:

1. Pre-process KDD dataset with PSO algorithm on attribute set A= $\{a_1, a_2, a_3......a_{41}\}$.It reduces dataset attribute size to D =$\{a_1, a_2, a3......a_{1n}\}$, where $1 \leq n < 41$.
2. Divided D dataset into two parts: training dataset (D_{train}) and test dataset (D_{test}).
3. Training the hybrid model with D_{train} gives trained hybrid model, denoting it as $H_{trained.}$
4. Dataset D_{test} is provided as input to $H_{trained}$ that classifies D_{test} into two classes: known attack dataset $Attack_{known}$ and unclassified dataset $Attack_{unknown}$.
5. Create four replicas of hybrid model say H_1, H_2, H_3, H_4; and prepare four datasets of type DoS, U2R, R2L, Probe.
6. Train H_1 H_2, H_3, and H4 with DoS, U2R, R2L, and Probe dataset, respectively.
7. $Attack_{known}$ is passed through H_1 H_2, H_3, and H4 sequentially. H_1 filters DoS data from $Attack_{known}$ and pass unfiltered dataset to $H_{2.}$
8. Repeat step 7 for each hybrid model H_2, H_3, and $H_{4.}$ In each iteration, the dataset is filtered into their respective categories, say, $D_{Dos,}$ $D_{U2R,}$ D_{R2L}, and $D_{Probe,}$
9. Create six replicas of hybrid model for DoS attack categorization (H_{D1}, H_{D2},…,H_{D6}), eight replicas for R2L categorization (H_{U1}, H_{U2},..., H_{U8}), four replicas for U2R categorization (H_{R1}, H_{R2},..., H_{R4}), and four replicas for Probe categorizations (H_{P1}, H_{P2},..., H_{P4}),

10. Prepare six datasets for DoS subcategories (back , land, pod, smurf, neptune, teardrop), denoting as $D_{1,}$ $D_2,\ldots,D_6$)

    For i=1 to 6
     Train H_{Di} by $D_{i.}$
    End
    For i=1 to 5
     Pass D_{Dos} dataset to $H_{Di.}$ //it filters data from D_{Dos} to D_i
     Pass unfiltered data to $H_{Di+1.}$
    End
    Filter D_6 from D_{Dos} by H_{D6}

11. Prepare 8 datasets $R_{1,}$ $R_2,\ldots,R_8$ for R2L subcategories (ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster)

    For i=1 to 8
     Train H_{Ri} by $R_{i.}$
    End
    For i=1 to 7
     Pass D_{R2L} dataset to $H_{Ri.}$ //it filters data from D_{R2L} to R_i
     Pass unfiltered data to $H_{Ri+1.}$
    End
    Filter R_8 from $D_{R2L.}$ by H_{R8}.

12. Prepare 4 datasets $U_{1,}$ $U_2,\ldots,U_4$ for U2R subcategories (buffer_overflow, rootkit, loadmodule , perl)

    For i=1 to 4
     Train H_{Ui} by $U_{i.}$
    End
    For for i=1 to 3
     Pass D_{U2R} data set to $H_{Ui.}$ // it filters the data from D_{U2R} to U_i and
     Pass unfiltered data to $H_{Ui+1.}$
    End
    Filter R_4 from D_{U2R} by H_{U4}

13. Prepare 4 datasets for Probe subcategories (satan, ipsweep,nmap, portweep, says $P_{1,}$ $P_{2....}$,P_4)

    For i=1 to 4
     Train H_{Pi} by $P_{i.}$
    End
    For
     Pass D_{Probe} dataset to $H_{Pi.}$ // it filters the data from D_{Probe} to P_i
     Pass unfiltered data to $H_{Pi+1.}$
    End
    Filter P_4 from D_{Probe} by H_{P4}.

5 Experimental Results and Discussion

This section discusses the presentation and effectiveness of our proposed multilevel classifier hybrid intrusion detection model. Afterward, we have analyzed the accuracy of our proposed model for the KDD'99 dataset. Though Baghari et al. [33] have discussed various limitations of the KDD cup'99 dataset, yet it is still standard dataset and extensively applied for network intrusion detection. We have taken 10% dataset, which is classified into training and testing datasets. In our experiment, the ten percentage KDD dataset has been extracted that contains 390,459 instances after removing irrelevant instances and 15 features which have been categorized into four essential attack classes: DoS, probe, U2R, and R2L. All experiments have been carried out on the extracted dataset in order to analyze the performance of the proposed model.

The configuration of the system used for experiments is Microsoft Windows 8, 64 bit operating systems on Intel Core i7 processor with 2.40 GHz and 4.00 GB RAM. Table 2 shows the experimental training and testing dataset.

We have used the confusion matrix, also called inaccuracy matrix, which is a statistical classification measure that is used for machine learning algorithm problems. It is represented by a specific table layout as shown in Table 3. In matrix, columns represent the predicted class instances, and the rows represent the instances of actual classes. It contains true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values of instances, where TP and TN refer to the combination of correctly classified instances; FP and FN refer to the combination of incorrectly classified instances [36].

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TN} + \text{TP} + \text{FN} + \text{FP});$$

Table 2 Size of experimental dataset

Attack	Size of dataset (training)	Size of dataset (testing)
DoS	181,472	35,743
U2R	196	126
R2L	1178	856
Probe	23,732	15,789
Normal	82,947	35,496
Total	289,525	88,010

Table 3 Confusion matrix

Actual class	Predicted class	
	C	NC
C	TN	FP
NC	FN	TP

$$\text{Detection Rate(DR)} = \text{TP}/(\text{TP} + \text{FN});$$
$$\text{False Positive Rate(FPR)} = \text{FP}/(\text{FP} + \text{TN});$$

Representation of class instances are as follows;

$$\text{NC} = \text{Normal Class, C} = \text{Abnormal Class,}$$
$$\text{TN} = \text{True Negative, TP} = \text{True Positive}$$
$$\text{FN} = \text{False Negative, FP} = \text{False Positive}$$

In the proposed model, we have first reduced the dimensionality of KDD cup’ 99 dataset using the particle swarm optimization as this dataset which is huge and also contains many irrelevant attributes. Table 4 shows the impact of dimensionality reduction in training time and accuracy. As evident from this table, the dimensionality reduction improves the training time significantly with slightly variation in accuracy. In Table 5, we have shown a comparative performance of the proposed model, IBK, random tree classifier, and basis classifier such as MLP and RBF computed the detection rate and false positive rate. As evident from this table, the training time of the proposed model is much less than that of the basis classifier; it is due to the sequential approach used in the basis classifier. The detection rate of the proposed model is 98.7%, which is better than all other basis classifiers. The overall detection rate, false positive rate, and accuracy of the all five type’s attacks are shown in Table 6. In the particular, the detection rate of DoS attack is 97.52%, and the accuracy is 98.12% that is greater than other attacks. The false positive rate of DoS is 1.5%, which is slightly more as compared to the U2R due to misclassification of DoS attack connections. In this experimental work, the DoS and U2R attacks have much 97.52%, 96.73% higher detection rate and 1.05%, 1.02% less false positive rate as well as better overall detection rate and false positive rate. It may be concluded that the proposed model is a reliable model for intrusion detection.

Table 4 Dimensionality reduction analysis

Approach	Training time (s)	Accuracy (%)
With dimensionality reduction using PSO	5.34	97.81
Without dimensionality reduction	6.30	96.12

Table 5 Comparative performance of proposed ids model with various classifiers

Classifiers	Detection rate (%)	False positive rate (%)	Training time (s)
MLP	97.13	0.72	1898.87
RBF	96.42	1.41	147.89
Random tree	97.84	0.85	32.90
IBK	97.51	0.87	26.91
Proposed hybrid IDS model	98.73	0.53	32.90

Table 6 Performance of proposed IDS model

Attacks	Detection rate (%)	False positive rate (%)	Accuracy (%)
DoS	97.52	1.05	98.12
U2R	96.73	1.02	96.25
R2L	95.34	1.28	97.65
Probe	96.21	2.54	94.76
Normal	99.56	0.50	99.67

We have evaluated the performance of the proposed method for the sub-categories of DoS, U2R, R2L, and probe attacks as shown in Table 7. As evident from this table, the proposed classifier model performs better in all sub-categories attacks. Further, the overall detection rate for all sub-categories attacks have been compared with that of the existing bagged boosted C5 trees [37] and Kernel Miner [38] hybrid intrusion

Table 7 Performance of proposed multilevel hybrid classifier ids model

		Sub-categories attacks	Detection rate (%)	False positive rate (%)	Accuracy (%)
Attacks	DoS	Back	97.81	0.51	98.41
		Land	98.12	1.42	97.91
		Pod	98.71	0.73	98.89
		Neptune	96.85	2.52	97.12
		Smurf	95.89	1.52	96.54
		Teardrop	97.79	0.83	97.97
	U2R	Buffer overflow	96.83	0.54	97.98
		Loadmodule	95.73	1.03	95.97
		Perl	96.29	0.52	96.59
		Rootkit	95.50	1.03	97.39
	R2L	Ftp_write	95.20	1.66	96.36
		Guess_pwd	96.35	1.50	96.93
		Imap	93.40	1.16	94.51
		Multihop	95.20	2.50	92.34
		Phf	96.50	1.00	94.78
		Spy	97.75	0.56	95.73
		Warezclient	95.12	2.54	93.85
		Warezmaster	96.32	2.54	95.44
	Probe	Ipsweep	95.51	2.30	94.66
		Nmap	97.45	2.28	96.37
		Portsweep	94.62	1.23	96.19
		Satan	96.81	1.19	95.77

Table 8 Comparative detection rate of proposed ids model and existing model

Attacks	Bagged boosted C5 trees model	Kernel miner model	Proposed model
DoS	95.10	93.47	97.52
U2R	92.78	94.84	96.73
R2L	92.67	89.78	95.34
Probe	89.16	87.35	94.21
Normal	99.48	99.34	99.42

Table 9 Comparative performance of proposed ids model and existing approach

Model	Accuracy (%)	False positive rate (%)	Detection rate (%)
Proposed multilevel model	98.55	1.23	97.67
CSI-KNN-based model	96.53	2.81	95.48
J48-BN-based model	95.12	1.79	96.12
Two-level hybrid approach	97.12	1.01	96.88

detection approaches as shown in Table 8. As evident from this table, the proposed model provides the detection rates 97.52% in DoS, 96.73% in U2R, 95.34% in R2L, and 94.21% in probe attacks, which are much better than that of the C5tree and kernel miner approaches. The details of overall detection accuracy, detection rate, and false positive rate have been given in Table 9. Here, also, the proposed model provides better performance for all parameters, i.e., overall detection accuracy, detection rate, and false positive rate than that of the CSI-KNN [39], J48-BN [40] and two levels of hybrid approaches [41–44]. It may be concluded that the sequential approach gives better performance than an ensemble approach using the individual approaches. Thus, our proposed model shows promising performance for detecting the known attacks.

6 Conclusions

In this research work, we have investigated new multilevel hybrid classifier model for detection of intrusive activities and examined its performance on standard and irregular of KDD'99 dataset. The proposed model is an innovative combination of the classifiers that can be trained on a dataset in parallel, thus saves the training time and makes the system processing faster. The experimental results on the KDD cup'99 dataset have proved that the proposed model provides high detection rate of 97.67%, detection accuracy of 97.67%, and 1.23% false positive rate, which are better than the CSI-KNN and J48BN hybrid intrusion detection techniques. It has provided the detection rate for DoS attack as 97.52%, U2R attack as 96.73%, R2L attack as 95.34%, and probe attack as 94.21, which are better than the bagged boosted C5 trees and Kernel miner intrusion detection systems. Although the proposed multilevel classifier hybrid intrusion detection system is reliable model, it, however, requires a large room to improve the detection rate for the unknown attacks.

In future work, we plan to focus on detection of novel attacks with hybrid computational intelligence.

References

1. Desale, K. S., Kumathekar, C. N., & Chavan, A. P. (2015). Efficient intrusion detection system using stream data mining classification technique. In *International Conference on Computing Communication Control and Automation (ICCUBEA)* (pp. 469–473). IEEE.
2. Mohammad, M. N., Sulaiman, N., Muhsin, O. A. (2011). A novel intrusion detection system by using intelligent data mining in weka environment. *Procedia Computer Science, 3*, 1237–1242.
3. Murtaza, S. S., Khreich, W., Hamou-Lhadj, A., & Couture, M. (2013). A host-based anomaly detection approach by representing system calls as states of kernel modules. In *24th International Symposium on Software Reliability Engineering* (pp. 431–440).
4. Govindarajan, M., & Chandrasekaran, R. M. (2011). Intrusion detection using neural based hybrid classification methods. *Computer Networks, 55*(8), 1662–1671.
5. Cannady, J., & Harrell, J. (1996). A comparative analysis of current intrusion detection technologies. In *Proceedings of the Fourth Technology for Information Security Conference* (p. 96).
6. Ning, P., & Jajodia, S. (2003). Intrusion detection techniques. The Internet Encyclopedia.
7. Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16–24.
8. Dorj, E., & Altangerel, E. (2013). Anomaly detection approach using hidden markov model. In *8th International Forum on Strategic Technology (IFOST)* (Vol. 2, pp. 141–144), IEEE.
9. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials, 16*(1), 303–336.
10. Jabez, J., & Muthukumar, B. (2015). Intrusion detection system (IDS): Anomaly detection using outlier detection approach. *Procedia Computer Science, 48,* 338–346.
11. Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H., & Zhou, S. (2002). Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 265–274).
12. Elekar, K., Waghmare, M. M., & Priyadarshi, A. (2015). Use of rule base data mining algorithm for intrusion detection. In *International Conference on Pervasive Computing*. IEEE.
13. Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., & Kannan, A. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. *EURASIP Journal on Wireless Communications and Networking, 1,* 271.
14. Elekar, K. S. (2015). Combination of data mining techniques for intrusion detection system. In *International Conference on Computer, Communication and Control (IC4)* (pp. 1–5). IEEE.
15. Zhang, J., & Zulkernine, M. (2006). A hybrid network intrusion detection technique using random forests. In *First International Conference on Availability, Reliability and Security.* IEEE.
16. Dhakar, M., & Tiwari, A. (2014). A novel data mining based hybrid intrusion detection framework. *Journal of Information and Computing Science, 9*(1), 37–48.
17. Golmah, V. (2014). An efficient hybrid intrusion detection system based on C5.0 and SVM. *International Journal of Database Theory and Application, 7*(2), 59–70.
18. Pan, Shengyi, Morris, Thomas, & Adhikari, Uttam. (2015). Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid, 6*(6), 3104–3113.

19. Li, F. (2010). Hybrid neural network intrusion detection system using genetic algorithm. In *Multimedia Technology International Conference* (pp. 1–4).
20. Malik, A. J., Khan, F. A. (2013). A hybrid technique using multi-objective particle swarm optimization and random forests for PROBE attacks detection in a network. In *International Conference on Systems, Man, and Cybernetics* (pp. 2473–2478). IEEE.
21. Panda, M., Abraham, A., & Patra, M. R. (2012). A hybrid intelligent approach for network intrusion detection. *Procedia Engineering, 30*, 1–9.
22. Powers, Simon T., & He, Jun. (2008). A hybrid artificial immune system and self organising map for network intrusion detection. *Information Sciences, 178*(15), 3024–3042.
23. Nalini, N., & Rao, G. R. (2006). Network intrusion detection via a hybrid of genetic algorithms and principal component analysis. In *International Conference on Advanced Computing and Communications* (pp. 173–178).
24. Zhou, Y. P. (2009). Hybrid model based on artificial immune system and PCA neural networks for intrusion detection. In *Information Processing, Asia-Pacific Conference* (Vol. 1, pp. 21–24).
25. Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics,. Computational Cybernetics and Simulation* (Vol. 5, pp. 4104–4108).
26. Wang, Z., Sun, X., & Zhang, D. (2006). Classification rule mining based on particle swarm optimization. In *International Conference on Rough Sets and Knowledge Technology* (pp. 436–441). Berlin: Springer.
27. Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics, 43*(6), 1656–1671.
28. Zheng, H., Hou, M., & Wang, Y. (2011). An efficient hybrid clustering-PSO algorithm for anomaly intrusion detection. *Journal of Software, 6*(12), 2350–2360.
29. Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing, 38*, 360–372.
30. Htun, P. T., & Khaing, K. T. (2013). Detection model for daniel-of-service attacks using random forest and k-nearest neighbors. *International Journal of Advanced Research in Computer Engineering & Technology, 2*.
31. Govindarajan, M., & Chandrasekaran, R. M. (2009). Intrusion detection using k-nearest neighbor. In *International Conference on Advanced Computing* (pp. 13–20).
32. Sewaiwar, P., & Verma, K. K. (2015). Comparative study of various decision tree classification algorithm using WEKA.
33. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6).
34. Lin, S. W., Ying, K. C., Lee, C. Y., & Lee, Z. J. (2012). An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing, 12*(10), 3285–3290.
35. Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications, 34*(4), 1184–1199.
36. Gautam, S. K., & Om, H. (2017). Comparative analysis of classification techniques in network based intrusion detection systems. In *Proceedings of the First International Conference on Intelligent Computing and Communication* (pp. 591–601). Singapore: Springer.
37. Pfahringer, B. (2000). Winning the KDD99 classification cup: Bagged boosting. *ACM SIGKDD Explorations Newsletter, 1*(2), 65–66.
38. Levin, Itzhak. (2000). KDD-99 classifier learning contest: LLSoft's results overview. *SIGKDD Explorations, 1*(2), 67–75.
39. Kuang, L., & Zulkernine, M. (2008). An anomaly intrusion detection method using the CSI-KNN algorithm. In *Proceedings of the 2008 ACM symposium on Applied Computing* (pp. 921–926).

40. Khor, K.-C., Ting, C.-Y., & Phon-Amnuaisuk, S. (2012). A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. *Applied Intelligence, 36*(2), 320–329.
41. Guo, C., Ping, Y., Liu, N., & Luo, S. S. (2016). A two-level hybrid approach for intrusion detection. *Neurocomputing, 214,* 391–400.
42. Kaliannan, J., Baskaran, A., Dey, N., & Ashour, A. S. (2016). Ant colony optimization algorithm based PID controller for LFC of single area power system with non-linearity and boiler dynamics. *The World Journal of Modelling and Simulation, 12*(1), 3–14.
43. Kaliannan, J., Baskaran, A., & Dey, N. (2015). Automatic generation control of thermal-thermal-hydro power systems with PID controller using ant colony optimization. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 6*(2), 18–34.
44. Jagatheesan, K., Anand, B., Samanta, S., Dey, N., Ashour, A. S., & Balas, V. E. (2017). Particle swarm optimisation-based parameters optimisation of PID controller for load frequency control of multi-area reheat thermal power systems. *International Journal of Advanced Intelligence Paradigms, 9*(5–6), 464–489.

Maintaining Manpower in Technical College Using Fusion of Quadratic Programming and Fuzzy Logic

Manoj Kumar Mandal, B. K. Mahatha, Arun Prasad Burnwal, Santosh Kumar Das, and Joydev Ghosh

1 Introduction

In last few decades, the number of colleges and universities increases rapidly due to highly increasing ratio of the students in our country. Each college and university has its own rules and regulations. Some of the rules based on higher authority of the Government. The most of the colleges follows its university rules in which the college is affiliated. Institute contains several departments and each department consists of several students. But compare to number of students, number of teachers is less, not only compare to number of students, but compare to courses and academic works also. Along with engineering, teacher and general department's teacher is also less. The working staffs in the institute are also less compare to work load in the institute. So, managing manpower as teachers in several department and staffs in

M. K. Mandal (✉)
Department of Mathematics, Jharkhand Rai University, Ranchi 835222, India
e-mail: mkmandal346@gmail.com

B. K. Mahatha
Amity School of Engineering and Technology, Amity University, Ranchi, Jharkhand 834001, India
e-mail: bhupeshmahatha@gmail.com

A. P. Burnwal
Department of Mathematics, GGSESTC, Bokaro, Jharkhand 827013, India
e-mail: apburnwal@yahoo.com

S. K. Das
School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Institute Park, Pallur Hills, Berhampur, Odisha 761008, India
e-mail: sunsantosh2014@gmail.com

J. Ghosh
School of Computer Science and Robotics, National Research Tomsk Polytechnic University (TPU), Tomsk, Russia
e-mail: joydev.ghosh.ece@gmail.com

S. K. Das et al. (eds.), *Nature-Inspired Computing for Smart Application Design*, Springer Tracts in Nature-Inspired Computing,
https://doi.org/10.1007/978-981-33-6195-9_13

several departments along with offices is difficult task. This issue increases rapidly in day by day. It effects the quality of education in the institute and future of the student.

Therefore, the proposed method is used to solve the above-mentioned issues efficiently with the help of mathematical optimization. The proposed method is the extension of the work mentioned in [1]. The mathematic optimization used in the proposed method is the fusion of two techniques such as quadratic programming and fuzzy logic. Quadratic programming is a mathematical optimization model where objective function is a format of nonlinear of degree two and constraints may be linear or nonlinear depends on the situation and problem [2]. Fuzzy logic is a soft computing technique; sometimes, it is also called as one of the meta-heuristic techniques which is used to reduce uncertainty of the problem by reducing imprecise information efficiently [3–5]. The combination of both efficiently manages the manpower of the technical college.

The roadmap of the paper is as follows. Unit 2 described some information about existing works. Unit 3 illustrated the basic preliminaries information related to the proposed method. Unit 4 describes the details of the proposed method. Unit 5 describes the simulation analysis. And Unit 6 concludes the paper.

2 Literature Review

The proposed method is based on optimization technique which is a fusion of quadratic programming and fuzzy logic. Quadratic programming is an extended form of linear programming in the form of nonlinear formulation and fuzzy logic is a component of soft computing for reducing uncertainty of any imprecise information. In this section, several literatures are described based on optimization related to linear programming, quadratic programming, and fuzzy logic, and some other mathematical formulations. Short descriptions are as follows. Aboelmagd [6] designed an application for construction sites. Basically, it is used to take decision for managing office equipment based on requirement in Arabian construction. The basic key element of this proposal is linear programming which is used to optimizing purpose. This linear programming is used to manage competitive strategy of the building structure. It is also used to predict influence of the decision-making system. Finally, it gives shortest computation structure which is affordable and less time consuming. Rodias [7] designed a combined technique for fertilizer application using linear programming. The proposed method used linear programming for optimizing several strategies of fertilizer such as harvesting handling operation, organic fertilization, mineral, and other several usage. These all strategies are mapped into linear programming for solving different issues of the system. Ji et al. [8] designed a multi-objective optimization technique for linear programming using a game method for supply chain management. The proposed method is the fusion of two method; first is duality method for mathematical approach and second is Karush Kuhn tucker method. In this proposed method, the author is tried to pareto equilibrium problem to achieve the purpose of the method. Finally, it achieves the purpose of the method by using multilinear technique. Nowadays, electricity is an important source of electricity energy and its demand has been increasing day by day. To forecast the future consumption of electricity demand, a time series-based ARIMA model presented in [9]. In [10], SB has considered for the student academic performance prediction. In this paper, student academic performance prediction evaluated with the help of different machine learning classification models. Further accuracy has been improved by using ensembling methods. Outliers are points that do not follow the patterns form the rest of the data. The clustering-based outlier detection method has been proposed [11]; in this method, three nearest $K - 1, K, K + 1$ clusters computed using the K-means clustering algorithm. Long distance point form all the clusters considers as an outlier point. Sales forecasting is an essential facet for the industries associated with sales, wholesale, manufacturing, etc. all around the globe. It is important with respect to resource allocation, revenue estimation, and market strategy planning. A two-level approach [12] performs better in comparison with other single-level model. Yang et al. [13] designed an intelligent system for transportation system in wireless network. This is based on an existing transportation system based on process structured system. Finally, it helps to enhance network capabilities and services of the network. It also helps to user function and usages in the network and network metrics properly to maintain the network. Loganathan and Subbiah [14] designed an energy-based communication system for device-to-device communication in the network. It is based on multi-criteria decision-making system where multiple criteria are involved for integrating the network metrics efficiently. Finally, it helps to enhance the network lifetime and helps in communication system. Chandrakar [15] designed an authentication system for the users in wireless network. This is basically based on healthcare system and used for medical purpose. This proposal is used for sensing patient body information and sends to the doctor for treatment and diagnosis purpose. It also helps in user authentication, privacy, and data security purpose, so that efficient result comes from the diagnosis system. Gharanjik et al. [16] designed max-min technique for application of information and signal processing. The proposed method is modelled by quadratic optimization technique. The proposed method usage Gram-Schmidt technique for handling approximation method to achieve the discrete requirement. Finally, it achieves the goal of the proposed method and reduces the computational cost of the method with respect to real life scenarios. Hempel et al. [17] proposed a method for hybrid control system using quadratic optimization technique. In this method, the used quadratic programming is mixed integer programming. In this method, the main issue is to manage piecewise affine system. This issue is efficiently controls and manages with the help of mixed integer quadratic optimization technique. Tripathi and Das [18] proposed a vague set-based routing technique for ad hoc network. Vague set is one of the extended versions of the fuzzy set where fuzzy set deals with the degree of membership value and vague set deals with degree of membership and degree of non-membership value. Later, this work is extended in [19] for evaluating more network metrics. The combination of both helps to recognize the imprecise network parameters efficiently, especially energy and distance both are the crucial parameters of the network. Finally, it helps to enhance the network metrics and network lifetime. Das et al. [20] designed a routing method for multiple destination ad hoc network where source is one, but destination node is situated in various form based on different energy system. In this work, fuzzy logic is used in the form of linguistic variable that divides into some membership function based on randomization which help to categorize the feasible as well as optimal route and reduce the uncertainty of the network. Tripathi and Das [21] designed a robustness method for routing in ad hoc network. Here, network is based on transparent system and heterogeneous in nature. Basic key elements of this network are fusion of linear programming and game theory. It consists of two objective functions for two players where both players are competitive to each other. Finally, it helps to reduce conflicting nature and pave the complexity of the network and find optimal route. Das et al. [22] designed an efficient routing method for ad hoc network where the nature of the ad hoc is wireless. The complete work is based on two stages; first stage is graph initiation phase for nodes with source and destination nodes and second stage contains route decision system named as reward calculation by the help of input parameters. Finally, it helps to reduce the complexity as well as uncertainty of the network. Mishra et al. [23] proposed a model for grinding process based on fuzzy logic. The proposed method is based on an intelligent operation which is named as compensatory operator. It is based on weighted factoring technique. The proposed method optimized several parameters such as speed, density, sectional area which is cross in this model all these parameters are imprecise which are compact and model by membership function of the fuzzy logic. Murmu et al. [24] designed a system for predicting and analysis for surface roughness. The proposed method is based on hard face component. In this model, fuzzy logic is used to optimize several factors of the machines efficiently and manage various parts of the machine. It uses hard surfacing technique that uses fuzzy logic, combine system enhance the service mechanism of the machine. Kumari and Burnwal [25] designed an interactive model for inventory control system. The proposed method is based on various mathematical operator for analysing different scenarios of the model. It uses fuzzy logic system for enhancing the model by reducing imprecise parameters of the network. Finally, it solves several objectives of the inventory by combining multi-objective optimization and fuzzy logic of the system.

3 Preliminaries

In this section, basic preliminaries are described that helps to understand the proposed method efficiently. Short descriptions are as follows.

3.1 Quadratic Programming

Quadratic programming is a part of nonlinear programming which relates with objective function and constraints nonlinearly. This is based on second-order polynomial technique. In this mathematical modelling, the objective function is always nonlinear in nature but constraints are linear or nonlinear based on the situation.

3.2 Fuzzy Logic

Fuzzy logic is a multi-valued logic which is based on the relation between partial truth and partial false depends on degree of truth value. Degree of truth value is evaluated based on relation of universe of discourse and degree of membership function. Fuzzy logic deals with linguistic variables for reducing uncertainty of information and estimates imprecise parameters of the system.

4 Proposed Method

In this section, the proposed method is illustrated efficiently in term of mathematical modelling. The proposed method is divided into two phases such as preliminary assumption and mathematical modelling. Preliminary assumption phase is used for preliminary information which gives basic information about variables and notations which are used in the mathematical optimization. Mathematical modelling is used to map these variables in the form of optimization model that produce feasible as well as optimal solutions based on the goal of the paper.

4.1 Preliminary Assumption

Let D is the set departments of an academic institute that contains several faculty members along with several staffs. Equations 1–3 show set of different departments, set of different faculties, and set of different staffs. In this method, faculty members are divided into two types such as engineering department's teacher and general department's teacher. Tables 1, 2 and 3 show list of assumptions for engineering department, general department, and staffs, and list of manpower required in the engineering department, general department, and staff selection shown in Tables 4, 5 and 6.

$$D = \{d_1, d_2, d_3, \ldots, d_m\} \tag{1}$$

Table 1 List of engineering department

Department	Faculty	Description
d_1	f_1	Chemical engineering
d_2	f_2	Civil engineering
d_3	f_3	Computer science and engineering
d_4	f_4	Electrical engineering
d_5	f_5	Electronics and communication engineering
d_6	f_6	Environmental engineering
d_7	f_7	Mechanical engineering
d_8	f_8	Mineral and metallurgical engineering
d_9	f_9	Mining engineering
d_{10}	f_{10}	Mining machinery engineering

Table 2 List of general department

Department	Faculty	Description
d_{11}	f_{11}	Chemistry
d_{12}	f_{12}	Physics
d_{13}	f_{13}	Mathematics and computing
d_{14}	f_{14}	Environmental science
d_{15}	f_{15}	Yoga

Table 3 List of staff

Notation	Description
s_1	Staff 1
s_2	Staff 2
s_3	Staff 3
s_4	Staff 4
s_5	Staff 5

$$F = \{f_1, f_2, f_3, \ldots, f_n\} \tag{2}$$

$$S = \{s_1, s_2, s_3, \ldots, s_k\} \tag{3}$$

where $F > D$, $S > D$, and D, F, S are may not be equal.

Table 4 List of manpower required in engineering department

Faculty	Permanent faculty	Contractual faculty	Adjunct faculty
f_1	p_1	c_1	a_1
f_2	p_2	c_2	a_2
f_3	p_3	c_3	a_3
f_4	p_4	c_4	a_4
f_5	p_5	c_5	a_5
f_6	p_6	c_6	a_6
f_7	p_7	c_7	a_7
f_8	p_8	c_8	a_8
f_9	p_9	c_9	a_9
f_{10}	p_{10}	c_{10}	a_{10}

Table 5 List of manpower required in general department

Faculty	Permanent faculty	Contractual faculty	Adjunct faculty
f_{11}	p_{11}	c_{11}	a_{11}
f_{12}	p_{12}	c_{12}	a_{12}
f_{13}	p_{13}	c_{13}	a_{13}
f_{14}	p_{14}	c_{14}	a_{14}
f_{15}	p_{15}	c_{15}	a_{15}

Table 6 List of manpower required for staff

Staff	Permanent staff	Contractual staff	Daily basis staff
s_1	p_{16}	c_{16}	a_{16}
s_2	p_{17}	c_{17}	a_{17}
s_3	p_{18}	c_{18}	a_{18}
s_4	p_{19}	c_{19}	a_{19}
s_5	p_{20}	c_{20}	a_{20}

4.2 Mathematical Modelling

In this subsection, mathematical modelling is illustrated which is based on three type models as (i) faculty member of engineering department, (ii) faculty member of general department, and (iii) staff member. Faculty members are divided into three types, such as permanent, contractual, and adjunct, and staff members are divided into three types, such as permanent, contractual, and daily basis staff member. The decision variables x_1, x_2, and x_3 are assumed for three different faculty members and in the case of staff three type of staffs (e.g. permanent, contractual, and daily basis).

In this paper, three optimization models are designed in the form of quadratic for better optimality as shown in Eqs. 4–6.

Minimize: $Z_1 = x_1^2 + x_2^2 + x_3^2$

Subject to constraints:

$$\begin{aligned}
&p_1x_1 + c_1x_2 + a_1x_3 \geq n_1\\
&p_2x_1 + c_2x_2 + a_2x_3 \geq n_1\\
&p_3x_1 + c_3x_2 + a_3x_3 \geq n_1\\
&p_4x_1 + c_4x_2 + a_4x_3 \geq n_1\\
&p_5x_1 + c_5x_2 + a_5x_3 \geq n_1\\
&p_6x_1 + c_6x_2 + a_6x_3 \geq n_1\\
&p_7x_1 + c_7x_2 + a_7x_3 \geq n_1\\
&p_8x_1 + c_8x_2 + a_8x_3 \geq n_1\\
&p_9x_1 + c_9x_2 + a_9x_3 \geq n_1\\
&p_{10}x_1 + c_{10}x_2 + a_{10}x_3 \geq n_1\\
&x_1, x_2, x_3 \geq 0
\end{aligned} \tag{4}$$

Minimize: $Z_2 = x_1^2 + x_2^2 + x_3^2$

Subject to constraints:

$$\begin{aligned}
&p_{11}x_1 + c_{11}x_2 + a_{11}x_3 \geq n_1\\
&p_{12}x_1 + c_{12}x_2 + a_{12}x_3 \geq n_1\\
&p_{13}x_1 + c_{13}x_2 + a_{13}x_3 \geq n_1\\
&p_{14}x_1 + c_{14}x_2 + a_{14}x_3 \geq n_1\\
&p_{15}x_1 + c_{15}x_2 + a_{15}x_3 \geq n_1\\
&x_1, x_2, x_3 \geq 0
\end{aligned} \tag{5}$$

Minimize: $Z_3 = x_1^2 + x_2^2 + x_3^2$

Subject to constraints:

$$\begin{aligned}
&p_{16}x_1 + c_{16}x_2 + a_{16}x_3 \geq n_1\\
&p_{17}x_1 + c_{17}x_2 + a_{17}x_3 \geq n_1\\
&p_{18}x_1 + c_{18}x_2 + a_{18}x_3 \geq n_1\\
&p_{19}x_1 + c_{19}x_2 + a_{19}x_3 \geq n_1\\
&p_{20}x_1 + c_{20}x_2 + a_{20}x_3 \geq n_1\\
&x_1, x_2, x_3 \geq 0
\end{aligned} \tag{6}$$

In Eq. 4, Z_1 is the objective function for minimization of manpower in engineering department for faculties. In this model, x_1, x_2, and x_3 are the decision variables for permanent faculty, contractual faculty, and adjunct faculty in the form of

Table 7 Membership functions of the mathematical model

Type	Low	Medium	High
Permanent	(0–10)	(8–15)	(13–20)
Contractual	(0–7)	(5–11)	(10–15)
Adjunct or daily basis	(0–4)	(3–7)	(5–12)

quadratic formulation where p_1 to p_{10}, c_1 to c_{10}, and a_1 to a_{10} are faculty members for different department d_i where i varies between 1 and 10 which based on permanent, contractual, and adjunct faculty members. In Eq. 5, Z_2 is the objective function for minimization of manpower in general department for faculties. In this model, x_1, x_2, and x_3 are the decision variables for permanent faculty, contractual faculty, and adjunct faculty in the form of quadratic formulation, where p_{11} to p_{15}, c_{11} to c_{15}, and a_{11} to a_{15} are faculty members for different department d_i where i varies between 11 and 15 which based on permanent, contractual, and adjunct faculty members. In Eq. 6, Z_3 is the objective function for minimization of manpower for staff in several departments. In this model, x_1, x_2, and x_3 are the decision variables for several type staffs such as permanent staffs, contractual staffs, and daily basis staffs in the form of quadratic formulation where p_{16} to p_{20}, c_{16} to c_{20}, and a_{16} to a_{20} are staffs of different departments as permanent staffs, contractual staffs, and daily basis staffs. For reducing uncertainty of the information and increase the efficiency and robustness of the problem, fuzzy membership functions are designed for each mathematical model which is shown in Table 7.

In Table 7, permanent faculty members of engineering and general departments and permanent staff for all department are considered as 20, contractual faculty members for engineering and general departments and contractual staff for all department are considered as 15, adjunct faculty members for engineering and general departments and daily basis staff members of all department are considered as 12. These fuzzy membership functions are mapped into the above-mentioned mathematical models as shown in Eqs. 4–6. Membership function makes the constraint of all mathematical models as imprecise for varying the parameters efficiently and reduces the uncertainty of the information. The proposed method is evaluated and analysed and showed that maximum manpower is required in engineering department, then general department and last in staff of all department.

5 Simulation and Analysis

In this section, details of simulation and analysis are illustrated. The proposed method is simulated in LINGO optimization software-based nonlinear formulation of quadratic programming. The proposed method is simulated and verified in three optimization models based on three rounds for each linguistic variable. The parameters details are shown in Table 8.

Table 8 Simulation parameter details

Parameter	Description
Windows	Windows 10 pro
MS Office	2013
Optimization software	LINGO
Permanent faculty member	20
Contractual faculty member	15
Staff of all department	12
Optimization	Minimization
Input parameters	Three (permanent, contractual, adjunct or daily basis)
Output parameter	Minimum requirement
Constraints for first model	10
Constraints for second model	5
Constraints for third model	5
Nature of objectives	Nonlinear
Nature of constraints	Linear
Linguistic variables of engineering department	3
Linguistic variables of general department	3
Linguistic variables of staff	3
Name of the linguistic variables	Low, medium, high

In Table 8, simulation parameters are illustrated. In this illustration, Windows 10 pro operating system is used with MS Office 2013 along with LINGO mathematical optimization software. In this optimization, permanent faculty member is considered as 20 members, contractual faculty member is considered as 15 members, and staff of all department is considered as 12 members. The purpose of these three model is minimization that is used to minimize minimum requirement of manpower of engineering department, general department, and staff of all department. So, in this optimization, there are three inputs as three mathematical models for permanent faculty members, contractual faculty members, and adjunct faculty members or daily basis staff members. And output is minimum requirement. In first mathematical model, total constraints are used 10 with nonlinear quadratic objective function. In second mathematical model, total constraints are used 5 with nonlinear quadratic objective function. In third mathematical model, total constraints are used 5 with nonlinear quadratic objective function. The nature of each objective function is nonlinear but nature of each constraint of each model is linear. There are three types of linguistic variables which are used in each mathematical model named as low, medium, and high.

Figures 1, 2 and 3 show illustrations of minimum manpower requirements of engineering department, general department, and staff of all departments with respect of

```
Solution Report - MPT -Eng
  Global optimal solution found.
  Objective value:                              11.11111
  Infeasibilities:                              0.000000
  Total solver iterations:                             8
  Elapsed runtime seconds:                          0.10
  Model is convex quadratic

  Model Class:                                        QP

  Total variables:                     3
  Nonlinear variables:                 3
  Integer variables:                   0

  Total constraints:                  14
  Nonlinear constraints:               1

  Total nonzeros:                     36
  Nonlinear nonzeros:                  3


                              Variable           Value        Reduced Cost
                                    X1        2.222417        0.3884660E-03
                                    X2        1.110850       -0.5218349E-03
                                    X3        2.222158       -0.1284440E-03

                                   Row    Slack or Surplus      Dual Price
                                     1        11.11111        -1.000000
                                     2        21.11189        0.1530501E-07
                                     3        9.999344       -0.2028946E-07
                                     4        11.11014        0.9118590E-08
                                     5        7.777842        0.2659759E-07
                                     6      0.4050685E-06     -2.222223
                                     7        16.66634        0.1069643E-07
                                     8        13.33209        0.8199686E-08
                                     9        7.777126       -0.2956586E-07
                                    10        13.33307        0.3695167E-08
                                    11        9.999344         0.000000
                                    12        2.222417         0.000000
                                    13        1.110850         0.000000
                                    14        2.222158         0.000000
```

Fig. 1 Minimum requirement of manpower in engineering department in case of "low" linguistic

```
Solution Report - MPT -Gn
  Global optimal solution found.
  Objective value:                              7.142858
  Infeasibilities:                              0.000000
  Total solver iterations:                            14
  Elapsed runtime seconds:                          0.10
  Model is convex quadratic

  Model Class:                                        QP

  Total variables:                     3
  Nonlinear variables:                 3
  Integer variables:                   0

  Total constraints:                   9
  Nonlinear constraints:               1

  Total nonzeros:                     21
  Nonlinear nonzeros:                  3


                              Variable           Value        Reduced Cost
                                    X1        1.429100        0.2386626E-03
                                    X2        2.142328       -0.2399048E-03
                                    X3       0.7148151        0.2395938E-03

                                   Row    Slack or Surplus      Dual Price
                                     1        7.142858        -1.000000
                                     2        5.717989       -0.3260771E-08
                                     3      0.3302905E-06     -1.427953
                                     4      0.3175087E-02     -0.1999197E-03
                                     5      0.1587708E-02     -0.4190540E-03
                                     6        13.57249        0.5386633E-07
                                     7        1.429100         0.000000
                                     8        2.142328         0.000000
                                     9       0.7148151         0.000000
```

Fig. 2 Minimum requirement of manpower in general department in case of "low" linguistic

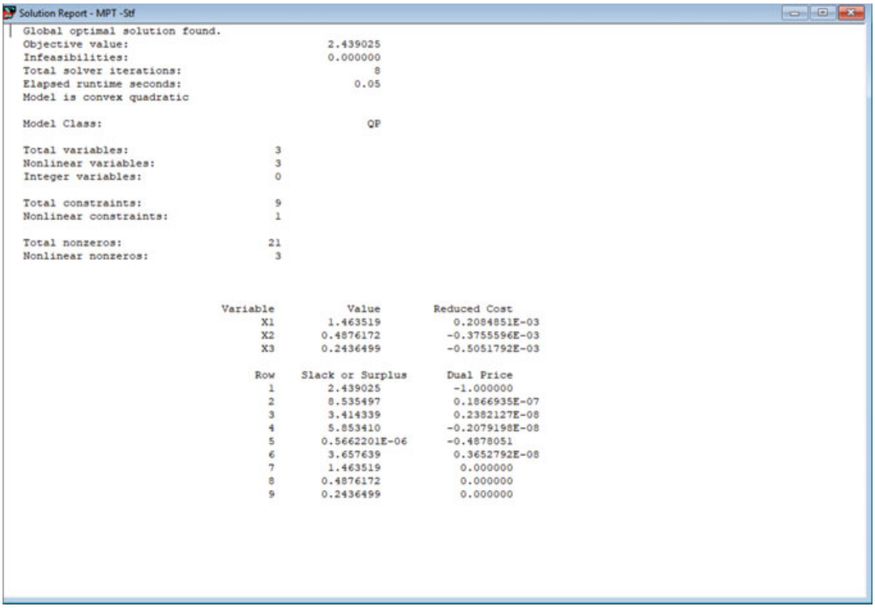


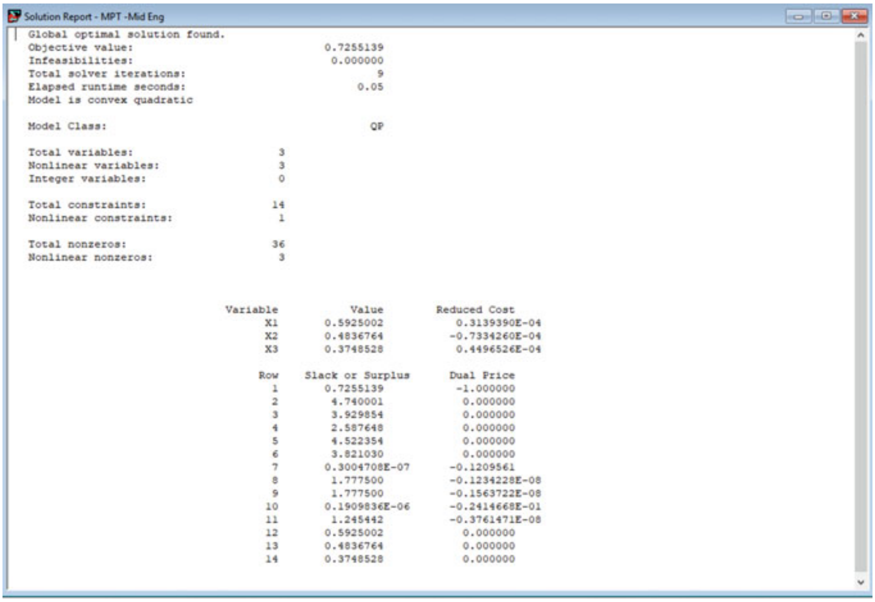
Fig. 3 Minimum requirement of staff in all departments in case of “low” linguistic

“low” linguistic variables where in engineering department, requirement is 11.11111, in general department, requirement is 7.142858, and in staff of all department is 2.439025.

Figures 4, 5 and 6 show illustrations of minimum manpower requirements of engineering department, general department and staff of all departments with respect of “medium” linguistic variables where in engineering department requirement is 0.7255139, in general department requirement is 0.7042254, and in staff of all department is 0.6615779.

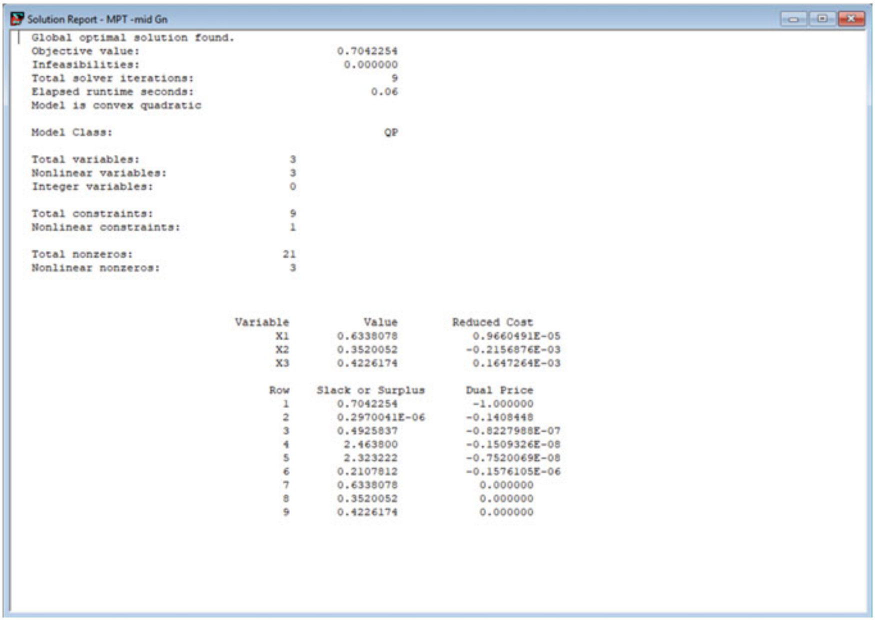
Figures 7, 8 and 9 show illustrations of minimum manpower requirements of engineering department, general department and staff of all departments with respect of “high” linguistic variables where in engineering department requirement is 0.2923977, in general department requirement is 0.2652520, and in staff of all department is 0.2579536.

Tables 9, 10 and 11 show dataset of faculty member of engineering department based on linguistic variables “low”, “medium”, and “high” with $n_1 = 10$ where values of low, medium, and high are generated randomly based on linguistic range of the linguistic variables and finally produce different optimal values based on linguistic variables. In faculty members of engineering department, based on “low” linguistic variable, optimal value is 11.11111 where values of decision variables are 2.222417, 1.110850, and 2.222158. Based on “medium” linguistic variable, optimal value is 0.7255139, where decision variables are 0.5925035, 0.4836758, and 0.3748483.



```
Solution Report - MPT -Mid Eng
Global optimal solution found.
Objective value:                          0.7255139
Infeasibilities:                          0.000000
Total solver iterations:                         9
Elapsed runtime seconds:                      0.05
Model is convex quadratic

Model Class:                                    QP

Total variables:                  3
Nonlinear variables:              3
Integer variables:                0

Total constraints:               14
Nonlinear constraints:            1

Total nonzeros:                  36
Nonlinear nonzeros:               3


                    Variable        Value        Reduced Cost
                          X1    0.5925002        0.3139390E-04
                          X2    0.4836764       -0.7334260E-04
                          X3    0.3748528        0.4496526E-04

                         Row   Slack or Surplus     Dual Price
                           1    0.7255139          -1.000000
                           2     4.740001           0.000000
                           3     3.929854           0.000000
                           4     2.587648           0.000000
                           5     4.522354           0.000000
                           6     3.821030           0.000000
                           7    0.3004708E-07      -0.1209561
                           8     1.777500          -0.1234228E-08
                           9     1.777500          -0.1563722E-08
                          10    0.1909836E-06      -0.2414668E-01
                          11     1.245442          -0.3761471E-08
                          12    0.5925002           0.000000
                          13    0.4836764           0.000000
                          14    0.3748528           0.000000
```

Fig. 4 Minimum requirement of manpower in engineering department in case of "medium" linguistic



```
Solution Report - MPT -mid Gn
Global optimal solution found.
Objective value:                          0.7042254
Infeasibilities:                          0.000000
Total solver iterations:                         9
Elapsed runtime seconds:                      0.06
Model is convex quadratic

Model Class:                                    QP

Total variables:                  3
Nonlinear variables:              3
Integer variables:                0

Total constraints:                9
Nonlinear constraints:            1

Total nonzeros:                  21
Nonlinear nonzeros:               3


                    Variable        Value        Reduced Cost
                          X1    0.6338078        0.9660491E-05
                          X2    0.3520052       -0.2156876E-03
                          X3    0.4226174        0.1647264E-03

                         Row   Slack or Surplus     Dual Price
                           1    0.7042254          -1.000000
                           2    0.2970041E-06      -0.1408448
                           3    0.4925837          -0.8227988E-07
                           4     2.463800          -0.1509326E-08
                           5     2.323222          -0.7520069E-08
                           6    0.2107812          -0.1576105E-06
                           7    0.6338078           0.000000
                           8    0.3520052           0.000000
                           9    0.4226174           0.000000
```

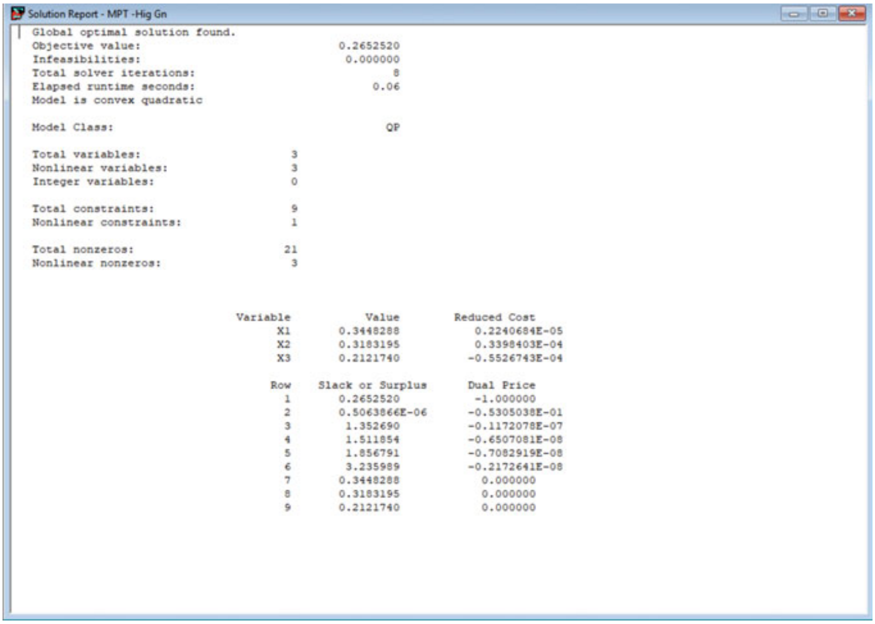
Fig. 5 Minimum requirement of manpower in general department in case of "medium" linguistic

```
Solution Report - MPT -Mid st
Global optimal solution found.
Objective value:                              0.6615779
Infeasibilities:                               0.000000
Total solver iterations:                              7
Elapsed runtime seconds:                           0.05
Model is convex quadratic

Model Class:                                         QP

Total variables:                        3
Nonlinear variables:                    3
Integer variables:                      0

Total constraints:                      9
Nonlinear constraints:                  1

Total nonzeros:                        21
Nonlinear nonzeros:                     3


                        Variable           Value        Reduced Cost
                              X1       0.5700183        0.8368475E-04
                              X2       0.4325502       -0.2602237E-04
                              X3       0.3867263       -0.9491690E-04

                             Row    Slack or Surplus      Dual Price
                               1       0.6615779          -1.000000
                               2        4.809409         0.7914774E-08
                               3        5.058668         0.3966694E-08
                               4       0.7056175E-06    -0.8142842E-01
                               5        3.577725        -0.5996828E-08
                               6       0.4325324E-05    -0.5088716E-01
                               7       0.5700183          0.000000
                               8       0.4325502          0.000000
                               9       0.3867263          0.000000
```

Fig. 6 Minimum requirement of staff in all departments in case of "medium" linguistic

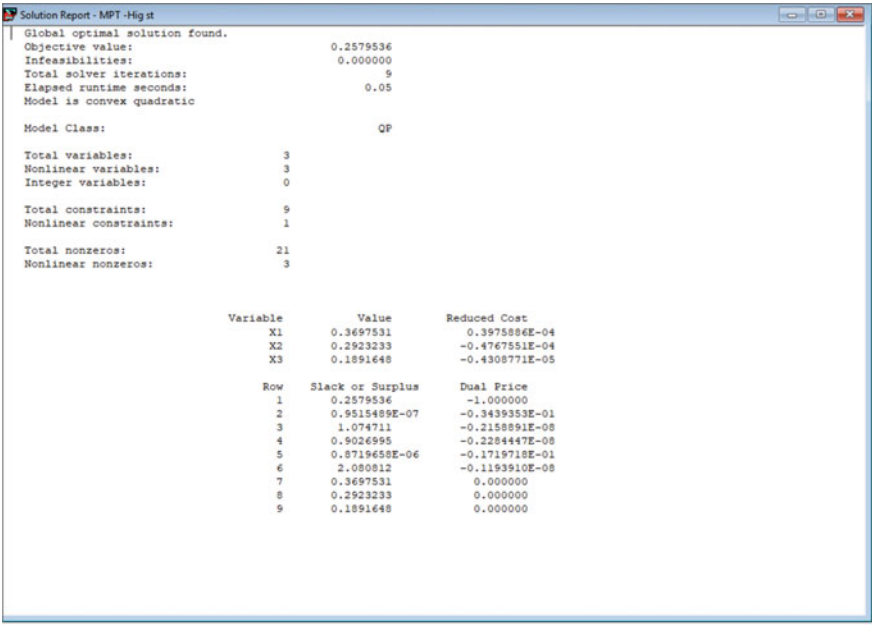
```
Solution Report - MPT -Hig Eng
Global optimal solution found.
Objective value:                              0.2923977
Infeasibilities:                               0.000000
Total solver iterations:                              7
Elapsed runtime seconds:                           0.15
Model is convex quadratic

Model Class:                                         QP

Total variables:                        3
Nonlinear variables:                    3
Integer variables:                      0

Total constraints:                     14
Nonlinear constraints:                  1

Total nonzeros:                        36
Nonlinear nonzeros:                     3


                        Variable           Value        Reduced Cost
                              X1       0.4093851        0.5658097E-04
                              X2       0.3215861       -0.1028029E-03
                              X3       0.1462323        0.6679616E-04

                             Row    Slack or Surplus      Dual Price
                               1       0.2923977          -1.000000
                               2       0.5888482E-06    -0.5847954E-01
                               3       0.7895952        -0.1767792E-08
                               4        3.070552        -0.1118038E-08
                               5        2.309971         0.2735425E-08
                               6        2.573124         0.2197274E-08
                               7        1.549688        -0.1741921E-08
                               8        1.432822        -0.1022221E-07
                               9        2.426946        -0.2235540E-08
                              10       0.3508984        -0.1802684E-08
                              11        3.245662         0.2591588E-08
                              12       0.4093851          0.000000
                              13       0.3215861          0.000000
                              14       0.1462323          0.000000
```

Fig. 7 Minimum requirement of manpower in engineering department in case of "high" linguistic



```
Solution Report - MPT -Hig Gn
Global optimal solution found.
Objective value:                              0.2652520
Infeasibilities:                              0.000000
Total solver iterations:                             8
Elapsed runtime seconds:                          0.06
Model is convex quadratic

Model Class:                                        QP

Total variables:                      3
Nonlinear variables:                  3
Integer variables:                    0

Total constraints:                    9
Nonlinear constraints:                1

Total nonzeros:                      21
Nonlinear nonzeros:                   3

                    Variable           Value        Reduced Cost
                          X1       0.3448288        0.2240684E-05
                          X2       0.3183195        0.3398403E-04
                          X3       0.2121740       -0.5526743E-04

                         Row    Slack or Surplus      Dual Price
                           1       0.2652520         -1.000000
                           2       0.5063866E-06     -0.5305038E-01
                           3        1.352690         -0.1172078E-07
                           4        1.511854         -0.6507081E-08
                           5        1.856791         -0.7082919E-08
                           6        3.235989         -0.2172641E-08
                           7       0.3448288          0.000000
                           8       0.3183195          0.000000
                           9       0.2121740          0.000000
```

Fig. 8 Minimum requirement of manpower in general department in case of “high” linguistic



```
Solution Report - MPT -Hig st
Global optimal solution found.
Objective value:                              0.2579536
Infeasibilities:                              0.000000
Total solver iterations:                             9
Elapsed runtime seconds:                          0.05
Model is convex quadratic

Model Class:                                        QP

Total variables:                      3
Nonlinear variables:                  3
Integer variables:                    0

Total constraints:                    9
Nonlinear constraints:                1

Total nonzeros:                      21
Nonlinear nonzeros:                   3

                    Variable           Value        Reduced Cost
                          X1       0.3697531        0.3975886E-04
                          X2       0.2923233       -0.4767551E-04
                          X3       0.1891648       -0.4308771E-05

                         Row    Slack or Surplus      Dual Price
                           1       0.2579536         -1.000000
                           2       0.9515489E-07     -0.3439353E-01
                           3        1.074711         -0.2158891E-08
                           4       0.9026995         -0.2284447E-08
                           5       0.8719658E-06     -0.1719718E-01
                           6        2.080812         -0.1193910E-08
                           7       0.3697531          0.000000
                           8       0.2923233          0.000000
                           9       0.1891648          0.000000
```

Fig. 9 Minimum requirement of staff in all departments in case of “high” linguistic

Table 9 Dataset of faculty members in engineering department under $n_1 = 10$ for "low" variable

p_i = Low(0–10)	c_i = Low(0–7)	a_i = Low(0–4)
$p_1 = 10$	$c_1 = 4$	$a_1 = 2$
$p_2 = 5$	$c_2 = 6$	$a_2 = 1$
$p_3 = 3$	$c_3 = 5$	$a_3 = 4$
$p_4 = 4$	$c_4 = 2$	$a_4 = 3$
$p_5 = 2$	$c_5 = 1$	$a_5 = 2$
$p_6 = 7$	$c_6 = 6$	$a_6 = 2$
$p_7 = 4$	$c_7 = 7$	$a_7 = 3$
$p_8 = 3$	$c_8 = 4$	$a_8 = 3$
$p_9 = 6$	$c_9 = 5$	$a_9 = 2$
$p_{10} = 5$	$c_{10} = 6$	$a_{10} = 1$

Objective value (Z_1) = 11.11111 with $x_1 = 2.222417$, $x_2 = 1.110850$, $x_3 = 2.222158$

Table 10 Dataset of faculty members in engineering department under $n_1 = 10$ for "medium" variable

p_i= Medium(8–15)	c_i= Medium(5–11)	a_i= Medium(3–7)
$p_1 = 14$	$c_1 = 11$	$a_1 = 3$
$p_2 = 15$	$c_2 = 5$	$a_2 = 7$
$p_3 = 13$	$c_3 = 7$	$a_3 = 4$
$p_4 = 14$	$c_4 = 9$	$a_4 = 5$
$p_5 = 13$	$c_5 = 8$	$a_5 = 6$
$p_6 = 8$	$c_6 = 7$	$a_6 = 5$
$p_7 = 9$	$c_7 = 11$	$a_7 = 3$
$p_8 = 10$	$c_8 = 9$	$a_8 = 4$
$p_9 = 9$	$c_9 = 5$	$a_9 = 6$
$p_{10} = 12$	$c_{10} = 7$	$a_{10} = 2$

Objective value (Z_1) = 0.7255139 with $x_1 = 0.5925035$, $x_2 = 0.4836758$, $x_3 = 0.3748483$

Based on "high" linguistic variable, optimal value is 0.2923977, where decision variables are 0.4093851, 0.3215861, and 0.14262323.

Tables 12, 13 and 14 show dataset of faculty member of general department based on linguistic variables "low", "medium", and "high" with $n_1 = 10$ where values of low, medium, and high are generated randomly based on linguistic range of the linguistic variables and finally produce different optimal values based on linguistic variables. In faculty members of general department, based on "low" linguistic variable, optimal value is 7.142858, where values of decision variables are 1.429100, 2.142328, and 0.7148151. Based on "medium" linguistic variable, optimal value is 0.7042254, where decision variables are 0.6338078, 0.3520052, and 0.4226174.

Table 11 Dataset of faculty members in engineering department under $n_1 = 10$ for "high" variable

p_i= High (13–20)	c_i= High(10–15)	a_i= High(5–12)
$p_1 = 14$	$c_1 = 11$	$a_1 = 5$
$p_2 = 13$	$c_2 = 12$	$a_2 = 11$
$p_3 = 19$	$c_3 = 11$	$a_3 = 12$
$p_4 = 14$	$c_4 = 15$	$a_4 = 12$
$p_5 = 15$	$c_5 = 15$	$a_5 = 11$
$p_6 = 14$	$c_6 = 14$	$a_6 = 9$
$p_7 = 16$	$c_7 = 12$	$a_7 = 7$
$p_8 = 18$	$c_8 = 13$	$a_8 = 6$
$p_9 = 13$	$c_9 = 12$	$a_9 = 8$
$p_{10} = 17$	$c_{10} = 15$	$a_{10} = 10$

Objective value $(Z_1) = 0.2923977$ with $x_1 = 0.4093851$, $x_2 = 0.3215861$, $x_3 = 0.14262323$

Table 12 Dataset of faculty members in general department under $n_1 = 10$ for "low" variable

p_i= Low(0–10)	c_i= Low(0–7)	a_i= Low(0–4)
$p_{11} = 7$	$c_{11} = 2$	$a_{11} = 2$
$p_{12} = 2$	$c_{12} = 3$	$a_{12} = 1$
$p_{13} = 4$	$c_{13} = 1$	$a_{13} = 3$
$p_{14} = 3$	$c_{14} = 2$	$a_{14} = 2$
$p_{15} = 7$	$c_{15} = 6$	$a_{15} = 1$

Objective value $(Z_2) = 7.142858$ with $x_1 = 1.429100$, $x_2 = 2.142328$, $x_3 = 0.7148151$

Table 13 Dataset of faculty members in general department under $n_1 = 10$ for "medium" variable

p_i= Medium (8–15)	c_i= Medium(5–11)	a_i= Medium(3–7)
$p_{11} = 9$	$c_{11} = 5$	$a_{11} = 6$
$p_{12} = 10$	$c_{12} = 7$	$a_{12} = 4$
$p_{13} = 11$	$c_{13} = 12$	$a_{13} = 3$
$p_{14} = 10$	$c_{14} = 11$	$a_{14} = 5$
$p_{15} = 9$	$c_{15} = 8$	$a_{15} = 4$

Objective value $(Z_2) = 0.7042254$ with $x_1 = 0.6338078$, $x_2 = 0.3520052$, $x_3 = 0.4226174$

Table 14 Dataset of faculty members in general department under $n_1 = 10$ for "high" variable

p_i= High(13–20)	c_i= High(10–15)	a_i= High(5–12)
$p_{11} = 13$	$c_{11} = 12$	$a_{11} = 8$
$p_{12} = 16$	$c_{12} = 11$	$a_{12} = 11$
$p_{13} = 18$	$c_{13} = 10$	$a_{13} = 10$
$p_{14} = 15$	$c_{14} = 15$	$a_{14} = 9$
$p_{15} = 19$	$c_{15} = 13$	$a_{15} = 12$

Objective value $(Z_2) = 0.2652520$ with $x_1 = 0.3448288$, $x_2 = 0.31833195$, $x_3 = 0.2121740$

Based on "high" linguistic variable, optimal value is 0.2652520, where decision variables are 0.3448288, 0.31833195, and 0.2121740.

Tables 15, 16 and 17 show dataset of staff members of all departments based on linguistic variables "low", "medium", and "high" with $n_1 = 10$ where values of low, medium, and high are generated randomly based on linguistic range of the linguistic variables and finally produce different optimal values based on linguistic variables. In staff members of all departments, based on "low" linguistic variable, optimal value is 2.439025, where values of decision variables are 1.463519, 0.4876172, and 0.24364993. Based on "medium" linguistic variable, optimal value is 0.6615779, where decision variables are 0.5700183, 0.4325502, and 0.3867263. Based on "high" linguistic variable, optimal value is 0.2579536, where decision variables are 0.3697531, 0.2923233, and 0.1891648. Summarized table shown in

Table 15 Dataset of staff members in all department under $n_1 = 10$ for "low" variable

p_i= Low(0–10)	c_i= Low(0–7)	a_i= Low(0–4)
$p_{16} = 10$	$c_{16} = 6$	$a_{16} = 4$
$p_{17} = 8$	$c_{17} = 2$	$a_{17} = 3$
$p_{18} = 9$	$c_{18} = 5$	$a_{18} = 1$
$p_{19} = 6$	$c_{19} = 2$	$a_{19} = 1$
$p_{20} = 7$	$c_{20} = 6$	$a_{20} = 2$

Objective value $(Z_3) = 2.439025$ with $x_1 = 1.463519$, $x_2 = 0.4876172$, $x_3 = 0.24364993$

Table 16 Dataset of staff members in all department under $n_1 = 10$ for "medium" variable

p_i= Medium(8–15)	c_i= Medium(5–11)	a_i= Medium(3–7)
$p_{16} = 15$	$c_{16} = 10$	$a_{16} = 5$
$p_{17} = 14$	$c_{17} = 11$	$a_{17} = 6$
$p_{18} = 9$	$c_{18} = 5$	$a_{18} = 7$
$p_{19} = 13$	$c_{19} = 8$	$a_{19} = 7$
$p_{20} = 8$	$c_{20} = 9$	$a_{20} = 4$

Objective value $(Z_3) = 0.6615779$ with $x_1 = 0.5700183$, $x_2 = 0.4325502$, $x_3 = 0.3867263$

Table 17 Dataset of staff members in all department under $n_1 = 10$ for "high" variable

p_i= High(13–20)	c_i= High(10–15)	a_i= High(5–12)
$p_{16} = 15$	$c_{16} = 12$	$a_{16} = 5$
$p_{17} = 14$	$c_{17} = 15$	$a_{17} = 8$
$p_{18} = 12$	$c_{18} = 16$	$a_{18} = 11$
$p_{19} = 13$	$c_{19} = 10$	$a_{19} = 12$
$p_{20} = 17$	$c_{20} = 14$	$a_{20} = 9$

Objective value $(Z_3) = 0.2579536$ with $x_1 = 0.3697531$, $x_2 = 0.2923233$, $x_3 = 0.1891648$

Table 18 Optimal values of all mathematical modelling based on all linguistic variables

Linguistic variable	Engineering department	General department	Staff members
Low	11.11111	7.142858	2.439025
Medium	0.7255139	0.7042254	0.6615779
High	0.2923977	0.2652520	0.2579536

Table 18 where optimal values of both departments and staff of all department show clearly. It is observed that highest optimal value in all linguistic variables is for engineering department, then general department, and then staff members. Hence, maximum manpower required in engineering department, then general department, and then for staff members.

6 Conclusions

The proposed method efficiently illustrates the managing strategy of the manpower system in engineering and general departments as well as staffs of all department with the help of quadratic programming and fuzzy logic. Quadratic programming plays the role of mathematical optimization that modelled the different type of faculty members in the context of decision variables. Fuzzy logic is used to estimate imprecise information by reducing uncertainty related to the optimal manpower in the institute. The combination of both provides an efficient mathematical modelling that easily derives optimal solutions of each manpower strategy. Finally, it is observed that highest manpower is required in engineering department with context of all possibility of the fuzzy logic variables and lowest manpower is required for staffs in combination of all department with context of all possibility of the fuzzy logic variables.

References

1. Gupta, K. M. (2014). *Application of linear programming techniques for staff training*. Department of Science & Humanity (Mathematics).
2. Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly, 3*(1–2), 95–110.
3. Binh, H. T. T., & Dey, N. (Eds.). (2018). *Soft computing in wireless sensor networks*. CRC Press.
4. Binh, H. T. T., Hanh, N. T., Nghia, N. D., & Dey, N. (2020). Metaheuristics for maximization of obstacles constrained area coverage in heterogeneous wireless sensor networks. *Applied Soft Computing, 86,* 105939.
5. Yager, R. R., & Zadeh, L. A. (Eds.). (2012). *An introduction to fuzzy logic applications in intelligent systems* (Vol. 165). Springer Science & Business Media.
6. Aboelmagd, Y. M. (2018). Linear programming applications in construction sites. *Alexandria Engineering Journal, 57*(4), 4177–4187.

7. Rodias, E. C., Sopegno, A., Berruto, R., Bochtis, D. D., Cavallo, E., & Busato, P. (2019). A combined simulation and linear programming method for scheduling organic fertiliser application. *Biosystems Engineering, 178,* 233–243.
8. Ji, Y., Li, M., & Qu, S. (2018). Multi-objective linear programming games and applications in supply chain competition. *Future Generation Computer Systems, 86,* 591–597.
9. Jain, P. K., Quamer, W., & Pamula, R. (2018, April). Electricity consumption forecasting using time series analysis. In *International Conference on Advances in Computing and Data Sciences* (pp. 327–335). Singapore: Springer.
10. Kumari, P., Jain, P. K., & Pamula, R. (2018, March). An efficient use of ensemble methods to predict students academic performance. In *2018 4th International Conference on Recent Advances in Information Technology (RAIT)* (pp. 1–6), IEEE.
11. Mishra, G., Agarwal, S., Jain, P. K., & Pamula, R. (2019). Outlier detection using subset formation of clustering based method. *International Conference on Advanced Computing Networking and Informatics* (pp. 521–528). Singapore: Springer.
12. Punam, K., Pamula, R., & Jain, P. K. (2018, September). A two-level statistical model for big mart sales prediction. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 617–620). IEEE.
13. Yang, W., Wang, X., Song, X., Yang, Y., & Patnaik, S. (2018). Design of intelligent transportation system supported by new generation wireless communication technology. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications* (pp. 715–732). IGI Global.
14. Jayakumar Loganathan, J., & Subbiah, J. (2020). Energy aware dynamic mode decision for cellular D2D communications by using integrated multi-criteria decision making model. *International Journal of Ambient Computing and Intelligence, 11*(3), 7 February 2020, IGI Global.
15. Chandrakar, P. (2019). A secure remote user authentication protocol for healthcare monitoring using wireless medical sensor networks. *International Journal of Ambient Computing and Intelligence (IJACI), 10*(1), 96–116.
16. Gharanjik, A., Soltanalian, M., Shankar, M. B., & Ottersten, B. (2019). Grab-n-Pull: A max-min fractional quadratic programming framework with applications in signal and information processing. *Signal Processing, 160,* 1–12.
17. Hempel, A. B., Goulart, P., & Lygeros, J. (2012). Inverse parametric quadratic programming and an application to hybrid control. *IFAC Proceedings Volumes, 45*(17), 68–73.
18. Das, S. K., & Tripathi, S. (2015). Energy efficient routing protocol for manet based on vague set measurement technique. *Procedia Computer Science, 58,* 348–355.
19. Das, S. K., & Tripathi, S. (2016). Energy efficient routing protocol for MANET using vague set. In *Proceedings of fifth international conference on soft computing for problem solving* (pp. 235–245). Singapore: Springer.
20. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015, February). Fuzzy based energy efficient multicast routing for ad-hoc network. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (pp. 1–5). IEEE.
21. Das, S. K., & Tripathi, S. (2018). Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming. *Applied Intelligence, 48*(7), 1825–1845.
22. Das, S. K., Tripathi, S., & Burnwal, A. P. (2015, February). Design of fuzzy based intelligent energy efficient routing protocol for wanet. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (pp. 1–4). IEEE.
23. Mishra, B. K., Yadav, B., Jha, S. K., & Burnwal, A. P. (2015). Fuzzy set theory approach to model super abrasive grinding process using weighted compensatory operator. *Int J Res Comput Appl Robot-IJRCAR, 3,* 62–68.

24. Murmu, S., Jha, S. K., Burnwal, A. P., & Kumar, V. (2015). A proposed fuzzy logic based system for predicting surface roughness when turning hard faced components. *International Journal of Computer Applications, 125*(4).
25. Kumari, N., & Burnwal, A. P. (2017). Interactive fuzzy programming model in multi-objective inventory control problem using various operators. *International Journal of Students' Research in Technology & Management, 5*(4), 18–26.