RESEARCH ON KEY TECHNOLOGIES IN WIRELESS COMMUNICATIONS BASED ON EVOLUTIONARY ALGORITHMS

by

Jie Zhou

Dissertation submitted in fulfilment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

Department of Engineering
Faculty of Science and Engineering
Macquarie University
Sydney, Australia

08th of Month 15

ABSTRACT

Wireless communications have been developing rapidly in both the research and industry. However, such developments have raised many challenges across various layers in wireless communications. In particular, many optimization processes have been proved to be NP-hard problems, which hinder further technological advances.

Evolutionary Algorithm (EA), as part of the Artificial Intelligence, provides a generic heuristic optimization technique motivated by natural evolution. EAs usually work well in a category of combinatorial NP-hard problems by building better solutions through the recombination of the best part of past solutions, rather than attempting all possible combinations. As such EA is able to find near optimal solutions through solution generation, selection and rearrangement, reducing the complexity of solving NP-hard problems.

Our research is motivated by the need to optimize difficult discrete optimization problems in wireless communication systems. In particular, we developed novel EA algorithms to address some of the NP-hard problems across various layers, ranging from Physical layer, Data Link layer, to Network layer, in wireless communication systems. We demonstrate that our EA designs achieve significant performance improvements for the systems under investigation with lower computational complexity and fast convergence.

The main innovations and contributions of this thesis are as follows:

In the Physical layer, we take on the challenge of the reduction of the peak-

to-average power ratio (PAPR) in orthogonal frequency division multiplexing (OFDM) systems. We propose a modified chaos clonal shuffled frog leaping algorithm (MCCSFLA) for PAPR reduction. We also analyze MCCSFLA using Markov chain theory and prove that the proposed algorithm converges to the global optimum. Simulation results show that the proposed algorithm achieves better PAPR reduction than using others heuristics. Additionally, MCCSFLA has lower computational complexity and faster convergence than other heuristics.

In the Data link layer, we investigate the target coverage problem in large-scale self-organizing wireless sensor networks (WSNs), we propose a method based on a quantum ant colony evolutionary algorithm. We build the WSNs target coverage system model, design and apply the proposed method to WSNs. Simulation results show that the proposed method can significantly increase the target coverage rate in WSNs. Furthermore, we expand our investigation to the communication coverage problem in WSNs. We propose a new quantum immune clonal evolutionary algorithm for the duty cycle sequence design with a full coverage constraint. Simulation results show that the proposed algorithm not only maintains full coverage of all the targets in the monitoring area but also extends the network lifetime of the WSN. Additionally, in order to reduce the communication energy consumption in large-scale WSNs, we propose a fuzzy simulated evolutionary computation clustering method. We design a fuzzy controller for the algorithm parameter adjustment. Simulation results show that the proposed method can significantly reduce the energy consumption of large-scale WSNs.

In the Network layer, we take on the challenge of providing Quality of Service (QoS) routing for multimedia wireless sensor networks. A novel parallel elite clonal quantum evolutionary algorithm is proposed to solve the multi-constraints

QoS routing problem. Simulation results demonstrate that the proposed algorithm achieves lower energy consumption at a faster convergence rate than the other heuristic algorithms.

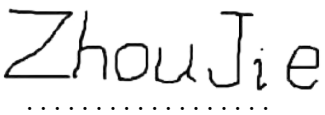
Wireless communication is a key technology in the modern society and we believe new evolutionary algorithms can contribute a growing number of solutions in this area.

STATEMENT OF CANDIDATE

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me.

In addition, I certify that all information sources and literature used are indicated in the thesis.



Jie Zhou

ACKNOWLEDGMENTS

I would like to acknowledge the help and support of all people in making this dissertation possible.

First and foremost, I would like to thank my supervisors Professor Eryk Dutkiewicz, Dr. Ren Ping Liu, Dr. Gengfa Fang for their trust, guidance and support over the past years.

Moreover, I would like to thank Dr. Xiaojing Huang for his help.

Thanks also to the many people who have provided help and guidance at BUPT, including Prof. Yuanan Liu, Prof. Bihua Tang, Prof. Yunxiao Zu and other teachers.

I am grateful for the support and help from the Department of Engineering and Faculty of Science and Engineering, staff and graduate students of the Macquarie University.

My special thanks go to Macquarie University, who kindly provided the iMQRES scholarship to support my study and living.

Last but not least, I want to express my deepest gratitude to my family, my parents and my wife, for their endless love, encouragement, and support.

To My Parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first describe the definitions of relevant terms and background of our research, then present motivation and problems statements of the thesis. We then summarize the contributions of this thesis. The outline of the thesis is presented at the end of this chapter.

1.1 Background and Definitions

Recent innovations in signal processing, mobile networks and microprocessor technology have assisted rapid development and application of wireless communications, in particular, cellular networks and wireless sensor networks [1]. Such systems attract specific attention due to their benefits, such as low deployment costs, no need for fixed cables, wide coverage, flexible deployment and robustness. These characteristics are specifically valuable in applications including earthquake relief, wartime signal transmission and communications in harsh environment, in which the deployment of wired communication facilities isn't feasible or is impractical [2].

Orthogonal frequency-division multiplexing (OFDM) is a technique of coding digital data on a number of carrier frequencies [3]. OFDM separates the accessible spectrum into

numerous tightly spaced orthogonal sub-carriers and each sub-carrier is modulated with a traditional modulation method. It has been widely employed in applications such as DSL Internet access, broadband wire-line and wireless communications, power-line networks, audio broadcasting and digital television. OFDM offers a high capabilities physical layer and provides excellent benefits of high spectrum efficiency, flexibility to channel conditions and sturdiness towards inter-carrier as well as inter-symbol interference.

However, it is widely recognized that OFDM systems are afflicted by high *peak-to-average power ratio (PAPR)*. In OFDM systems, the potential for the sub-carrier signals adding up coherently to generate high peaks in the time domain remains to be one among its main issues [4]. High PAPR leads to the receiver's performance becoming sensitive to nonlinear equipment, such as power amplifier and digital-to-analog converter. Such high PAPR requires the linear amplifier to have a larger active range which can be hard to accommodate.The key issue is that if a high PAPR signal goes through a power amplifier, the power amplifier with nonlinear features might be in a saturation region, causing undesirable out-of-band and in-band distortion. The existence of extremely high peaks suggest that the power amplifier needs to work beyond its linear region to fit the full amplitude signal swings.

Wireless sensor networks (WSNs) are self-organizing sensor nodes that keep tracking the entity or surrounding situations, including heat level, audio, humidity, and collaboratively transfer the information to the user towards a sink node [5]. The growth of WSNs was initially inspired by army programs like battleground monitoring. Nowadays such systems are employed in many commercial and individual programs, including manufacturing procedure supervision and management, equipment fitness tracking, etc. WSNs consist of sensors, ranging from a few to many thousands connected sensor nodes. Each WSN node consists of a number of components: a micro-computer, an inner or outer aerial, a digital circuit for integrating with a battery and sensor(s). WSN nodes may differ in

sizes from several cubic decimeters to the length of a rice. The cost of a sensing unit also ranges between less than one dollar to thousands of dollars, depending upon the intricacy of the sensing unit. Price and size restrictions on sensing units lead to limitations on equipment including power, storage, computer performance and spectrum bands. The scale of WSNs may differ from a common single hop structure to sophisticated multihop systems. The distribution strategy among the hops in the system can be point-to-point, multipoint-to-point, or point-to-multipoint.

The *target coverage* is one of the fundamental issues in WSN for environment monitoring, military affairs and homelands security. It determines how well a target is tracked or monitored by sensors [6]. The target coverage in a WSN needs to guarantee that some specific physical quantity of the targets over the entire area is monitored with a certain level of reliability. Target coverage is often classified as k-coverage, Q-coverage and simple coverage [7]. In simple coverage, each target must be examined by no less than one sensor node. In practical engineering practice, this is the most common situation. So in this thesis, we only study the simple coverage problem.

In many applications, the nodes in WSNs must work for long-term uses and the network lifespan can vary from a couple of months to many years. On the other hand, the nodes must operate with a battery with limited power sources and the battery may not be replaced easily. To deal with the contradiction between restricted power and application lifespan demands, it is crucial to reduce energy consumption via sleep-wake scheduling, i.e. *duty cycle* operation [8]. *Clustering* is another common strategy that could lower power consumption of sensors and extend the WSN lifespan [9].

Duty cycles turn off sensors periodically to save power and a period is the time period it requires for one node to accomplish an on-and-off cycle. In a common WSN system, sensor nodes are spread within an area from where they gather information to accomplish specific objectives. Direct communications between sensors and the sink node are not efficient, so

long-range transmissions ought to be prevented to minimize energy consumption so as to extend the system lifespan. Sensors usually send their data to the distant sink node in a coordinated manner. Each cluster includes a manager, known as the cluster head, as well as several member nodes. Clustering produces a two-level structure where cluster heads make up the higher level and member nodes make up the lower level. Through clustering, a head node gathers information from member nodes in their cluster and then the cluster head delivers the gathered data towards the sink node. The clustering techniques are able to enhance network lifetime and increase power efficiency by reducing total power consumption.

Recent developments in multimedia components and wireless communications have enabled the application of multimedia sensors in different monitoring programs. Systems of connected multimedia sensor nodes are generally known as *multimedia wireless sensor networks*, where a number of multimedia sensors work together to produce enriched monitoring of surroundings [10]. Recent studies have demonstrated the significance of power saving in the networks. As the networks have higher data rate and higher QoS requirement than ordinary WSNs, the layout of the networks routing procedure with good power efficiency remains crucial and encounters more difficulties, with considerations to bandwidth, delay, delay jitter and packet loss rate. An essential issue in the networks is the way to enhance quality of service (QoS) and minimize power consumption since the multimedia like videos and voices have high QoS requirements. Within the networks, each node may behave as either a source node or a relay node or both. The routing problem noticeably impacts the end-to-end service quality of the networks, raising many questions [11]. As such, finding the best route that could satisfy the required QoS and reduce the power consumption is important to the networks.

1.2 Problem Statements and Motivations

With the rapid development of wireless communications and technology, more and more researchers pay great attention to some bottleneck problems in the optimization process of wireless communication, especially NP-hard problems[12][13].

Evolutionary algorithm (EA) is a generic heuristic optimizing technique, which is a part of artificial intelligence (AI)[14]. EA employs methods motivated by natural evolution. Prospective answers to the problem act as individuals, while the fitness function figures out the grade of the individuals. Evolution of the individuals then happens after the repeated use of generic operators, including selection, crossover, mutation and recombination, etc.

Our research is motivated by the need to optimize difficult discrete optimization problems in wireless communication systems[15][16][17][18][19]. We investigate the discrete optimization issues in five different scenarios, establish system models and objective functions for the problems and design novel evolutionary algorithms for the relevant problems.

1.2.1 Efficient and Low Complexity PAPR Reduction Method in OFDM Systems

Reduction of PAPR is an implementation challenge in OFDM systems [20] [21] [22]. One way to reduce PAPR is to apply a set of selected partial transmission sequences (PTS) to the transmit signals [23] [24] [25]. However, PTS selection is a highly complex NP-hard problem and the computational complexity is very high when a large number of subcarriers are used in the OFDM system [26] [27] [28].

In Chapter 3, we propose a new heuristic PTS selection method, the modified chaos clonal shuffled frog leaping algorithm (MCCSFLA). MCCSFLA is inspired by natural clonal selection of a frog colony and it is based on the chaos theory. We also analyse

MCCSFLA using the Markov chain theory and prove that the algorithm converges to the global optimum. Simulation comparisons are conducted by using the proposed method, genetic, quantum evolutionary and selective mapping algorithms.

1.2.2 Efficient Target Coverage Method for Self-organizing Wireless Sensor Networks

Target coverage is a critical issue in self-organizing wireless sensor networks since the sensing range of each sensor node is limited and usually cannot be adjusted[29][30]. A coverage design with a target selection mechanism is useful for improving the target coverage rate by selecting a set of targets for each sensor node within its sensing range[31][32]. However, target selection for a large amount of sensor nodes is a highly complex NP-hard problem and the computational complexity for searching all the combinations is too high for practical implementation[33].

In Chapter 4, we propose a target coverage scheme based on the quantum ant colony evolutionary algorithm (QACEA) taking into consideration of both the sensor node distribution and the target distribution. We also establish a system model for the target coverage problem. Simulations are conducted for the proposed method and comparisons are made with the schemes based on the genetic algorithm and the simulated annealing algorithm.

1.2.3 Energy Efficient Duty Cycle Design in Wireless Sensor Networks

Duty cycle design is an important topic in WSNs[34][35][36]. As small sensors are equipped with a limited power source, the extension of network lifetime is generally achieved by reducing network energy consumption, for instance, through duty cycle schemes. However,

duty cycle design is a highly complex NP-hard problem and its computational complexity is too high with an exhaustive search algorithm for practical implementation.

We propose in Chapter 5 a novel quantum immune clonal evolutionary algorithm (QICEA) for duty cycle design to extend WSNs lifetime and maintain full coverage in the monitoring area. QICEA is tested by simulations, and performance comparisons are made with simulated annealing and genetic algorithm.

1.2.4 Low Energy Clustering in High-density Wireless Sensor Networks

Energy efficient clustering is an NP-hard problem in wireless sensor networks [37] [38] [39] [40] [41] [42]. In Chapter 6, a low energy clustering method of high-density wireless sensor networks based on fuzzy simulated evolutionary computation (FSEC) is proposed. To reduce communications energy consumption, we also design a fuzzy controller to dynamically adjust the crossover and mutation probability. Simulations are conducted by using the proposed method, and results are compared with that of the clustering methods based on the particle swarm optimization and the method based on the quantum evolutionary algorithm.

1.2.5 Energy Efficient QoS Routing for Multimedia Wireless Sensor Networks

Quality of Service (QoS) routing is one of the key enabling techniques for multimedia wireless sensor networks [43] [44] [45] [46]. However, the multi-constrained QoS routing problem is an NP-hard problem and the computational complexity of an exhaustive search over all the paths is too high for large scale multimedia wireless sensor networks.

In Chapter 7, a novel parallel elite clonal quantum evolutionary algorithm (PECQEA)

is proposed to solve the multi-constrained QoS routing problem. The proposed algorithm minimizes network energy consumption, while guaranteeing QoS performance, including delay, bandwidth, delay jitter and packet loss rate, in multimedia wireless sensor networks. We analyze the convergence property of PECQEA by Markov chain theory and then prove the algorithm converges to the global optimum. The algorithm is tested by extensive simulations and its performance is compared with the genetic algorithm and ant colony optimization.

1.3 Summary of Contributions

In this thesis, we propose five new solutions in wireless communications based on evolutionary algorithms. The main innovations and contributions of this thesis are as follows:

(1) We propose a novel modified chaos clonal shuffled frog leaping algorithm for PAPR reduction in OFDM systems. The algorithm is motivated by the biological character of frogs, and inspired by chaos theory and clonal selection. MCCSFLA combines the nature inspired local search together with the global information exchange between groups and employs clonal selection. With these novel combinations of strategies, MCCSFLA can prevent local suboptimal points and lead the search to the global optimum solution. We present detailed algorithm design of MCCSFLA in chapter 3. We analyze MCCSFLA using Markov chain theory and prove that the algorithm can converge to the global optimum. Simulation results show that the MCCSFLA is more efficient in terms of PAPR reduction. In a typical OFDM system with QPSK modulation, the average PAPR obtained by MCCSFLA is 0.58dB to 0.71dB less than GA, 1.37dB to 1.59dB less than QEA, and 1.62dB to 1.93dB less than SLM with different number of subcarriers. In a typical OFDM system with 16QAM modulation, the average PAPR obtained by MCCSFLA is 0.74dB to 0.91dB less than GA, 1.77dB to 2.17dB less than QEA, and 2.68dB to

2.75dB less than SLM with different number of subcarriers. Similar conclusions can be also obtained with 64QAM modulation. Moreover, MCCSFLA has lower computational complexity and faster convergence speed than other heuristics.

(2) We propose a new method based on quantum ant colony evolutionary algorithm for the target coverage problem in large-scale self-organizing WSNs. We use the concepts of quantum computation as well as the quantum theory. We develop a novel set of binary ants and a set of quantum ants within the ant colony, that is a mixture of deterministic and probabilistic illustration and uses both ant investigation and a quantum gate to push the ant colony to the optimum solution. In addition to the colony, we maintain an elite binary ant in each iteration. Since the quantum bit illustration can signify a linear superposition of all possible answers, QACEA features a superior characteristic of diversity than GA and SA. Simulation results show that the proposed method can significantly increase WSN target coverage rate in contrast to the traditional GA and SA. As the results show, the target coverage rate of the proposed QACEA method is approximately 10% and 20% higher than that of GA and SA respectively with different sensing radius. When the number of nodes are 100 to 200, it can be seen that the number of detected targets by QACEA is 6.90% to 19.09% higher than that of GA and 32.67% to 54.27% higher than that of SA which indicates a higher target coverage rate.

(3) In WSNs, we propose a new quantum immune clonal evolutionary algorithm for the duty cycle sequence design with a full coverage constraint. With the new design, QICEA mixes quantum computation and an artificial immune clonal algorithm in which the components of the antibody population are Q-bits. Compared with the conventional repetitive optimization techniques, QICEA with quantum bit illustration can discover the search area using a linear superposition of states. Unlike the traditional QEA, a novel clonal selection operation is used in QICEA to look for the antibody with the maximum affinity. We also design a novel QICEA based duty cycle design in WSNs that can obtain

a longer network lifetime while keeping full coverage in the monitoring area. Simulation results show that the proposed algorithm not only maintains full coverage of all the targets in the monitoring area, but also extends the WSN lifetime. Results show that the network lifetime of the proposed QICEA method is 4.74% and 5.67% longer than that of GA and 6.04% and 6.74% longer than that of SA which means QICEA has a higher energy efficiency. It also can be seen from results that QICEA finds a high-quality duty cycle scheme much faster than SA and GA.

(4) In order to reduce communications energy consumption, we present a novel fuzzy simulated evolutionary computation clustering method for clustering design in large-scale WSNs. We also design a novel fuzzy controller and the relevant membership function based on the fuzzy control concept. The method can adaptively adjust the crossover probability and mutation probability. With the new design, when the average fitness of the population is too high, the fuzzy controller changes the crossover probability and mutation probability to a low value to prevent population diversity decrease. In the same way, if the average fitness of the population is low, the fuzzy controller changes crossover probability and mutation probability to a high value to improve the population diversity. Simulation results show that the proposed method can significantly reduce the energy consumption of large-scale WSNs communications. More specifically, with 100, 200, 300 and 400 nodes and cluster head proportion of 10%, the communications energy consumption of the fuzzy simulated evolutionary computation is 4.6% to 9.8% less than that of PSO and 20.6% to 29.7% less than that of QEA for different numbers of nodes. When the number of the nodes increased from 200 to 1200, the communications energy consumption of the fuzzy simulated evolutionary computation is 2.34% to 36.02% lower than that of PSO and 18.41% to 61.31% lower than that of QEA for different clusterhead proportions and node numbers.

(5) For multimedia wireless sensor networks, we develop a novel parallel elite clon-

al quantum evolutionary algorithm for QoS routing in multimedia WSNs. We design PECQEA with multiple populations, which combines the advantages of both the clonal selection algorithm and quantum evolutionary algorithm. It has fast search speed and global search ability in complicated search spaces which is ideal for combinatorial optimization problems. In order to improve the search effectiveness, a novel clonal operator is designed in every sub-population of PECQEA. A novel chaotic mutation operator is also designed in PECQEA, which could efficiently add diversity to the population. With the clonal and the mutation operators, the algorithm can efficiently prevent a local optimum. Furthermore, it can quickly converge towards the optimal area throughout the search process. In addition, PECQEA is flexible and powerful when searching in a large space, as it can keep individual diversity in different populations. The Markov chain theory is then employed to assess the convergence of PECQEA. First, we demonstrate that the evolution process of PECQEA is a Markov chain. Then we model and study the transition matrix of the improvement procedure for PECQEA. The proof demonstrates PECQEA can converge to the global optimum within limited generations.

Furthermore we develop an objective function to reduce route energy consumption under several restrictions. Simulation results demonstrate that the proposed algorithm achieves lower energy consumption at a faster convergence rate than the other heuristic algorithms. After 50 iterations, the average energy cost found by PECQEA is 0.6629J. In contrast, GA cost is 0.6863J, and ACO cost is 0.7201J. These results show that PECQEA achieved more than 3.4% and 7.9% energy cost reductions over GA and ACO, respectively. Moreover, PECQEA also enjoys a faster convergence rate than GA and ACO. The energy cost of PECQEA is reduced to below 0.7 after only 9 generations, while GA takes 34 generations to reach that cost level and ACO takes more than 50 generations.

1.4 List of Publications

The publications of the author during his PhD study are listed as follows.

Journal Papers

1. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang, Yuanan Liu, Xiaojing Huang, "A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems," *IEEE Transactions on Broadcasting*, Accepted by IEEE Early Access Articles, 2015. (Corresponding to Chapter 3)

2. **Jie Zhou**, Yuan-An Liu, Fan Wu, Hong-Guang Zhang, Yun-Xiao Zu. "Allocation of multi-objective cross-layer wireless sensor network resource based on chaotic parallel genetic algorithm," Acta Phys. Sin., vol 60, 2011, pp. 090504. (indexed in Science Citation Index: WOS:000295114000020) (Corresponding to Chapter 6)

3. **Jie Zhou**, Yun-Xiao Zu, "A parallel immune genetic algorithm in adaptive resource allocation for cognitive radio network," Acta Phys. Sin., vol 59, 2010, pp. 7508 7515. (indexed in Science Citation Index: WOS:000283406400121) (Corresponding to Chapter 5)

4. Yun-Xiao Zu, **Jie Zhou**. "Cognitive radio resource allocation based on combined chaotic genetic algorithm," Acta Phys. Sin., vol 60, 2011, pp. 079501. (indexed in Science Citation Index: WOS:000293366300131) (Corresponding to Chapter 6)

5. Yun-Xiao Zu, **Zhou Jie**, Chang-Chang Zeng. "Cognitive radio resource allocation based on coupled chaotic genetic algorithm," Chinese Physics B, vol 19, 2010, pp. 119501. (indexed in Science Citation Index: WOS:000284973100113) (Corresponding to Chapter 6)

6. Yun-Xiao Zu, **Zhou Jie**, "Multi-user cognitive radio network resource allocation based on the adaptive niche immune genetic algorithm," Chinese Physics B, vol 21, 2012, pp. 019501. (indexed in Science Citation Index: WOS 000300258200089) (Corresponding to Chapter 5)

Conference Papers

1. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang,Yuanan Liu, "QoS Routing Based on Parallel Elite Clonal Quantum Evolution for Multimedia Wireless Sensor Networks," Proceeding of 2014 IEEE Wireless Communications and Networking Conference, Istanbul: IEEE, 2014: 2498-2503. (Corresponding to Chapter 7)

2. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang, Yuanan Liu, "Modified Elite Chaotic Immune Clonal Selection Algorithm for Sever Resource Allocation in Cloud Computing Systems," Proceeding of the 17th International Symposium on Wireless Personal Multimedia Communication, Sydney: IEEE, 2014: 226-231. (Corresponding to Chapter 5)

3. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Xiaojing Huang, Gengfa Fang, Yuanan Liu, "Modified Elite Chaotic Artificial Fish Swarm Algorithm for PAPR Reduction in OFDM Systems," Proceeding of the 2014 14th International Symposium on Communications and Information Technologies, Incheon: IEEE, 2014: 503-507. (Corresponding to Chapter 3)

4. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang, Yuanan Liu, Xiaojing Huang, "A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems," Proceeding of the IEEE TENCON 2014, Bangkok: IEEE 2014:1-6. (Corresponding to Chapter 3)

5. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang,Yuanan Liu, "Target Allocation of WSN Based on Parallel Chaotic Elite Quantum-Inspired Evolutionary Algorithm," Proceeding of 15th International Symposium on Communications and Information Technologies (ISCIT 2015), Nara, Japan 7-9 October 2015. (Accepted) (Corresponding to Chapter 4)

6. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang,Yuanan Liu, "Low Energy Clustering in BAN Based on Fuzzy Simulated Evolutionary Computation," Proceeding

of 10th International Conference on Body Area Networks (BodyNets 2015), Sydney, Australia, September 28-30, 2015. (Accepted) (Corresponding to Chapter 5)

7. **Jie Zhou**, Eryk Dutkiewicz, Ren Ping Liu, Gengfa Fang,Yuanan Liu, "Energy Efficient Duty Cycle Design based on Quantum Immune Clonal Evolutionary Algorithm in Body Area Networks," Proceeding of 10th International Conference on Body Area Networks (BodyNets 2015), Sydney, Australia, September 28-30, 2015. (Accepted) (Corresponding to Chapter 5)

Patents

1. Yuanan Liu, **Jie Zhou**, Fan Wu, Lijia Zhang, Hongguang Zhang, Bihua Tang, Wenhao Fan, Yang Yang, "An approach of multi-mode mobile terminal selects high quality service sector," Application No. 2012105643130, Publication No. CN103052145B. (China Patent) (Corresponding to Chapter 3)

2. Yuanan Liu, **Jie Zhou**, Fan Wu, Lijia Zhang, Hongguang Zhang, Bihua Tang, Wenhao Fan, Yang Yang, "A method for Multimode terminal selects target service network to ensure communication quality," Application No. 2012101850841, Publication No. CN103002520B. (China Patent) (Corresponding to Chapter 4)

3. Yuanan Liu, **Jie Zhou**, Fan Wu, Lijia Zhang, Hongguang Zhang, Bihua Tang, Wenhao Fan, Yang Yang, "Multimode terminal sector selection method based on simulated annealing algorithm," Application No. 2012105590542, Publication No. CN103096431B. (China Patent) (Corresponding to Chapter 3)

4. Fan Wu, **Jie Zhou**, Hongguang Zhang, Yuanan Liu, Lijia Zhang, Bihua Tang, Wenhao Fan, Bo Huang, "A multimode wireless terminal cell selection method," Application No. 2011103710140, Publication No. CN102421153B. (China Patent) (Corresponding to Chapter 3)

5. Yunxiao Zu, **Jie Zhou**, Changchang Zeng, Shibo Mao, Yuting Wang, "Cognitive wireless network cross-layer resource allocation and packet scheduling," Application

No. 201110286908X, Publication No. CN102316594B. (China Patent) (Corresponding to Chapter 6)

1.5 Outline of the Thesis

The rest of the thesis is organized as follows. Chapter 2 briefly presents the necessary theoretical backgrounds of evolutionary algorithms. In the same chapter, we also give a brief literature survey on the applications of these evolutionary algorithms in wireless communications. In Chapter 3, MCCSFLA for PAPR reduction in OFDM systems as well as its detailed convergence proof are presented. In Chapter 4, QACEA is proposed and the method to solve the target coverage problem in large-scale self-organizing wireless sensor networks is presented. Chapter 5 discusses details of QICEA for duty cycle sequence design with full coverage constraint. Chapter 6 briefly introduces FSEC and discusses the implementation details of FSEC for energy efficient clustering design in large-scale wireless sensor networks. In Chapter 7, PECQEA is proposed and the method to solve the multi-constraints QoS routing problem in large-scale self-organizing wireless sensor networks is described.

Chapter 2

Methodology and Literature Review

In this chapter we provide a summary of methodologies and a relevant literature review for the themes covered in this thesis.

2.1 Introduction to Evolutionary Algorithms

EA is a branch of AI which can be based on the Darwinian concepts [14]. EAs employ methods motivated by natural evolution, such as fitness-based selection, crossover, mutation, and reproduction. Prospective answers to the optimization problem act as individuals within a group, while the utility function ascertains the fitness within the environment. During the iterative procedure, two kinds of operations form the foundation of EAs: crossover and mutation produce a new gene and the required variety, while recombination and selection enhance the group fitness. In the fitness-based selection, solutions that have a greater fitness values obtain better opportunities to be chosen compared to solutions with smaller fitness values. Replaced parts of individuals caused by mutation and crossover can be randomly selected. Selections can be either randomly chosen or be selected proportional to fitness. Renewing of the population then occurs after the continued implementation of these operations. Numerous components of this kind of iteration

procedure are stochastic.

Evolutionary algorithms work well for a variety of issues since they do not build any sort of presumption regarding the actual solution scenery. They can be typically employed for multi-objective optimization problems with multiple constraints.

EAs mainly include heuristic iteration algorithms. Generally speaking, the strategies include: artificial immune systems (AIS), swarm intelligence (SI), quantum evolutionary algorithms (QEA) and simulated evolutionary computation (SEC), etc.

2.2 Simulated Evolutionary Computation in Wireless Communications

2.2.1 Introduction to Simulated Evolutionary Computation

Simulated evolutionary algorithms is a group of evolutionary algorithms that are analogous to the natural evolvement, including evolution strategy (ES), genetic programming (GP), evolutionary programming (EP) and genetic algorithms.

2.2.2 Evolution Programming

Evolutionary programming is among the four main EA subsets. EP was first proposed by Lawrence J. Fogel in 1960s and its major evolution measure is mutation [47]. EP is just like genetic programming, however the framework in the method has limitations, with regards to the mathematical factors that are permitted to improve during the evolution. In EP, individuals of the population are thought to be elements of a particular group instead of elements of the identical types.

For EP, like GA, it comes with a fundamental presumption that the fitness landscape is usually indicated with regards to variables, and therefore the best solution (or several

optimum solutions) exists. For instance, if someone were searching for the maximum value in a knapsack problem, each solution is a collection. The collection could be suggested as a number, which may perform the duties of the fitness. The fitness landscape can be described as a hypersurface relative to the value in a solution space. The aim of the knapsack problem is to obtain the globally highest value within this space.

The fundamental EP technique consists of 3 procedures (Repeated until the number of generations is reached or a satisfactory solution is gathered):

(1) Select a preliminary population randomly. The population size is tightly related to convergence speed, but no certain information is offered about what amount of solutions is suitable.

(2) Every single solution is copied towards an additional group. All offspring are changed by different mutation operators. The degree of change is evaluated judging by the transformation on the parents.

(3) Every single offspring is evaluated by calculating its fitness. Generally, an assessment is placed to ascertain the number of solutions to be preserved in the population. There is no criteria that the number must be constant or only one offspring is produced by each parent.

It ought to be noticed that EP normally will not employ the crossover operation as GA does.

2.2.3 Evolution Strategies

In information technology, an evolution strategy refers to an optimization approach using concepts associated with variation and evolvement. It is one of the common types of ES strategies. The ES optimizing approach was designed around the 1960s and improved in depth during the 1970s and then by Ingo R. as well as his co-workers [48].

ES employs biological scenario-based expressions and general selection and mutation,

and explore operations. In ES, the operators are used in a cycle. A repeat of the cycle is termed a generation. The succession of iterations is ongoing until an ending criteria is satisfied.

The general selection in evolution procedures is fixed and simply depends on the fitness (utility function value) rankings and not on the exact fitness figures. The subsequent method thus remains irrelevant with the forms of the fitness function. The easiest improvement approach runs using a group with two individuals: the existing individual as well as the outcome of its mutation. Provided that the outcome's fitness is higher than its parent, it can be the new parent in the next iteration. In terms of real-coded exploration areas, mutation is generally done by increasing a random number to every single element. The movement length and mutation depth are commonly controlled by self-adaptation[49]. Normally, mutants are usually developed and contest with the present solution, known as $(1+\lambda)-ES$. In $(1+\lambda)-ES$ the top mutant turns into the new solution in the following iteration and the present solution is usually ignored. For many of such versions, proofs of convergence are already made on unimodal fitness functions.

Contemporary derivative versions of ES usually employ a group of solutions as well as recombinations for supplemental techniques, known as $(\mu/\rho+\lambda)-ES$. This makes it less likely to fall into local optima[50].

2.2.4 Genetic Programming

In evolutionary algorithms, genetic programming is referred to an iteration strategy motivated by natural evolution to design applications for a certain task. It is a machine learning approach employed to improve a group of software applications based on the utility value distribution driven by an application's capability to conduct a certain computing project.

Typically GP can be a group of commands as well as a utility function to determine

how well an application has performed in a mission. In GP, every individual can be a software application. It shapes software applications, typically stored in RAM as tree structures. These trees could be judged in a recursive way. Each tree node provides a manager attribute and each leaf node possesses an operand, helping to make numerical forms simple to develop and determine.

Randomness in GP leads it to get away from a local optimum where deterministic strategies are likely to be trapped. Similar to GA, GP is very effective at developing new ways of dealing with NP-hard problems. GP employs random operations to create new applications to move out of local optimum where deterministic strategies are likely to be trapped. Specifically, GP evaluate the application performance of the random operations, and select a set of high quality applications for the next iteration where new applications are generated[51]. The key procedures which are employed to generate novel applications from present ones are:

Crossover: The production of an offspring application by mixing arbitrarily selected components from two specific parent applications.

Mutation: The production of a novel offspring application by arbitrarily changing a randomly selected section of a particular parent application.

Principle procedures in the GP method are provided as follows:

1: Arbitrarily generate a starting group of applications through the offered primitives.

2: Start the loop.

3: Run every single application and assess its fitness.

4: Choose 1 or 2 application(s) in the group using a possibility depending on the fitness to take part in genetic procedures.

5: Generate novel child application(s) by making use of genetic procedures with selected chances.

6: Unless a satisfactory result is obtained or other ending condition is satisfied (e.g.,

the highest number of iterations is achieved), go to step 2.

7: Return the best solution.

2.2.5 Genetic Algorithm

In the area of AI, a genetic algorithm refers to an optimization method that imitates the procedure of biological evolution. GA is a member of a group of EAs, which produce answers to optimization issues employing strategies motivated by biological evolution, including fitness selection, mutation, selection and population update. This meta-heuristic is typically employed to produce good answers to searching and optimization problems. GA is the most widely used EA. It has applications in biology, computer science, video processing, statistics, actuarial science, communications and economics, finance, chemistry, numerical analysis and other fields.

In GA, each prospective individual contains a group of components (called chromosomes) that could be mutated and modified. Typically, individuals are symbolized in binary, decimal or float numbers, although alternative encodings may also be feasible[52].

The evolution generally begins with a group of arbitrarily created solutions, and is a repetitive procedure. The group in every loop called a generation. In every iteration, the fitness of each solution in the group is assessed; the fitness is generally the amount of the target function in the issue. The better fit solutions have a greater chance to be chosen from the present group, and every solutions gene is changed (arbitrarily mutated and perhaps recombined) to create a new solution group. The generated group will be employed in the following generation. Generally, GA ends if the maximum amount of iterations reached, or an acceptable degree of fitness was achieved for the group.

2.2.6 Fuzzy Simulated Evolutionary Computation

In Chapter 6, we develop a novel fuzzy simulated evolutionary computation (FSEC) to adaptively adjust the crossover probability and mutation probability. We also design a novel fuzzy controller and the relevant membership function based on the fuzzy control concept.

In FSEC, the crossover probability and mutation probability are key variables which have a substantial influence on the algorithm behavior and performance. Once the crossover probability is bigger, the algorithm would produce new individuals more quickly. However, if the crossover probability is too high, the structure of outstanding individuals in the population will be ruined which results in a slow algorithm convergence rate. If the crossover probability is too low, the search speed of the algorithm becomes very slow. By doing this, the individuals framework will be difficult to destroy that will result in evolutionary stagnation. Similarly, if the mutation probability is larger, the algorithm is likely to ruin the original structure in the population and generate new individuals. However, once the mutation probability is too high, the structure of remarkable individuals in the population will be ruined which results in convergence stagnation. If the mutation probability is too low, the search speed of the algorithm can become very slow. By doing this, the individuals structure can be difficult to destroy, which can result in evolutionary stagnation. In FSEC, when the average fitness of the population is too high, the fuzzy controller changes the crossover probability and mutation probability to a low value to prevent population diversity decrease. In the same way, if the average fitness of the population is low, the fuzzy controller changes crossover probability and mutation probability to a high value to improve the population diversity.

2.2.7 Simulated Evolutionary Computation for Wireless Communications

In wireless sensor networks, a research effort on routing protocols based on GA can be found in [53]. They reduce power usage and improve the system life span under power harvesting restrictions for power harvesting WSNs. A genetic algorithm strategy is suggested in [54] for enhancing the life span of WSNs by arranging procedures. They create a power reliable WSN with the strategy following the arbitrary implementation of nodes in the objective region. An elitism approach dependent GA for streaming pattern recognition is suggested for WSNs in [55]. The technique adjusts operations of GA with an elitism approach into the generations in keeping track of weights. A heterogeneous node placement using multiple-objective adjustable-size GA is presented in [56]. They design wireless transmitters with the algorithm to satisfy specific goals, such as coverage and expense. An easy method to optimum coverage arrangement in WSNs employing effective GA was suggested in [57]. A Monte Carlo technique is additionally used to create a powerful fitness function. In [58], a GA assisted proportionate reasonable resource distribution for multicast OFDM devices is suggested. The technique satisfies various service quality needs in multi-media signals over wireless systems. In [59], a genetic technique for wireless mesh network operation is suggested. Both of these channel allocation and routing issues are resolved through the suggested approach. In [60], an optimum routing and packet arrangement for multihop cellular systems using GA is suggested. They give consideration to a multi-cell circumstance with a nonuniform packet distribution in multihop wireless systems and look for the best topology in a large system. A GA with immigration and recollection strategies for adaptive shortest way routing issue is suggested in [61] for wireless ad hoc systems. They apply the method to MANETs. A GA method for Power-Effective Multicast Routing is suggested in [62]. In that paper, energy-sensitive

multicasting is suggested to reduce the energy consumption. A stability-depending spectrum distribution is suggested in [63] for mobile computing by utilizing GA. The strategy produces an efficient information transmissions performance with regard to the effective connection, terminal range of motion and trusted routing efficiency.

2.3 Swarm Intelligence in Wireless Communications

2.3.1 Introduction to Swarm Intelligence

Swarm intelligence is an uncentered optimization algorithm that was created by Gerardo B. in late 1980s [64]. The motivation derives from biology, particularly natural systems. The elements observe quite easy principles, even though there is no centralized management system controlling how elements ought to perform. Communications among these types of elements result in the appearance of intelligent behavior as a whole. Instances for biological models of SI involve bee colonies, wolves, sheep herding, bat swarms, frog leaping and ant intelligence. Examples of SI are the grey wolf optimizer, the artificial bee colony algorithm (ABCA), particle swarm optimization (PSO), the artificial fish swarm algorithm (AFSA), the bat algorithm, ant colony optimization (ACO), the shuffled frog leaping algorithm (SFLA), etc.

SI has been applied to optimization problems. The techniques are generally made up of a number of basic elements related with each other and with their surroundings. The effective use of SI principles to optimization problems is known as swarm based optimization, while EAs represents a broader group of algorithms.

2.3.2 Ant Colony Optimization

In information technology and processes study, ACO is a strategy for dealing with issues for obtaining better routes in graphs. ACO is part of SI and it was firstly suggested by

Marco Dorigo in 1991[65]. The initial algorithm was looking for a best route in a graph, using the character of ants searching for a route around their nest. The initial concept has since varied to settle broader types of mathematical issues.

Within the biological environment, ants firstly move arbitrarily, and after obtaining food go back to their own nest while setting up pheromone routes. When some other ants discover this kind of route, they will be probably not continue exploring arbitrarily, but alternatively to adhere to the route, moving back and strengthening it when they ultimately obtain food. As time passes, the pheromone route begins to evaporate, thus decreasing its interesting power. The longer time it requires for an ant to search around the path and returning, the more pheromone evaporate. A fast route, in contrast, receives visits more often, and therefore the pheromone density gets increased. Pheromone evaporation has the benefit of staying away from the sub-optimal result. Without evaporation, the routes selected by the initial ants might be extremely encouraging the following. If so, the search area will be restricted. As a result, if a single ant discovers an excellent route through the nest to an eating place, additional ants will probably adhere to that route, and resulting in many of the ants adopting the path. The concept of ACO would be to imitate this specific character.

2.3.3 Modified Chaos Clonal Shuffled Frog Leaping Algorithm

Lately, swarm motivated methods have been confirmed to be successful in exploring best solutions, like SFLA. SFLA has been used to solve a lot of optimization problems. In [106], Eusuff et al. suggested a SFLA for discrete optimization by using a group-based cooperative search. SFLA is motivated through the biological character of the frog [108]. SFLA has been employed for a number of non-linear optimization issues. However, SFLA is easy to fall into premature convergence during the evolutionary process.

In Chapter 3, we develop a new PTS method using a modified chaos clonal shuffled

frog leaping algorithm called MCCSFLA. It is motivated by the biological character of frogs, and inspired by the chaos theory and clonal selection. MCCSFLA combines the nature inspired local search together with the global information exchange between groups and employs clonal selection. With these novel mixed strategies, MCCSFLA can prevent local suboptimal points and lead the search for the global optimum solution. We present detailed algorithm design of MCCSFLA in Chapter 3. The convergence of MCCSFLA is also proved through the Markov chain theory. Extensive simulations are conducted comparing the proposed algorithm with the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm and the original method without PTS. Simulation results demonstrate the superior performance of the proposed MCCSFLA in both PAPR reduction as well as fast convergence.

2.3.4 Quantum Ant Colony Evolutionary Algorithm

In Chapter 4, we develop a novel quantum ant colony evolutionary algorithm (QACEA) to improve the target coverage rate of self-organizing wireless sensor networks. We use the concepts of quantum computation as well as the quantum theory. QACEA is basically based on a number of procedures, including ant and colony encoding, ant initialization, quantum state observation, fitness evaluation, path selection and pheromone update, quantum rotation operation, quantum mutation operation, etc. QACEA uses a set of binary ants and a set of quantum ants within the ant colony, that is a mixture of deterministic and probabilistic illustration and uses both ant investigation and a quantum gate to push the ant colony to the optimum solution. In addition to the colony, we maintain an elite binary ant in each iteration. Since the quantum bit illustration can signify a linear superposition of all possible answers, QACEA features a superior characteristic of diversity than GA and SA.

We designed QACEA for target coverage in self-organizing wireless sensor networks

that can gain a higher target coverage rate and greater quantity of detected targets in contrast to the traditional GA and SA. In the simulation, the result of QACEA is compared to GA and SA for the self-organizing wireless sensor networks with various amounts of sensor nodes and sensing radius. Results demonstrate that the target coverage rate of the suggested QACEA strategy is about 10 percentage and 20 percentage greater than that of GA and SA respectively with different sensing radii. In addition, the amount of targets that are successfully detected by the proposed QACEA is much higher than that of GA and SA. So the proposed method significantly enhanced the monitoring results.

2.3.5 Swarm Intelligence for Wireless Communications

An investigation of the spectrum time distribution according to particle swarm optimization for gigabit multi-media mobile systems is made in [66]. The approach optimizes the period assigned to every single equipment within the system so as to increase the service quality experienced by every single user. In [67], an optimum energy distribution for relevant data fusion in WSNs is suggested based upon restrained particle swarm optimization. They decrease the entire system energy consumption to obtain a required fusion error probability. In [68], a particle swarm optimization approach for contingency transmition taking into account of spectrum conditions is suggested for ad hoc networks. In the work, a decentralized contingency transmition approach for communal spectrum based on an imperfect data game is introduced. A novel timeframe with particle swarm optimization is suggested in [69] for WiMAX systems. The article presented a unique distributor for links with particle swarm optimization which assures a supply of excellent services. A PTS generating strategy based on an ABCA for PAPR reduction is suggested in [70] for multicarrier CDMA devices. In their work, the ABCA strategy can decrease the computational complexity when compared with alternative means in the multicarrier CDMA platforms. Another PAPR reduction method can be found in [71]. In their paper,

a novel approach based on modified ABC is suggested to find a more suitable mixture of phase components which could considerably decrease the computational difficulty for larger sized subblocks while offering lower PAPR simultaneously.

2.4 Artificial Immune Systems in Wireless Communications

2.4.1 Introduction to Artificial Immune Systems

In EAs, artificial immune systems (AIS) are motivated by the concepts and procedures of the biological immune technique. The strategies are motivated by certain immunological concepts that specify the capability and patterns of the vertebrate immune system.

AIS is a sub-area of naturally-motivated artificial intelligence and EAs. The concept of AIS is involved with the framework and characteristic of the immune technique to artificial intelligence. AIS explores these techniques for optimization issues from industrial, commercial and research areas.

AIS was introduced in the mid 1980s by Farmer and Packard [73]. Initially AIS attempted to identify effective simplifications of procedures located in the immune mechanism. However, lately, it has been used for simulating natural systems as well as in implementing immune methods to biological information issues. Recent development on AIS systems, including a threat principle and methods motivated by the natural immune technique, are being investigated. Other improvements include the investigation of degeneration in AIS systems which is inspired by its virtualized function. More specifically, the common techniques of AIS includes the Negative Selection Algorithms (NSA), the Immune Network Algorithms (INA), the Dendritic Cell Algorithms (DCA) and the Clonal Selection Algorithm (CSA), etc.

2.4.2 Negative Selection Algorithms

NSA are motivated by the negative and positive selection procedures, which appear within the growth of the T cellular material. NSA involves the recognition and removal of over reacting cellular material, which can be T cells that assault own cells. These types of methods can be employed for several kinds of optimization problems in which the solution space is made from readily accessible information.

The initial NSA was proposed by Forrest in 1994 to identify information adjustment because of a computer virus in a PC [74]. The beginning point of NSA is to generate a group of self strings S, which outline the regular condition of the program. The work then is to produce a group of sensors D, that just identify the complement of S. These kinds of sensors will then apply to fresh data to categorize them to be self or non-self, mentioning the possibility that data has become altered. Forrest creates the group of sensors through the procedure given below.

input: S = group of self components

output: D = group of produced sensors

begin

start the loop

Randomly produce prospective sensors and put them in a collection P

Identify the affinity of every individual in P with every individual in the set S

If a minimum of one factor in S acknowledges a sensor in P as outlined by a acknowledgement limit,then the sensor is declined, if not it is put into the group of sensor D.

loop end till ending conditions is fulfilled

end

2.4.3 Immune Network Algorithms

INA are motivated by the genetic system principle suggested by Niels K. J [75], which explains the control of the body's defense mechanism through antibodies selection. INA methods include description of networking topology properties in which antibody cells stand for the nodes and the coaching criteria stand for links connecting neighboring nodes.

2.4.4 Dendritic Cell Algorithms

DCA is an instance of a medically motivated technique employing a multi-level strategy. DCA is enforced and applied by way of a procedure for studying and abstracting several characteristics of dendritic cells, through the interactions found inside the cell and the features shown by way of a group of cells in its entirety.

2.4.5 Clonal Selection Algorithm

CSA is a category of methods motivated by the principle of the immune system that shows the way T and B cells enhance their reaction to antigens eventually known as affinity growth. The clonal selection idea is known as a medical concept within immunology which describes the features of lymphocytes reacting to particular antigens infiltrating the physique. Such techniques concentrate on the Darwinian features of the idea in which the clone operation is motivated by cellular division, mutation is motivated by cell variants and selection is motivated by the affinity of antibody-antigen relationships.

The theory was created by Frank M. B. in 1957 to describe the occurrence of the variety of antibodies for the start of the immune system reaction [76]. The earliest experimental proof was available in 1958, during the time Gustav N. and Joshua L. demonstrated that a single B cell generally generates just one antibody [77]. The thought has developed into a broadly approved structure for the way the body's defense mechanism reacts to infection

and the way particular sorts of T and B cells are chosen do defend against damage of particular antigens. The idea claims that within a selection of B cells, a certain antigen just invokes its particular cell to make sure that the specific cell is stimulated to copy for antibody generation. In other words the thought can be description of the working principle for the creation of variety of antibody appearances.

CSA are most frequently used in optimization areas, several of which are similar to random search and GA with no crossover operation.

2.4.6 AIS Algorithms for Wireless Communications

The method in [78] make use of AIS to identify node bad behavior within a movable ad hoc system employing dynamic source routing. They present their solution for the categorization using the artificial immune system. In [79], a method that employs an arbitrary key service based on AIS for discovering spoofing attacks is suggested. The strategy is for clustered WSNs and the technique is performed on the Low-Energy Adaptive Clustering Hierarchy (LEACH) scheme. [80] suggests an AIS design to use in secure routing in wireless mesh networks (WMN). The paper discusses possible threat degrees and presents accountable variables and model threats in WMNs. In WSNs, a research effort for topology management using AIS is presented in [81]. In that work, the approach is employed to resolve the multi-target minimal power system connection issue. [82] studies the channel distribution issue in wireless communications by AIS. The suggested strategy is used for data sets that have been produced according to user needs. [83] suggests a strategy based on GA and AIS for active intrusion recognition in Mobile Ad-Hoc Networks. In the work, a collection of spherical sensors is deployed employing a Niche genetic algorithm to monitor an area. [84] provides a naturally-motivated fuzzy set identification system for discovering bad behaving nodes in an ad-hoc system. The paper also studies the potential of learning bad behaviors.

2.5 Quantum Evolutionary Algorithms in Wireless Communications

2.5.1 Introduction to Quantum Computation

Quantum techniques are probably the biggest accomplishments within the twentieth century. Quantum computing researches quantum devices that employ quantum mechanisms to complete data processing. It includes principles such as quantum machines and quantum strategies.

Quantum computing machines are not the same as traditional machines. While traditional computing machines need information being secured within discrete systems which is actually in one of several particular forms, quantum computing makes use of Q-bits, and this can be in superpositions of several different forms.

Quantum mechanised computing devices had been suggested during the early 80s in the last century, then the outline of quantum technical computing units was proposed. Quantum computing was first proposed by Yuri M., Richard F and David D. in early 1980s [86]. Quantum computing devices have been developed from 1990s. They are demonstrated to be more efficient than traditional computers. They reduce the computing load and make it possible to solve large-scale issues. Recently several research efforts demonstrated that quantum computing procedures can be taken with a small number of Q-bits.

Both theoretical and practiced research continue in order to build quantum computing devices for military, commerce and other purposes. Quantum computation can be used in a variety of ways, for instance, in quantum annealing for optimization.

2.5.2 Quantum Evolutionary Algorithms

Quantum Evolutionary Algorithm (QEA) is a new division of evolutionary algorithms. QEA for a traditional computing device is a branch of research on evolutionary computing which is described as major rules of quantum techniques including uncertainness, superposition, collapse, rotation etc. Based on the idea of quantum computing as well as a mixture of the quantum principle employing the evolutionary concept, Han and Kim suggested the quantum-inspired evolutionary algorithm to further improve traditional EAs [85]. It provides greater stability for searching as well as obtaining superior answers, using a smaller group of individuals, in contrast to the traditional evolutionary algorithms. It needs to be mentioned that even though quantum evolutionary algorithms are founded on the idea of QC, they are not quantum algorithms, but innovative optimization heuristics for a traditional computing device.

QEA with quantum bits provides a superior group variety compared to other forms, as it uses superposition of discrete states. This cuts down on the computational complexity and makes it possible to solve large scaled problems. Similar to numerous evolutionary algorithms, quantum evolutionary algorithms are made up of the description of the populations and the utility operating together with group mechanics. It can avoid a local optimum and premature convergence. In contrast to conventional heuristic strategies, like GA and PSO, QEA is robust, has global convergence and can be used for many combinatorial optimization problems, even though their efficiency may be impacted by these specific issues.

In the past decade, quantum evolutionary algorithms gained plenty of interests and have currently proven their efficiency in contrast to traditional evolutionary algorithms for dealing with complicated issues. QEA was initially presented for dealing with large-scale function optimization issues. Then the algorithm was employed to solve the traveling salesman and the knapsack problem.

2.5.3 Quantum Immune Clonal Evolutionary Algorithm

Based on the quantum theory and the clonal selection theory, we proposed a novel Quantum Immune Clonal Evolutionary Algorithm (QICEA) to improve efficiency of the conventional EA in Chapter 5. QICEA mixes quantum computation and an artificial immune clonal algorithm in which the components of the antibody population are Q-bits. Compared with the conventional repetitive optimization techniques, QICEA with quantum bit illustration can discover the search area using a linear superposition of states. The main steps of QICEA include antibody encoding, initial antibody generation, affinity calculation, antibody selection and clonal mutation operation.

Unlike the traditional QEA, a clonal selection is used in QICEA to look for the antibody with the maximum affinity. We suppose there are S antibodies in the population. Initially all the antibodies are ranked in a decreasing order of affinity. Then the first r antibodies with a greater affinity are chosen for cloning. Then the clonal mutation is conducted on the cloned antibody population with a quantum gate.

We also design a novel QICEA based duty cycle operation in WSNs that can obtain a longer network lifetime while keeping full coverage in the monitoring area. Simulation results show that this QICEA for the duty cycle design issue in WSNs enjoys an extended network lifetime compared to the conventional GA and SA.

2.5.4 Parallel Elite Clonal Quantum Evolutionary Algorithm

In Chapter 7, we develop a novel parallel elite clonal quantum evolutionary algorithm (PECQEA) for QoS routing in multimedia WSNs. We design PECQEA with multiple populations, which mixes the advantages of both the clonal selection algorithm and quantum evolutionary algorithm. It has fast search speed and global search ability in complicated search spaces which is ideal for combinatorial optimization problems. In order to

improve the search effectiveness, a clonal operator is designed in every sub-population of PECQEA. The binary individual with the best fitness while fulfilling all the constraints in each sub-population is selected for cloning. First a fixed quantity of copies of the binary individual are produced, then the chaotic mutation operator is used to every copy, which could efficiently add diversity to the population. Then, we record the binary solution present in sub-population, and employ it in the individual update procedure. With the clonal and the mutation operators, the algorithm can efficiently prevent a local optimum. Furthermore, it can quickly converge towards the optimal area throughout the search process. In addition, PECQEA is flexible and powerful when searching in a large space, as it can keep individual diversity in different populations.

The Markov chain theory is then employed to assess the convergence of PECQEA. First, we demonstrate that the evolution process of PECQEA is a Markov chain. Then we model and study the transition matrix of the improvement procedure for PECQEA. The proof demonstrates PECQEA can converge to the global optimum within limited generations.

Furthermore we develop an objective function to reduce route energy consumption under several restrictions. In order to show the effectiveness of PECQEA, simulations are carried out for the multi-constrained QoS unicast routing problem and performance evaluations are made with GA and ACO. Results demonstrate that PECQEA provides faster convergence and lower energy consumption than existing algorithms.

2.5.5 Quantum Evolution Algorithms for Wireless Communications

[87] suggests an ideal quantum-motivated approach for routing optimization in self-organizing networks. It examines reliable paths using the idea of the Pareto optima at decreased complexity. Efficiency of the proposed algorithm was compared to other EAs, showing

that the proposed method gets near-optimum results. [89] provides a synchronised organization of relay stations and base stations with connection streams for a broadband mobile system. The paper proposed a collection of relay stations and base stations that could assist all clients and meet their needs at the cheapest price. In [90], a topology design algorithm is proposed to maximize the number of active clients served with a minimum number of facilities. The paper suggests a quantum motivated evolutionary algorithm to obtain the best answer for the issue. In [91], quantum inspired evolutionary decoder algorithms are proposed. In the paper, an efficient decoding scheme in Multiple-Input Multiple-Output (MIMO) communicating networks is proposed employing the proposed algorithm.

2.6 Conclusion

In this chapter, we summarize the EAs, and give literature review for EAs in wireless communications. In the next five chapters, we will present five new evolutionary algorithms and apply them for optimization in different scenarios in wireless communications.

Chapter 3

A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems

3.1 Introduction

Orthogonal Frequency Division Multiplexing is a multicarrier technique with high bandwidth efficiency. By dividing the bandwidth into many orthogonal subcarriers, it can minimize the impact of multi-path delay and multipath fading [4]. However, one major problem is the high peak-to-average power ratio, which not only reduces system power efficiency but also results in significant nonlinear distortion when signal passes the amplifier.

In order to reduce PAPR, techniques such as signal scrambling and signal pre-distortion have been proposed. Signal scrambling techniques include coding methods [92], phase optimization [93][94], partial transmission sequence method and selective mapping (SLM). Signal pre-distortion techniques include clipping methods [95]-[97].

The PTS method is a popular technique for PAPR reduction in OFDM systems. However, the partial transmit sequence selection is a highly complex NP-hard problem and the computational complexity is very high for a large number of OFDM subcarriers. Previous studies aimed at finding a set of selected partial transmit sequences with heuristic and evolutionary algorithms. Among the studies, simulated annealing (SA) and particle swarm optimization [98] can achieve lower PAPR than SLM. However, in practice heuristic algorithms suffer from a low convergence rate. The SLM method is proposed in [85] for QAM modulated OFDM signals. It has a lower computational complexity but the PAPR reduction is not as good as those of the heuristic algorithms. Chen et al. tried to solve the PAPR reduction problem with a quantum evolutionary algorithm [100][101], which provides a wider search space. However, quantum evolutionary algorithms also suffer from a low convergence speed. As one of the iteration based approaches, Y. Wang et al. proposed a PAPR reduction method based on Parametric Minimum Cross Entropy for OFDM system [103]. Their method not only reduces PAPR significantly but also decreases the computational complexity. Another useful method using a real-valued genetic approach has been proposed by J.-K. Lain et al. [104] Their design is a similar concept to the genetic algorithm.

Recently, nature inspired approaches have been proved to be very effective in searching for optimal solutions, such as the genetic algorithm [14], the ant colony optimization [102], and the artificial bee colony algorithm [105]. They have been used to solve discrete and continuous non-linear optimization problems. In [106], Eusuff et al. proposed a shuffled frog-leaping algorithm for discrete optimization by using a population-based cooperative search metaphor inspired by natural memetics. Their design is conceptually similar to the genetic algorithm and is effective for solving combinatorial optimization problems. In their method, the worst frog in each group first jumps to the best frog in the same group to create a new frog. If the new frog is better, the new frog will replace the worst frog.

Otherwise, the worst frog will jump to the global best frog. However, their algorithm is easy to fall into premature convergence during the evolutionary process. As a result, it is hard to find a good solution in the billions of possible combinations for the PAPR reduction problem in a limited number of iterations.

In this chapter, we propose a novel PTS method based on a modified chaos clonal shuffled frog leaping algorithm called MCCSFLA. It is inspired by the natural biological behavior of frogs, and motivated by the chaos theory and clonal selection. MCCSFLA combines the nature inspired local search with the global information exchange between groups and takes advantage of clonal selection. With such combined strategies, MCCSFLA is able to avoid local suboptimal points and direct the search toward the global optimum PTS that minimizes PAPR. We present a detailed algorithm design of MCCSFLA for PAPR reduction. The convergence of MCCSFLA is proved through the Markov chain theory, where the MCCSFLA iteration process is modeled with Markov chains. Extensive simulations are conducted comparing the proposed algorithm with the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm and the original method without PTS. Simulation results demonstrate the superior performance of the proposed MCCSFLA in both PAPR reduction as well as fast convergence.

This chapter is organized as follows. The system model is given in Section 3.2. In Section 3.3, the modified chaos clonal shuffled frog leaping algorithm for PAPR reduction is presented. Section 3.4 analyzes the convergence of the proposed algorithm with Markov chain theory. In Section 3.5 the simulation results are presented and discussed. Finally, Section 3.6 draws the conclusions.

3.2 System Model

This section describes the system model for PAPR reduction in OFDM systems. In [100], the authors proposed an OFDM PAPR reduction model with binary Signal Sign-Selection from the set $\{-1, 1\}$. Later in [101], the same authors showed a more flexible model with signal sign-selection from the set $\{1, -1, i, -i\}$. In order to facilitate performance comparisons between different PAPR reduction algorithms, in this chapter we consider a similar system model as in [101]. Assume an OFDM system with L subcarriers. The discrete time transmitted signal can be represented as:

$$x_n = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} Z_l e^{i2\pi nl(\frac{1}{L})} \tag{3.1}$$

where n is the discrete time index, i equal to $\sqrt{-1}$, and $Z = [\ Z_0 \quad Z_1 \quad \cdots \quad Z_{L-1}\ ]$ is the input symbol sequence.

As defined in [100], the PAPR of the transmitted signal can be represented as:

$$PAPR = 10\log_{10} \frac{\max\left\{|x_n|^2\right\}}{E\left\{|x_n|^2\right\}} \tag{3.2}$$

where E is the expected value operation.

Then the input data $Z = [\ Z_0 \quad Z_1 \quad \cdots \quad Z_{L-1}\ ]$ can be divided in to V non-overlapping sub-blocks $\{Y_v, v = 0, 1, \cdots, V-1\}$, which can be shown as

$$Y = \left[\ Y_0 \quad Y_1 \quad \cdots \quad Y_{V-1}\ \right]. \tag{3.3}$$

The objective of the PTS method is to generate an appropriate phase weighting sequence that reduces the PAPR. For dividing the input data into V non-overlapping sub-blocks, we follow the criteria presented in [107].

The phase weighting sequence is a vector with length V, which can be represented as:

$$D = [\ d_0 \quad d_1 \quad \cdots \quad d_{V-1}\ ] \tag{3.4}$$

where $d_v = \exp(j\varphi_v)$ is the phase weighting factor and $\{\varphi_v, v = 0, 1, \cdots, V-1\}$ are phase factors selected from the range $\varphi_v \in [0, 2\pi)$. However in practice the phase factors are selected from a limited set, which can be represented as:

$$\varphi_v \in \{2\pi\omega/W | \omega = 0, 1, \cdots W-1\} \tag{3.5}$$

where W is the set of permitted phase factors. In this chapter we only consider $\omega = 0, 1, 2, 3$, which means $d_v \in \{1, i, -1, -i\}$.

After selecting a proper phase weighting factor, it is multiplied by the input data to reduce PAPR, which can be represented as:

$$Y' = [\ d_0 Y_0 \quad d_1 Y_1 \quad \cdots \quad d_{V-1} Y_{V-1}\ ]. \tag{3.6}$$

After the phase weighting factor optimization, the discrete time transmitted signal can be represented as ${x_n}'(D)$.

So the objective function of the PAPR reduction problem is equivalent to the phase factors search problem, which can be expressed as:

Minimize

$$f(D) = \frac{\max\left\{|{x_n}'(D)|^2\right\}}{E\left\{|{x_n}'(D)|^2\right\}} \tag{3.7}$$

subject to

$$\varphi_v \in \{2\pi\omega/W | \omega = 0, 1, \cdots W-1\}. \tag{3.8}$$

The goal of our algorithm is to minimize the fitness function $f(D)$. As each OFDM symbol has V sub-blocks, and each phase factor is selected from the set φ_v, the solution space is W^V. As changing a common angle on the sub-blocks cannot change PAPR, we can set φ_0 to a fixed value, so the solution space can be reduced to W^{V-1}.

Sometimes we have to make a balance between computing power and bandwidth. For example, with sub-blocks of $V = 8$ and $W = 2$, the possible combination of the phase weighting sequence is 2^7=128. In this case exhaustive search can be used to obtain the

optimal PTS. However, with the length of the phase weighting sequence $V = 16$ and QPSK, which are the parameter settings considered in this chapter, the possible combination of the phase weighting sequence is 4^{15}=1,073,741,824. In this case, exhaustive search is not possible in real time. As such, some lower complexity schemes must be used to reduce the computational complexity. The common approach is to design low complexity algorithms to obtain sub-optimal PTS.

In order to obtain the original signal, the receiver must obtain the PTS phase information from the transmitter. In practical systems, the side information can be transmitted via control channels. In some other systems, there can be reserved sub-carriers for side information. In this chapter, we assume the side information is transmitted to the receiver via one of the above two methods.

3.3 PAPR Reduction Based On MCCSFLA

To select a proper PTS, we propose a phase optimization scheme MCCSFLA for PAPR reduction. Inspired by the natural population of frogs, MCCSFLA combines a local search with global information exchange among groups and makes use of the advantages of clonal selection. This balanced strategy enables MCCSFLA to avoid local suboptimal points and direct the search towards a low PAPR solution. In our proposed MCCSFLA, the key improvement is that a clone selection operator is added to the traditional SFLA algorithm. By cloning the best individual and carrying out mutation to each copy, the clone selection operator can significantly improve convergence speed of the MCCSFLA algorithm.

In this section, we present the design of MCCSFLA. We first review the basic principle of the original shuffled frog leaping algorithm and explain encoding and population representation. We then proceed to the main parts of the algorithm in terms of initialization,

sorting and grouping, searching, clonal selection and shuffling, as well as the termination condition. We present MCCSFLA in Algorithm 1, followed by its complexity analysis.

3.3.1 Brief Description of Basic SFLA

The shuffled frog leaping algorithm (SFLA) is inspired from the natural behavior of the frog [108]. SFLA has been used for a group of discrete and continuous non-linear optimization problems. In SFLA, the population is partitioned into different groups of frogs and each group consists of a fixed number of frogs. Each frog is considered as a solution in the process of evolution and frogs in one group can be influenced by frogs in another group by the means of a shuffling process. In SFLA, the information is carried by a meme, which is similar to the gene in genetic algorithms. Groups of memes are called meme complexes, or "memeplexes" and the evolution process of SFLA is also called a memetic evolution. During a memetic evolution, a frog can improve its memes by leaping towards another frog with a better fitness. After the initialization, each group of frogs conducts a local search. Within each group or memeplex, frogs can exchange information with each other but only the worst frog can jump to another position. After each memeplex finishes the local search, a shuffling strategy is applied to all memeplexes by ordering and then all the frogs are reorganized into new memeplexes according to their fitness. Both local search and shuffling process are repeated until the termination condition is met.

3.3.2 Solution Encoding and Population Representations

Assume that there are M memeplexes in the entire population, and each memeplex contains N frogs. The whole frog population is represented as $P = \left\{ \begin{array}{cccc} F_1 & F_2 & \cdots & F_M \end{array} \right\}$, where F_m is the m_{th} memeplex, and $m \in [1, M]$ is the order number of the memeplex in the population. Each memeplex is represented as $F_m = \left\{ \begin{array}{cccc} D_1 & D_2 & \cdots & D_N \end{array} \right\}$, where D_n is the n_{th} frog in the m_{th} memeplex, and $n \in [1, N]$ is the frog index. Assume an

OFDM system with V non-overlapping sub-blocks, the n_{th} frog is represented by matrix $D_n = \begin{bmatrix} d_1 & d_2 & \cdots & d_{V-1} \end{bmatrix}$, where $d_v \in \{1, i, -1, -i\}$ and $v \in [1, V-1]$. In this chapter, we set the maximum number of iterations of the local search to 10.

3.3.3 Generation of Initial Population with a Logistic Map

Before the first iteration, initial memeplexes should be generated. MCCSFLA uses a Logistic map to generate each frog in each memeplexes. The Logistic map is a polynomial map with low complexity and chaotic behavior, which was first proposed in [109]. We first generate a random number between 0 and 1 with a Logistic map as:

$$x_{v+1} = 4x_v(1 - x_v). \tag{3.9}$$

For an OFDM system with V non-overlapping sub-blocks, a feasible solution space for the problem can be represented as $D_n = \left\{ \begin{array}{cccc} d_1 & d_2 & \cdots & d_{V-1} \end{array} \right\}$. So for each frog, there are four options for each meme. We use a simple map to fix d_v as:

$$d_v = \begin{cases} 1 & 0 < x_v < 0.25 \\ -1 & 0.25 \leq x_v < 0.5 \\ i & 0.5 \leq x_v < 0.75 \\ -i & 0.75 \leq x_v < 1 \end{cases} \tag{3.10}$$

where d_v are the phase weighting factors from the set $\{1, -1, i, -i\}$, and x_v is the chaotic sequence generated from (3.9). MCCSFLA also needs to set the global iteration counter $Gc = 0$ before the iteration starts.

3.3.4 Fitness Calculation

In this chapter, we compute fitness $f(D)$ with (3.7) for each frog. If the value of $f(D)$ is smaller, the frog is better.

3.3.5 Shuffling

Before the search process in each iteration, MCCSFLA does the shuffling process for the whole population. First all the frogs in different memeplexes are merged into one population, followed by sorting and grouping. The shuffling process helps the frogs exchange their information among different memeplexes and promotes the algorithm convergence to the global optimum.

In the sorting and grouping process, we sort the frogs in a descending order according to the fitness value calculated by (3.7) and then partition the generated frogs into M memeplexes. The process of partitioning is as follows:

Suppose there are M memeplexes represented as $\left\{ \begin{array}{cccc} F_1 & F_2 & \cdots & F_M \end{array} \right\}$ and each memeplex contains N frogs, so there are $M \times N$ frogs in the entire population. At the beginning, we put the No. 1 frog into F_1, and put the No. 2 frog into F_2, etc. After we put the No. M frog into F_M we then put the No. $M+1$ frog back to F_1, then put the No. $M+2$ frog to F_2, and so on until the $(M \times N)_{th}$ frog goes to F_M. Through this operation, within each memeplex, the frog ranked first has the largest fitness and PAPR, and the frog ranked last has least fitness and PAPR. That is, within each memeplex, the last frog is the best one. Also, we record the last frog in the last memeplex as the global best frog D_g.

3.3.6 Local Search

After the initialization, MCCSFLA starts a local search process to improve the quality of the frog fitness. Each memeplex conducts a local search independently according to a specific strategy. Within each memeplex, we record the frog ranked first which has the largest fitness as the worst frog D_w and the frog ranked last which has the least fitness as the best frog D_b. The fitness of the worst frog and the best frog in the memeplex can

be shown as $f(D_w)$ and $f(D_b)$. Before we enter the local search, we set the local search counter $Lc = 0$. Then the worst frog in each memeplex is updated in the following steps:

1) Jump Toward the Local Best

First, the worst frog in the memeplex D_w changes its position and jumps toward the best frog in the memeplex D_b. Unlike the traditional SFLA, we do not set maximum and minimum jump distance limitations. The meme of the worst frog D_w is changed as follows:

$$D_{sub} = rand \cdot (D_b - D_w) \tag{3.11}$$

$$D^1_{new} = D_w + D_{sub} \tag{3.12}$$

where in (3.11) $rand$ is a binary sequence with length $V - 1$ generated by the Logistic map. In sequence $rand$, if the result of the Logistic map is smaller than 0.5, the number on the corresponding position is equal to 0. Otherwise it is equal to 1. In this way, D_{sub} is the jump sequence with length $V - 1$. Then we calculate the fitness of D^1_{new}. If the fitness of a newly generated frog $f(D^1_{new})$ is smaller than $f(D_w)$, which means the PAPR of the solution D^1_{new} is lower, the algorithm replaces D_w with D^1_{new} and goes to step *5)*. Otherwise the algorithm goes to step *2)*.

2) Jump Toward the Global Best

Replace D_b in (3.11) with D_g, and repeat the procedure *1)*. D_g is the global best frog defined in section E. So (3.11) can be renewed as:

$$D_{sub} = rand \cdot (D_g - D_w) \tag{3.13}$$

and then a new frog is generated with:

$$D^2_{new} = D_w + D_{sub}. \tag{3.14}$$

We next evaluate the fitness of D^2_{new} with (3.7). If the fitness of the newly generated frog D^2_{new} is better than D_w, the algorithm replaces D_w with D^2_{new} and goes to step *5)*. If there is still no improvement in fitness, the algorithm goes to step *3)*.

3) Clonal Selection

Do the clonal selection with the clonal selection operator. The detailed procedure is described in section 3.3.6.

4) Replace with Random Frog

If there is still no improvement, we generate a random frog D^3_{new} with the Logistic map to replace the worst frog D_w in the memeplex. We generate the frog with the same rule as we generate the initial population of frogs but this time we generate only one frog. After completion of the replacement, the algorithm goes to step *5)*.

5) Reorder

Reorder all the frogs in the memeplex in a descending order according to the fitness value, and set $Lc = Lc+1$. By doing this a round of local search is completed. In this procedure, if any frog's fitness is better than the global best frog D_g, it replaces D_g.

The above procedure continues until Lc reaches the maximum number of the local search iterations. Each memeplex repeats a certain round of local searches independently for a specific number of generations.

The local search makes a memetic to the exploitation capability of the algorithm by making the information pass in the local space, and it improves the average fitness of the memeplexes.

3.3.7 The Clonal Selection Operator

Unlike the traditional SFLA, a chaotic clonal selection operator is applied in MCCSFLA to enhance the efficiency of the local search. Before generating a random frog solution in the local search step, a number of copies of the best solutions are made, and then the frog jumps to the position where PAPR goes down. The main steps of the clonal selection operator are: cloning the best solution, chaotic mutation, frog jump and fitness calculation.

1) Cloning the Best Solution

The local best frog and the global best frog were chosen for cloning. Both the local best frog and global best frog are cloned to a fixed number of copies. Unlike the traditional clonal selection algorithm, the number of copies has no relationship to the fitness.

2) Chaotic Mutation

The chaotic mutation operator randomly chooses some memes on the frog with a fixed rate and replaces them with the value from the set $\{1, -1, i, -i\}$. The chaotic mutation operator can add additional diversity to the cloned frog and prevent premature convergence on the offspring.

3) Frog Jump

We create new frogs with (3.15) and (3.16):

$$D_{sub} = rand \cdot (D_c - D_w) \tag{3.15}$$

$$D^4_{new} = D_w + D_{sub} \tag{3.16}$$

where in (3.15), D_c is the cloned frog which passed the chaotic mutation operation. The other parameters are the same as in Section 3.3.6. After that, these frogs are evaluated

and ranked in the descending order of fitness. If the fitness is improved, the frog ranked with the lowest PAPR will replace the worst frog in the memeplex. Otherwise the frog will be replaced with a random frog as in step *4)* in Section 3.3.6.

3.3.8 Population Mutation

In order to increase the diversity, a population mutation operator is applied to the whole population. The population mutation is the last operation in each iteration. In this procedure we randomly select a very small percentage of memes in the whole population and replace these memes with random values from the set $\{1, -1, i, -i\}$.

3.3.9 Upgrade the Elite Frog

We keep an elite frog D_e which has the lowest PAPR in all the generations of MCCSFLA. If the frog with the lowest PAPR after population mutation D_m has lower PAPR than D_e, we replace all values of D_e with the corresponding values of D_m.

3.3.10 Termination Condition

After shuffling, MCCSFLA will check whether the termination condition is satisfied. The termination condition is when the algorithm global iteration counter Gc reaches the designated number of iterations.

3.3.11 Basic Steps

The basic steps of MCCSFLA are described in Algorithm 1:

Algorithm 1 The modified chaos clonal shuffled frog leaping algorithm based PTS method

Begin
Generate the initial population with Logistic map
Evaluate the fitness value
Set $Gc = 0$
while Gc has not reached designated global iterations **do**
  Sort and group the population into memeplexes
  Set $Lc = 0$
  while Lc has not reached designated global iterations **do**
    Frog jumps to the local best frog
    if no improvement **then**
      Frog jumps to the global best frog
    end if
    if no improvement **then**
      Clonal selection operator
    end if
    if no improvement **then**
      Replace with a random frog
    end if
    Reorder all the frogs and set $Lc = Lc + 1$
  end while
  Evaluate the fitness value
  Shuffle different memeplex into one population
  Population Mutation
  Upgrade the Elite Frog
  $Gc = Gc + 1$
end while
Evaluate the fitness value
Output the best frog with lowest PAPR
End

3.3.12 Computational Complexity Analysis

As the computing power is mainly consumed in the IFFT operation and PAPR calculation, we only consider the computational complexity of these two parts. There are V sub-blocks and each sub-block need to be modulated with a L-point IFFT. Thus $(LV\log_2 L)/2$ complex multiplications and $LV\log_2 L$ complex additions are needed for IFFT. As the computational complexity of one complex multiplication equals to four real multiplications and two real additions and one complex addition equals to two real additions, the IFFT operation needs $2LV\log_2 L$ real multiplications and $3LV\log_2 L$ real additions.

For each sample of phase weighting sequence, we need $L(V-1)$ complex additions to generate the sample and L real additions and $2L$ multiplications to calculate the PAPR[111]. For conventional PTS, the number of sample is W^{V-1}. So the numbers of real multiplications and real additions for conventional PTS (CON-PTS) are

$$CP_{mul} = 2W^{V-1}L + 2LV\log_2 L \tag{3.17}$$

$$CP_{add} = W^{V-1}L + 2W^{V-1}(V-1)L + 3LV\log_2 L \tag{3.18}$$

For GA-PTS, the samples number equals to $S_1 = Pop1 \times Gen1$, where $Pop1$ is the number of individuals in the population and $Gen1$ is the maximum number of generations in GA-PTS. So the number of real multiplications and real additions for GA-PTS are

$$GA_{mul} = 2LS_1 + 2LV\log_2 L \tag{3.19}$$

$$GA_{add} = LS_1 + 2(V-1)LS_1 + 3LV\log_2 L \tag{3.20}$$

For QEA-PTS, the samples number equals to $S_2 = Pop2 \times Gen2$, where $Pop2$ is the number of individuals in the population and $Gen2$ is the maximum number of generation in QEA-PTS. So the numbers of real multiplications and real additions for QEA-PTS are

$$QEA_{mul} = 2LS_2 + 2LV\log_2 L \tag{3.21}$$

$$QEA_{add} = LS_2 + 2(V-1)LS_2 + 3LV\log_2 L \tag{3.22}$$

For MCCSFLA-PTS, as only the worst frog in each group (memeplex) is renewed, the samples number equals to the $S_3 = AJ \times M \times Gen3$, where $Gen3$ is the maximum number of generations in MCCSFLA and M is the number of groups in MCCSFLA. AJ is the average number of jumps for the worst frog in each group. If a better solution can be found via jumping toward the local best, the number of jumps is 1. If a better solution can be found via jumping toward the global best, the number of jumps is 2. If a better solution can be found via clonal selection, the number of jumps is equal to the number of clones plus 2. Otherwise the number of jumps is equal to the number of clones plus 3, as we need replace the worst frog in the group with a random frog.

$$MCCSFLA_{mul} = 2LS_3 + 2LV\log_2 L \tag{3.23}$$

$$MCCSFLA_{add} = LS_3 + 2(V-1)LS_3 + 3LV\log_2 L \tag{3.24}$$

For SLM, the samples number equals to the number of distinct sign sequences S_4. So the number of real multiplications and real additions for GA-PTS are

$$SLM_{mul} = 2LS_4 + 2LV\log_2 L \tag{3.25}$$

$$SLM_{add} = LS_4 + 2(V-1)LS_4 + 3LV\log_2 L \tag{3.26}$$

3.4 Convergence Analysis of MCCSFLA for PAPR Reduction in OFDM Systems

In this section we analyse the convergence of MCCSFLA for OFDM systems. We first model the iteration process of MCCSFLA as a Markov chain [110], and derive its properties. The base Markov chain is then extended to include the elite frogs, and the properties of such an extended Markov chain are investigated further. Under these properties, we prove MCCSFLA converges to the global optimum.

3.4.1 The Base Markov Chain

First, we give some basic definitions for Markov chains. A discrete-time Markov chain is a random process that can only take values from a discrete state space. In a Markov chain, the next state only depends on the current state and not on the previous states [112]. A Markov chain is homogeneous if it is irrelevant to time (time-invariant). In a homogenous Markov chain, a transition matrix can be represented as a two-dimensional matrix $P = [p_{ij}]$. With the initial distribution $\vec{\pi}(0)$, the probability distribution of the homogenous Markov chain in the time n only depends on $\vec{\pi}(0)$ and P. Some homogenous Markov chain can be Ergodic, which is both irreducible and aperiodic.

In the iteration process of MCCSFLA, the population renews from one iteration to another. If we use standard Markov tools to analyze this evolutionary process, MCCSFLA runs from one population distribution at one iteration to another at the next generation, and this can be viewed as a random process, and the population distribution at each iteration may be considered as a state. As the evolution process of the elite frog D_e is independent from the other frogs, first we just consider the evolution process without elite frog D_e.

Lemma 1: The population sequence $\{Y(n)\}_{n=1}^{\infty}$ *of MCCSFLA without elite frog* D_e *constitutes a Markov chain.*

Proof: Let $Y(n)$ be the whole frog population at iteration n with multiple memeplexes. From the steps of MCCSFLA, we know that the population $Y(n+1)$ is obtained from the population $Y(n)$ with a sequence of operations, which means that the distribution probability of the next iteration has nothing to do with the former iteration $Y(n-1)$ and initial population $Y(0)$. Thus, MCCSFLA can be modelled with a Markov chain and its character may be studied by the Markov chain theory.

In the analysis procedure, we use a binary string to represent the whole population in one state [113]. In this way, each gene in a frog is mapped to two binary bits, and

the number of bits for each frog is $2(V-1)$. As there are M memeplex in population and N frogs in each memeplex, the whole population in state n is mapped to $Y(n) = \{\ D_1,\ \cdots,\ D_{MN}\ \} = \{\underbrace{x_1 \cdots x_{2(V-1)}}_{D_1} \cdots \underbrace{x_{2(MN-1)(V-1)+1} \cdots x_{2MN(V-1)}}_{D_{MN}}\}$. For each frog, the number of possible strings for individual space is $2^{2(V-1)}$, so the solution space for each generation is $|\Phi| = 2^{2(V-1)MN}$, so the state space of the Markov chain is finite.

From the operation steps of MCCSFLA, we can obtain the following conclusions.

Lemma 2: The Markov chain of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is time homogeneous.

Proof: Because the transition probability of the operation on each iteration remains fixed, the Markov chain of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is homogeneous, and the transition probability $P_{ij}^{(n)}$ is irrelevant to step n, which can be denoted as $P = [p_{ij}]$ with size $2^{2(V-1)MN} \times 2^{2(V-1)MN}$. So the Theorem is proved.

Now we focus on some important properties of the operations in MCCSFLA. Each iteration process of the population sequence $\{Y(n)\}_{n=1}^{\infty}$ is divided into two steps. Step 1 is the operations before the population mutation and the corresponding line numbers in flowchart **Algorithm 1** are from 6 to 22. We use the transition matrix G to represent the first step. Step 2 is the population mutations and the corresponding line number in flowchart **Algorithm 1** is 23. We use the transition matrix C to represent the second step. We do not consider the elite operation in line 24 in the base Markov chain. So the whole transition matrix P is the product of the transition matrices in two steps.

Lemma 3: The transition matrix G of step 1 for the population sequence $\{Y(n)\}_{n=1}^{\infty}$ in MCCSFLA is stochastic.

Proof: Mathematically, these operations map probabilistically from one state to another in space Φ with transition matrix $G = [g_{ij}]$. Due to the length of the binary-coded frog being $2^{2((V-1)MN)}$, which is unchanged in the whole process of algorithm, the total probability of transiting from one state to all the states that is from $[0, 0, \cdots, 0]_{1\times 2^{2((V-1)MN)}}$ to

$[1, 1, \cdots, 1]_{1\times 2^{2((V-1)MN)}}$ in the next iteration is 1. Thus $g_{ij} \geq 0$ and $\sum_{j=1}^{2^{2(V-1)MN}} g_{ij} = 1$ for any $i \in [1, 2^{2(V-1)MN}]$. Therefore, the transition matrix before the mutation operation is stochastic.

Lemma 4: The state transition matrix C of step 2 for the population sequence $\{Y(n)\}_{n=1}^{\infty}$ *is both stochastic and positive.*

Proof: The population mutation operation in step 2 randomly maps from one state to another and it works on each bit string independently. Let $C = [c_{ij}]$ be the state transition matrix of the population mutation operation and $Pc \in (0, 1)$ be the mutation probability on each bit. The transition probability can be shown as

$$c_{ij} = Pc^{H_{ij}}(1 - Pc)^{2(V-1)MN-H_{ij}} > 0 \tag{3.27}$$

where c_{ij} is transition probability and H_{ij} is the Hamming distance between state i and state j [114]. As all c_{ij} are positive, the state transition matrix C is positive. From the above formula we can also get $\sum_{j=1}^{2^{2(V-1)MN}} m_{ij} = 1$ for any $i \in [1, 2^{2(V-1)MN}]$. Thus, C is stochastic.

From the above two lemmas, we can get the following conclusions.

Lemma 5: Let G be the transition matrix in Step 1 and C be the state transition matrix of Step 2 in MCCSFLA. Then the product GC is a positive and stochastic matrix.

Proof: Let transition matrix $P = GC$. So we have

$$P= \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,2^{2(V-1)MN}} \\ p_{21} & p_{22} & \cdots & p_{2,2^{2(V-1)MN}} \\ \vdots & \vdots & \ddots & \vdots \\ p_{2^{2(V-1)MN},1} & p_{2^{2(V-1)MN},2} & \cdots & p_{2^{2(V-1)MN},2^{2(V-1)MN}} \end{bmatrix} \tag{3.28}$$

Since for any j in stochastic matrix G, we have $\sum_{j=1}^{2^{2(V-1)MN}} g_{ij} = 1$, so there is at least

one positive element in each row in matrix G. As C is a stochastic positive matrix, for all i and j $c_{ij} > 0$, so for all i and j $p_{ij} = \sum_{k=1}^{2^{2(V-1)MN}} g_{ik} \cdot c_{kj} > 0$. Thus G is a positive and stochastic matrix.

Definition 1: [99] A Markov chain is called an ergodic chain if it is possible to go from every state to every state (not necessarily in one move).

Theorem 1: The population sequence $\{Y(n)\}_{n=1}^{\infty}$ *of MCCSFLA is an ergodic Markov chain.*

Proof: According to Lemma 5 $p_{ij} > 0$ for any i and j, which means that it is possible to reach every state from every state in just one move. As the solution space $|\Phi| = 2^{2(V-1)MN}$ for each generation is finite, according to Definition 1, the population sequence of MCCSFLA is an ergodic Markov chain.

According to the properties of the ergodic Markov chain, the population sequence $\{Y(n)\}_{n=1}^{\infty}$ has a stationary probability distribution $P^{\infty} = e' \cdot p^{\infty}$ when $n \to \infty$, where $p^{\infty} = (p_1, p_2, \cdots, p_m)$ and $\sum_{l=1}^{m} p_l = 1$. As transition matrix $P > 0$, according to the definition of the primitive matrix, P is also primitive.

3.4.2 The Extended Elite Markov Chain

Since in MCCSFLA we keep the elite frog D_e with the lowest PAPR found by the past and current generations, we can enlarge the population by adding D_e in front of other frogs. In this way, the new population at generation n can be expressed as $Y^{+}(n) = \{\ D_e,\ \ D_1 \cdots,\ \ D_{MN}\ \} = \{\underbrace{x_1 \cdots x_{2(V-1)}}_{D_e} \cdots \underbrace{x_{2MN(V-1)+1} \cdots x_{2(MN+1)(V-1)}}_{D_{MN}}\}$.

Lemma 6: The stochastic process of the enlarged population sequence $\{Y^{+}(n)\}_{n=1}^{\infty}$ *of MCCSFLA constitutes a Markov chain.*

Proof: Let $Y^{+}(n)$ be the whole frog population at iteration n with multiple memeplexes. From the steps of MCCSFLA, we know that population $Y^{+}(n+1)$ is obtained

from population $Y^+(n)$ with a sequence of operations, which means that the distribution probability of the next iteration has nothing to do with former iteration $Y^+(n-1)$. Thus, MCCSFLA can be modelled with a Markov chain and its properties may be studied by the Markov chain theory.

For each frog in population $Y^+(n)$, the number of possible states for individual space is $2^{2(V-1)}$ and with the best frog the total number of frogs in the population is $MN+1$, so the solution space for each generation is $|\phi'| = 2^{2(V-1)(MN+1)}$, so the state space of the Markov chain is also finite.

We divide the evolution process of population $Y^+(n)$ into two steps. We use transition matrix $B(n)$ to indicate the whole state transition matrix of the two steps. In the first step, all frogs will be renewed except the elite frog D_e. In the algorithm, this stage includes all the steps in the outermost loop before upgrading the elite frog and the corresponding line numbers in flowchart **Algorithm 1** are from 6 to 23. Mathematically, population $Y^+(n)$ will upgrade from $Y^+(n) = \{D_e, D_1, \cdots, D_{MN}\}$ to $Y^+(n) = \{D_e, D_1^{new}, \cdots, D_{MN}^{new}\}$. We use transition matrix $Q(n)$ to indicate this step. In the second step, the elite frog D_e will be upgraded, and the corresponding line number in flowchart **Algorithm 1** is 24. If the frog with the lowest PAPR in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ has lower PAPR than D_e, we replace all the values of D_e with this frog. We used the transition matrix $U(n)$ to represent this upgrade stage.

Lemma 7: The enlarged population sequence $\{Y^+(n)\}_{n=1}^{\infty}$ *of MCCSFLA constitutes a time homogeneous Markov chain.*

Proof: In the first stage, all the frogs are renewed except the elite frog D_e. We have proved this process is time homogeneous in Lemma 2, so the transition matrix $Q(n)$ can be written as Q. In the second stage, the upgrade only depends on the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ in the current generation. So the upgrade is irrelevant to step n and the transition matrix $U(n)$ can be represented as U. So the enlarged population sequence

$\{Y^+(n)\}_{n=1}^{\infty}$ of MCCSFLA is time homogeneous and the transition matrix $B(n)$ can be represented as B. So the whole transition matrix B is the product of the transition matrices Q and U in two steps.

Theorem 2: The transition matrix B of population sequence $Y^+(n)$ in MCCSFLA is stochastic.

Proof: Mathematically, these operations map probabilistically from one state to another in solution space $|\phi'| = 2^{2(V-1)(MN+1)}$ with transition matrix B and the total probability from one state to all the states in the next iteration is 1. Thus, if we use $B = [b_{ij}]_{2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}}$ to represent the transition matrix $B(n)$, we can get $\sum_{j=1}^{2^{2(V-1)(MN+1)}} b_{ij} = 1$ for any $i \in \left[1, 2^{2(V-1)(MN+1)}\right]$. Therefore, the transition matrix before the mutation operation is stochastic.

To simplify the description, we encode the state number in the solution space according to the PAPR of elite frog D_e. $Y^+(n)$ with better D_e has the higher position in the solution space and the states with the same D_e are listed together. In other words, D_e with lower PAPR will be listed higher.

To make the discussion easier, we assume the global best frog can be represented as D_{best} and it has the lowest PAPR. As the size of matrix B is $2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}$, the states with the elite frog equal to D_{best} are listed with the state number $i \in [1, 2^{2(V-1)MN}]$. Similarly, the states with the elite frog with the highest PAPR are listed with the state number $i \in \left[2^{2(V-1)MN}(2^{2(V-1)} - 1) + 1, 2^{2(V-1)(MN+1)}\right]$.

In the first stage, since the elite frog D_e is not affected by transition matrix P in MCCSFLA, state transition matrix Q can be represented as

$$Q = \begin{pmatrix} P & & & \\ & P & & \\ & & \ddots & \\ & & & P \end{pmatrix} \tag{3.29}$$

where in state transition matrix Q there are $2^{2(V-1)}$ matrices P on the diagonal and the size of each P is $2^{2(V-1)MN} \times 2^{2(V-1)MN}$.

In the second stage, as we need to upgrade the elite frog D_e with the best frog found in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$, we need to define the upgrade transition matrix $U = [u_{ij}]$. The upgraded transition matrix simply replaces the elite frog D_e with the best frog D_m found after the population mutation in the current generation if the PAPR of D_m is lower than D_e, or otherwise does nothing.

Lemma 8: Upgraded transition matrix U is a lower triangular matrix.

Proof: Suppose D_m is the best frog found in the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ after the population mutation at generation n, and D_e is the elite frog at generation $n-1$.

(1) For $Y^+(n)$ with state number i in generation n, if the PAPR of D_m is lower than D_e, then D_m will replace D_e in generation n and the state will transtion from state i to state j. According to the sequence of the state number, state j has a higher position in the solution space, so $i > j$ and $u_{ij} = 1$. In this case, the other elements in row i in the upgraded transition matrix U are 0.

(2) For state $Y^+(n)$ with state number i in generation n, if the PAPR of D_m is higher than D_e, then state i will not change. In this case, $u_{ii} = 1$ and the other elements in row i in the upgrade transition matrix U are 0.

The above results can directly prove upgraded transition matrix U is a lower triangular matrix.

We split U into matrix blocks with the same size as matrix P. In this way, U can be

expressed as:

$$U = \begin{pmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{2^{2(V-1)},1} & U_{2^{2(V-1)},2} & \cdots & U_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix} \tag{3.30}$$

Lemma 9: U_{11} *is a unit matrix.*

Proof: As the states with the elite frog equal to D_{best} are listed with state number $i \in [1, 2^{2(V-1)MN}]$, the elite frog in these states will not be replaced by other frogs, as it has the lowest PAPR in the whole solution space.

In matrix U, each row represents a current state and in such a state the frog with the lowest PAPR is fixed, so the next state is also fixed. With the upgraded matrix, the next state will either go up with a smaller state number or remain in the current state. With the above analysis, we can immediately draw the following conclusions: In each row there is only one $u_{ij} = 1$, other elements equal to 0, and $i \geq j$.

Lemma 10: $U_{21} \neq 0$ *and* $U_{22} \neq 0$.

Proof: Let $Y^+(n) = \{D_e, D_1^{new}, \cdots, D_{MN}^{new}\}$. Since the relevant rows of transition matrix U_{21} and U_{22} contain all the binary combinations of $\{D_1^{new}, \cdots, D_{MN}^{new}\}$, in some rows the set $\{D_1^{new}, \cdots, D_{MN}^{new}\}$ contains D_{best}, for example $D_1^{new} = D_{best}$, in this case the elite frog D_e will be replaced by D_1^{new}. So there are some elements in U_{21} with the form $u_{i1} = 1$. So $U_{21} \neq 0$. In some rows D_e has the lowest PAPR, so the status will not be changed. In this case $u_{ii} = 1$. So $U_{22} \neq 0$.

In this way, as U_{11} is a unit matrix, the total transition matrix can be expressed as

[116]

$$B = QU = \begin{pmatrix} P & & & \\ PU_{21} & PU_{22} & & \\ \vdots & \vdots & \ddots & \\ PU_{2^{2(V-1)},1} & PU_{2^{2(V-1)},2} & \cdots & PU_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix} \tag{3.31}$$

3.4.3 The Convergence for MCCSFLA

Before the convergence proof, we need to define the concept of convergence of MCCSFLA for PAPR reduction.

Definition 2: Let D_e be the elite frog found by MCCSFLA at generation n, and D_{best} be the best frog with lowest PAPR f_{best} in the whole population space Φ. If $\lim_{n\to\infty} \Pr(f(D_e) = f_{best}) = 1$, then population sequence $\{P(n)\}_{n=1}^{\infty}$ converges.

In order to analyze the convergence of MCCSFLA with the Markov chain theory, we give the following Lemma.

Lemma 11: [112]: Let P be a $m \times m$ primitive stochastic matrix that converges to $P^{\infty} = (\pi^T, \pi^T, \cdots, \pi^T)^T$ with π^T=$(p_1, p_2, \cdots, p_m)$, $R, T \neq 0$, $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times k}$,. If a $k \times k$ non-negative stochastic phalanx B has the form $B = \begin{pmatrix} P & 0 \\ R & T \end{pmatrix}$, then $B^{\infty} = \lim_{n\to\infty} \begin{pmatrix} P^n & 0 \\ \sum_{m=0}^{n-1} T^m R P^{n-m-1} & T^n \end{pmatrix} = \begin{pmatrix} P^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix}$ is a stable stochastic matrix, and $B^{\infty} = g' \cdot b^{\infty}$, where $b^{\infty} = (p_1, p_2, \cdots, p_m, 0, 0, \cdots 0)$ and $\sum_{l=1}^{m} p_l = 1$.

The detailed proof can be found in [112].

Based on the above analysis we can prove MCCSFLA converges to the global optimum when the number of iterations tends to be infinite.

Theorem 3: MCCSFLA can guarantee convergence to the global optimum.

Proof: Let the total transition matrix be expressed as $B = \begin{pmatrix} P & 0 \\ R & T \end{pmatrix}$, with $R = \begin{pmatrix} PU_{21} \\ \vdots \\ PU_{2^{2(V-1)},1} \end{pmatrix}$ and $T = \begin{pmatrix} PU_{22} & & \\ \vdots & \ddots & \\ PU_{2^{2(V-1)},2} & \cdots & PU_{2^{2(V-1)},2^{2(V-1)}} \end{pmatrix}$. In the above we proved that P and B are primitive stochastic matrices. As $P > 0$, $U_{21} \neq 0$, the product $PU_{21} \neq 0$, so $R \neq 0$. As $P > 0$, $U_{22} \neq 0$, the product $PU_{22} \neq 0$, so $T \neq 0$. According to Theorem 1, $\{Y(n)\}_{n=1}^{\infty}$ converges to $B^{\infty} = g' \cdot b^{\infty}$, where $b^{\infty} = (p_1, p_2, \cdots, p_{2^{2(V-1)MN}}, 0, 0, \cdots 0)$, and $\sum_{l=1}^{2^{2(V-1)MN}} p_l = 1$. So we have

$$\begin{aligned} B^{\infty} &= \lim_{n\to\infty} \begin{pmatrix} P^n & 0 \\ \sum_{m=0}^{n-1} T^m R P^{n-m-1} & T^n \end{pmatrix} \\ &= \begin{pmatrix} P^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix} \\ &= \begin{pmatrix} p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \\ p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_1 & \cdots & p_{2^{2(V-1)MN}} & 0 & \cdots & 0 \end{pmatrix} \end{aligned} \tag{3.32}$$

The size of matrix B is $2^{2(V-1)(MN+1)} \times 2^{2(V-1)(MN+1)}$, so it has $2^{2(V-1)(MN+1)}$ states. As for the status with state number $i \in [1, 2^{2(V-1)MN}]$ the elite frog $D_e = D_{best}$, according to the definition of the convergence in Definition 2, $\{Y(n)\}_{n=1}^{\infty}$ converges to the global optimal solution.

3.5 Experimental Study

In this section we present the simulation results of MCCSFLA. In order to demonstrate the algorithm's capabilities, we compare it against the genetic algorithm, the quantum

evolutionary algorithm, the selective mapping algorithm and the original method without using partial transmission sequences.

In our simulation, we consider the QPSK, 16QAM and 64QAM modulations respectively. We set the number of non-overlapping sub-blocks $V = 16$. The objective function in (3.7) is used to evaluate each PAPR reduction algorithm. For simplicity, the input symbol sequence is considered to be randomly distributed. The population sizes of both GA and QEA are 40 and the number of frogs in MCCSFLA is also 40. We set the number of groups in MCCSFLA to 4, which means there are 10 frogs in each group. In GA, we use a similar setting to that in [96]. We set the crossover probability to 0.9, and set the mutation probability to 0.05. In QEA, we use the same lookup table as Table II in [117]. In order to do a performance comparison, we just copy the table here in Table 3.1.

For comparison purposes, we set the number of distinct sign sequences of SLM to 8. In each simulation, we test the algorithms with 128, 256, and 512 subcarriers in one symbol respectively.

In order to compare the performance of MCCSFLA, the complementary cumulative distribution function (CCDF) is used to evaluate the algorithms. The CCDF is given as:

$$CCDF = P\{PAPR > PAPR_0\} \tag{3.33}$$

where P is the probability function.

Fig. 3.1 to Fig. 3.9 show the CCDF curves of MCCSFLA, GA, QEA, SLM and "original" with the QPSK, 16QAM and 64QAM modulations respectively. The maximum generation of MCCSFLA, GA and QEA is set to 20. "Original" means the PAPR without using PTS. We simulate the OFDM system with 128, 256 and 512 subcarriers respectively. Each case is tested with 1×10^5 symbols independently. As can be seen from Fig.3.1, among all the different subcarriers, MCCSFLA provides better performance than GA, QEA, SLM and "original". For example, with 20 iterations and 128 subcarriers, when $CCDF = 10^{-3}$, the average PAPR obtained by MCCSFLA is around 5.59 dB, which is

Table 3.1: The sign of quantum rotation angle

x_i	b_i	$f(x) \geq f(b)$	$s(\alpha_i\beta_i)$			
			$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0
0	0	True	0	0	0	0
0	1	False	0	0	0	0
0	1	True	-1	+1	±1	0
1	0	False	-1	+1	±1	0
1	0	True	+1	-1	0	±1
1	1	False	+1	-1	0	±1
1	1	True	+1	-1	0	±1

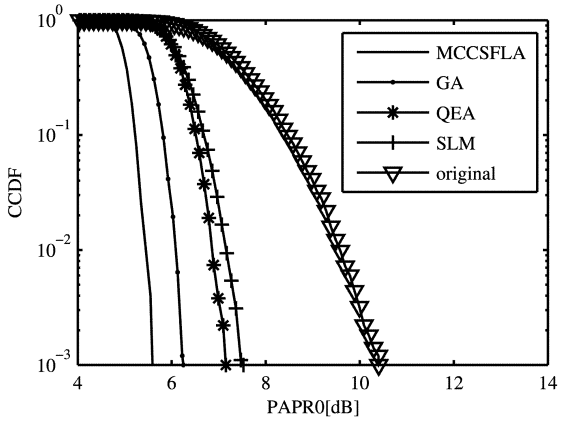


Figure 3.1: QPSK, CCDF of the PAPR with subcarrier=128.

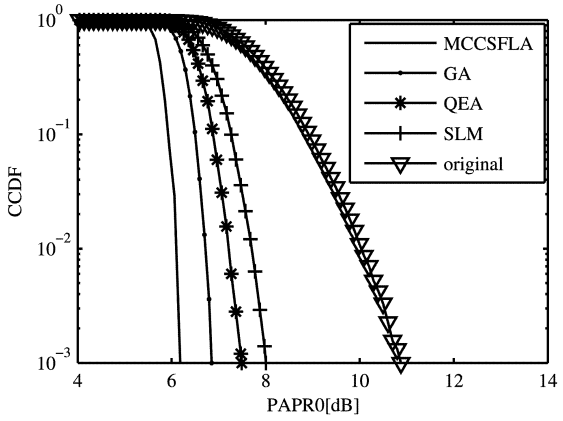


Figure 3.2: QPSK, CCDF of the PAPR with subcarrier=256.

the lowest among all the algorithms under evaluation. In contrast, GA and QEA achieve results for PAPR of around 6.25 dB and 7.15 dB. In comparison, SLM can only obtain a higher PAPR of 7.53 dB due to the few partial transmit sequences. "Original", however, suffers a substantial performance loss, with the highest PAPR of around 10.40 dB. Similar conclusions can be observed from Fig. 3.2 to Fig. 3.9.

As a numerical example for computational complexity, we use the parameters in Fig. 3.1, that is $W = 4$, $V = 16$, $L = 128$ $Gen1 = Gen2 = Gen3 = 20$, $Pop1 = Pop2 = 40$,

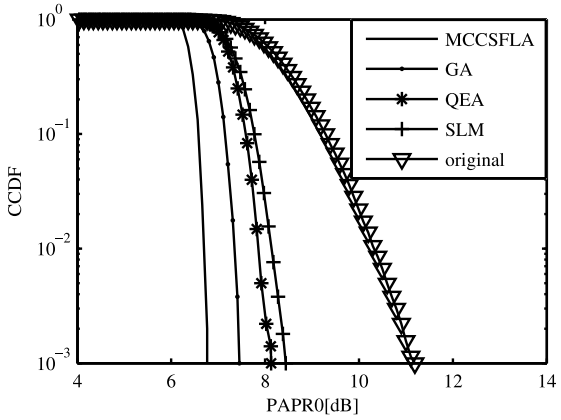


Figure 3.3: QPSK, CCDF of the PAPR with subcarrier=512.

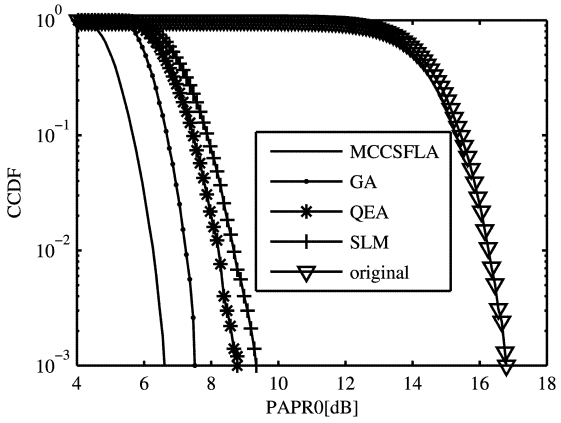


Figure 3.4: 16QAM, CCDF of the PAPR with subcarrier=128.

$M = 4$, and set the number of distinct sign sequences in SLM to 8. With 1×10^5 symbols simulation, we can get the average number of jumps for the worst frog in each group is $AJ = 8.7$. So we have $W^{V-1} = 1073741824$. According to Section 3.3.12, the computational complexity of the PAPR reduction schemes is calculated and shown in Table 3.2. From the table we can see that the computational complexity of MCCSFLA-PTS is a little bit lower than GA-PTS and QEA-PTS with $L = 128$. The proposed MCCSFLA-PTS achieved best PAPR reduction as shown in Fig. 3.1 to Fig. 3.9. As we

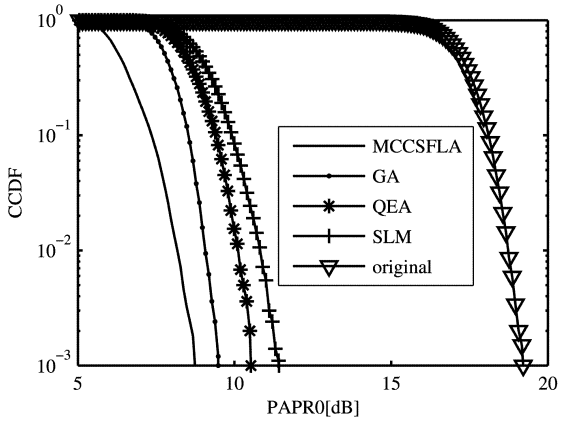


Figure 3.5: 16QAM, CCDF of the PAPR with subcarrier=256.

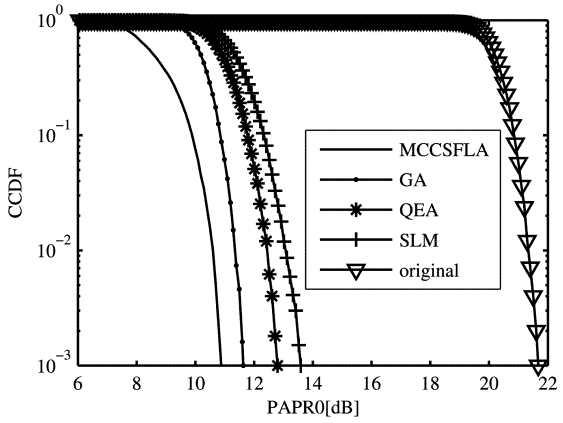


Figure 3.6: 16QAM, CCDF of the PAPR with subcarrier=512.

can see from Fig. 3.1 and Table 3.2, although SLM has a lower computational complexity, its PAPR reduction performance is the worst among heuristic algorithms.

Fig. 3.10, Fig. 3.11 and Fig. 3.12 show the convergence, defined as the average PAPR for 100 symbols, for QPSK modulation, MCCSFLA and GA with 128, 256 and 512 subcarriers. Fig. 3.10 shows clearly that MCCSFLA performs significantly better than GA in terms of the convergence speed with 128 subcarriers. It can be seen in Fig. 3.10, at the first generation, the average PAPR of MCCSFLA is lower than GA as the

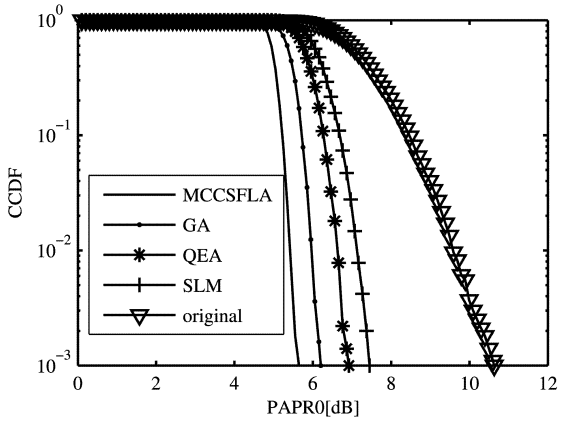


Figure 3.7: 64QAM, CCDF of the PAPR with subcarrier=128.

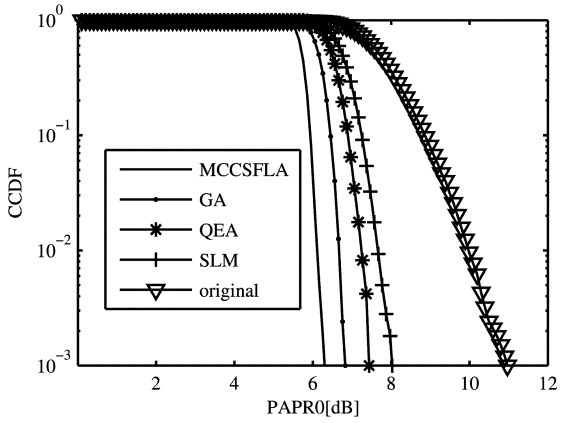


Figure 3.8: 64QAM, CCDF of the PAPR with subcarrier=256.

local search of MCCSFLA is more effective than GA. Within the initial 50 iterations, the PAPR of MCCSFLA decreased quickly with the growth of the generations, as the convergence speed improves with the local search and the shuffling of the population. On the other hand, GA shows a slower convergence rate than MCCSFLA. From 50 to 100 iterations, MCCSFLA has approached close to 4.55 dB, while GA is still far from it. Over all 100 iterations, MCCSFLA provides a lower average PAPR than GA, and MCCSFLA converges with a faster rate. From Fig. 3.11 and Fig. 3.12, similar conclusions can

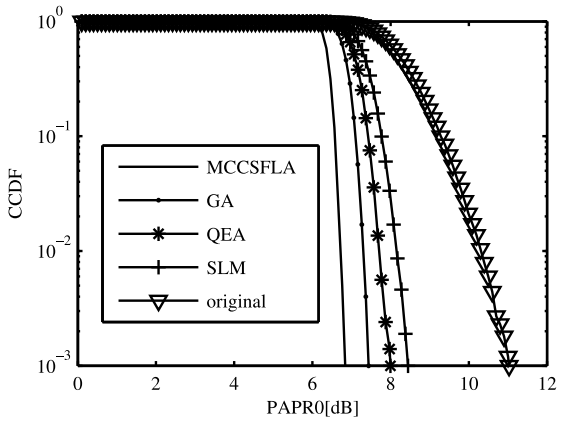


Figure 3.9: 64QAM, CCDF of the PAPR with subcarrier=512.

also be obtained when the number of the subcarriers is 256 and 512 respectively. Overall, MCCSFLA is more effective and suitable for PAPR reduction than GA. Moreover, similar conclusions can also be obtained from the simulation results that MCCSFLA converges faster than QEA with the same number of generations.

3.6 Summary

In this chapter, we propose a novel modified chaos clonal shuffled frog leaping algorithm for partial transmit sequence selection in OFDM systems. We analyze the proposed

Table 3.2: The number of real multiplications and real additions for different methods

Scheme	Real Multiplications	Real Additions
CON-PTS	274877935616	4260607600644
GA-PTS	233472	3217408
QEA-PTS	233472	3217408
MCCSFLA-PTS	206848	2804736
SLM	30720	74752

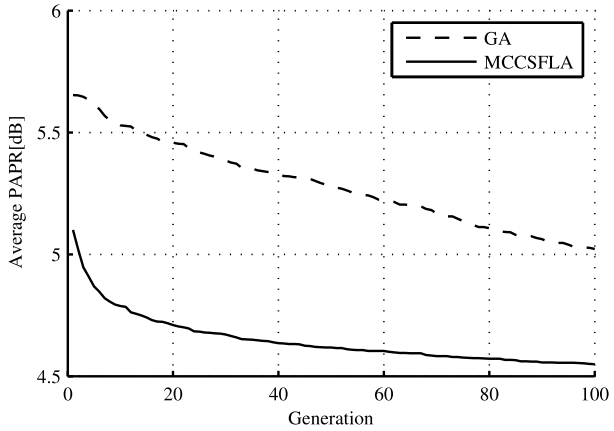


Figure 3.10: Average PAPR change by generation with subcarrier=128.

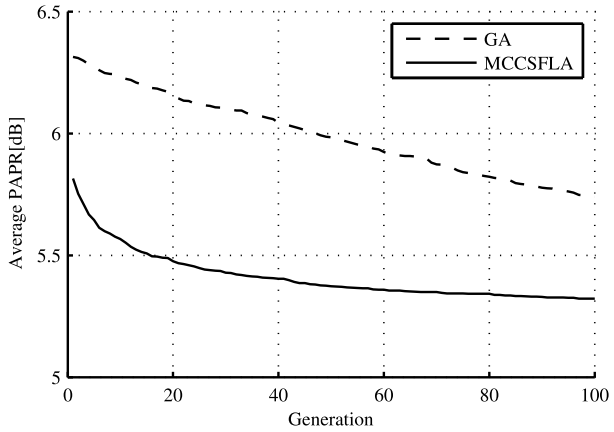


Figure 3.11: Average PAPR change by generation with subcarrier=256.

algorithm using the Markov chain theory and prove that the algorithm converges to the global optimum. Simulation experiments are conducted to compare the proposed algorithm with the genetic algorithm, the quantum evolutionary algorithm, the selective mapping algorithm and the original approach respectively. The results show that the modified chaos clonal shuffled frog leaping algorithm is more efficient in terms of both PAPR reduction and convergence speed.

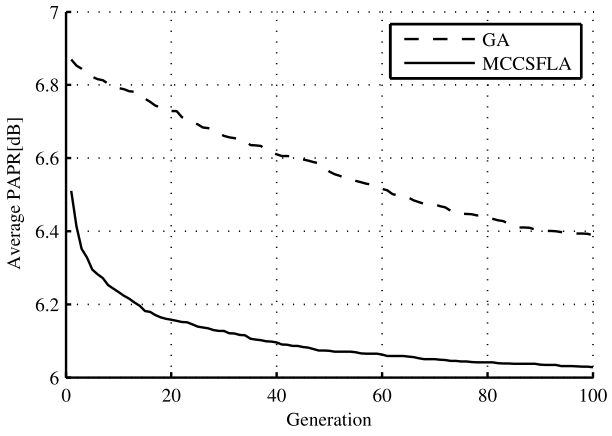


Figure 3.12: Average PAPR change by generation with subcarrier=512.

Chapter 4

Target Coverage Based on QACEA for Self-organizing WSN

4.1 Introduction

Recent advances in wireless communications and technology growth in the field of embedded computing has facilitated the deployment of more self-organizing wireless sensor nodes for many important applications, such as building security, military monitoring, medical diagnostic, environment surveillance, etc [119].

A self-organizing sensor network generally consists of a large number of low-power and low-cost sensor nodes for exact sensing a certain number of targets in the sensing areas. The self-organized micro nodes are usually installed in harsh environments, whose sensing range is limited and in some scenarios they have to monitor multiple targets simultaneously. In such scenarios, enhancing the target coverage rate is crucial because sensor nodes operate with a limited sensing range and sensing ability. Accordingly, the objective of a target coverage scheme is to select an optimal set of targets for the sensor nodes that maximize the target coverage rate [9].

However, when the scale of the sensor nodes and targets is relatively large, the overall computational complexity and space complexity increases exponentially with the number of sensor nodes and targets, which becomes a bottleneck in the target coverage problem.

In order to enhance the target coverage efficiency, many previous studies have addressed various heuristic approaches for the target coverage problem in wireless sensor networks [120]. A research effort on the optimal K-cover problem for targets in wireless sensor networks based on the genetic algorithm can be found in [57]. GA performs well in the beginning, but it suffers from a premature convergence and a low convergence rate only after few iterations. Another method based on heuristically solving the sensor networks coverage problem using simulated annealing has been presented in [121]. Their work focused on a coverage optimization with many directional sensor nodes. However, for SA, the search is prone to get stuck into local optima and fall into evolutionary stagnation. In [122], A particle swarm optimization method was proposed for sensor node autonomous deployment. However, its convergence speed is slow when the numbers of sensor nodes and target nodes are large.

In this chapter, a quantum ant colony evolutionary algorithm (QACEA) is proposed to enhance the target coverage rate of self-organizing wireless sensor networks. We also establish a mathematical model for the target coverage problem. By introducing QACEA into the target coverage, a fitness function for evaluating the target coverage rate is designed to optimize the selection of targets for each sensor node. In the simulation, the effect of QACEA is compared to those given by heuristic techniques of GA and SA for the self-organizing wireless sensor networks with different numbers of sensor nodes and sensing radius. Results show that the target coverage rate of the proposed QACEA method is about 10 percentage and 20 percentage higher than that of GA and SA respectively with different sensing radii. Besides that, the number of targets that are successfully detected by the proposed QACEA is much higher than that of GA and SA. So the proposed method

significantly enhanced the monitoring results.

4.2 System Model

This section describes the mathematical model for the target coverage problem with respect to the sensor distribution and the target distribution when the sensing radius and node capability are limited. Assume that there are M targets and N sensor nodes uniformly and randomly distributed over the monitoring area, the sensing range relationship can be shown as

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N-1} & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N-1} & c_{2,N} \\ \vdots & & c_{m,n} & & \vdots \\ c_{M-1,1} & c_{M-1,2} & \cdots & c_{M-1,N-1} & c_{M-1,N} \\ c_{M,1} & c_{M,2} & \cdots & c_{M,N-1} & c_{M,N} \end{bmatrix} \tag{4.1}$$

where C is the sensing range relationship matrix and $c_{m,n} \in \{0, 1\}$ represents the coverage relationship between the n^{th} sensor node and the m^{th} target. $c_{m,n} = 1$ means that the m^{th} target is within the monitoring range of the n^{th} sensor and the distance between the sensor node n and target m is smaller than the sensing radius. Similarly, $c_{m,n} = 0$ means the m^{th} target is outside the scope of the monitoring range of the n^{th} sensors and the distance between sensor node n and target m is larger than the sensing radius.

The monitoring targets set for sensor nodes in self-organizing wireless sensor networks

can be represented as

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N-1} & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,N-1} & s_{2,N} \\ \vdots & & s_{m,n} & & \vdots \\ s_{M-1,1} & s_{M-1,2} & \cdots & s_{M-1,N-1} & s_{M-1,N} \\ s_{M,1} & s_{M,2} & \cdots & s_{M,N-1} & s_{M,N} \end{bmatrix} \tag{4.2}$$

where S is the monitoring targets selection matrix. The variable $s_{m,n}$ will be either 1 or 0. $s_{m,n} = 1$ means that the m^{th} target is monitored by the n^{th} sensor and $s_{m,n} = 0$ means that the m^{th} target is not monitored by the n^{th} sensor.

Limited by their sensing, storage and computation capabilities, small wireless sensors can only monitor a limited number of targets, which can be represented as

$$\sum_{m=1}^{M} s_{m,n} \leq F, n = 1 \ldots N \tag{4.3}$$

where F is the maximum number of targets that a sensor node can monitor simultaneously.

As the sensor node can not monitor the target that is outside its monitoring range, we have

$$s_{m,n} \leq c_{m,n} \tag{4.4}$$

Given the sensor distribution and the target distribution, the objective function of the target coverage problem is to maximize the number of detected targets of the sensor network by optimally selecting a set of targets for each sensor node with respect to the sensing range and monitor capability constraints. We assume that each target needs to be monitored by E sensor, the system model can be shown as Objective:

$$\max f(s_{11}, s_{12}, \cdots, s_{MN}) = \sum_{m=1}^{M} w_m \tag{4.5}$$

Subject to:

$$\sum_{m=1}^{M} s_{m,n} \leq F, n = 1 \ldots N \tag{4.6}$$

$$s_{m,n} \leq c_{m,n} \tag{4.7}$$

where

$$w_m = \begin{cases} 1 & \sum_{n=1}^{N} s_{m,n} \geq E \\ 0 & \sum_{n=1}^{N} s_{m,n} < E \end{cases} \tag{4.8}$$

In formula (4.8), $w_m = 1$ means that the m^{th} target is detected by the sensor nodes and $w_m = 0$ otherwise.

4.3 Target Coverage in Self-organizing Wireless Sensor Networks based on QACEA

Based on the principles of quantum computation and the quantum theory, we developed a quantum ant colony evolutionary algorithm for target coverage in self-organizing wireless sensor networks which can achieve a higher target coverage rate and higher number of detected targets compared with the conventional GA and SA. Because the quantum bit representation can represent a linear superposition of all possible solutions, QACEA has a better characteristic of diversity than GA and SA.

QACEA is essentially defined by several steps, namely ant and colony encoding, ant initialization, quantum state observation, fitness evaluation, path selection and pheromone update, quantum rotation operation, quantum mutation operation, etc.

4.3.1 Ant and Colony Encoding

QACEA employs a fixed number of binary ants and a fixed number of quantum ants in the ant colony, which is a combination of deterministic and probabilistic representation and uses both ant search and a quantum gate to drive the ant colony towards the best

solution. Besides the colony, we keep an elite binary ant with the highest target coverage rate in each iteration.

We only encode the targets that are within the sensing range of the corresponding sensors. When the m^{th} target is within the monitoring range of the n^{th} sensor, we have $c_{m,n} = 1$ in formula (4.8), which means that the bit $s_{m,n}$ is valid. When the m^{th} target is outside the scope of the monitoring range of the n^{th} sensor, we have $c_{m,n} = 0$, which means that the bit $s_{m,n}$ is invalid. So we only encode $s_{m,n}$ when $c_{m,n} = 1$. An example can be shown as

$$C = \begin{bmatrix} \underline{1} & 0 & 0 & \underline{1} \\ 0 & \underline{1} & \underline{1} & 0 \\ \underline{1} & \underline{1} & 0 & 0 \\ 0 & 0 & 0 & \underline{1} \end{bmatrix} \quad S = \begin{bmatrix} \underline{0} & 0 & 0 & \underline{1} \\ 0 & \underline{1} & \underline{0} & 0 \\ \underline{1} & \underline{0} & 0 & 0 \\ 0 & 0 & 0 & \underline{0} \end{bmatrix} \tag{4.9}$$

where the underlined bits in matrix S are the valid bits that we need to encode. In this way, the code length of our ant equals to the number of valid bits and can be shown as

$$L = \sum_{m=1}^{M} \sum_{n=1}^{N} c_{m,n} \tag{4.10}$$

where L is the code length.

The binary ant uses a binary string representation which represents the sequence of underlined bits in matrix S and the value of each bit can be 0 or 1.

For the quantum ant, a quantum bit may be in the 1 state, signed as the state vector $|1\rangle$, in the 0 state, signed as the state vector $|0\rangle$ or in any superposition of the two which can be represented as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4.11}$$

where α and β are complex numbers that specify the probability amplitudes of the $|0\rangle$ state and $|1\rangle$ state. Specifically, $|\alpha|^2$ and $|\beta|^2$ denote the probability that the Q-bit will be observed as the $|0\rangle$ state and $|1\rangle$ state in the observation.

Normalization of the state guarantees that the two probabilities must satisfy

$$|\alpha|^2+|\beta|^2=1 \tag{4.12}$$

In this way, a quantum ant with the probabilistic representation is defined by a string of Q-bits with length L, which can be represented as

$$A = \left| \begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_L \\ \beta_1 & \beta_2 & \cdots & \beta_L \end{array} \right| \tag{4.13}$$

Thus, the valid bits in matrix S are mapped to the Q-bit in A.

4.3.2 Ant Initialization

In QACEA all the Q-bits in the quantum ant are initialized with the random value generated by the Logistic map which can be shown as

$$x_{k+1} = \mu x_k(1 - x_k), \;\; k = 0, 1, 2, \cdots, K \tag{4.14}$$

We set μ=4 and randomly generate an initial ant colony with $\alpha_k = \cos(2\pi x_k)$ and $\beta_k = \sin(2\pi y_k)$. In this way, all the possible quantum states appear with a random probability at the beginning.

4.3.3 Quantum State Observation

In order to produce a new binary solution, the observation operation of a quantum state results in its collapse to a binary representation for evaluation which can be shown as

$$z_l = \begin{cases} 0, & random[0,1] > |\alpha_l|^2 \\ 1, & random[0,1] \leq |\alpha_l|^2 \end{cases} \tag{4.15}$$

where $Z = (z_1, z_2, \cdots, z_L)$ is the generated binary solution, $random[0,1]$ is a uniform random number between 0 and 1.

4.3.4 Fitness Evaluation

Each binary solution Z in the colony can be mapped to a matrix S and then covert to a target coverage scheme. Each target coverage scheme represents a possible solution to the problem which can be evaluated in order to obtain the best scheme with the highest target coverage rate. We calculate the fitness value of each binary solution Z with formula (4.5) after mapping it to matrix S and update the elite binary ant if the fitness value of the binary solution Z is higher. In order to meet the constraints (4.6), we uniformly and randomly convert 1 to 0 in S on the relevant row before fitness evaluation. As we only encode the valid bit in the matrix S, all the solutions satisfy the constraints (4.7).

4.3.5 Path Selection and Pheromone Update

The ant-cycle system is applied to the evolution of the binary ants in the colony of the algorithm. QACEA constructs a binary ant by selecting a bit value with probabilities based on (4.16).

$$p_{ij}(t) = \frac{\tau_{ij}^{\lambda}(t)\eta_{ij}^{\mu}(t)}{\tau_{i0}^{\lambda}(t)\eta_{i0}^{\mu}(t) + \tau_{i1}^{\lambda}(t)\eta_{i1}^{\mu}(t)} \tag{4.16}$$

where t is the iteration, step $p_{ij}(t)$ is the transition probability at the i^{th} step, τ_{i0} and τ_{i1} are the pheromone on path 0 and 1 at the i^{th} step, η_{i0} and η_{i1} are the visibilities on path 0 and 1 at the i^{th} step, μ and λ reflect the relative importance of visibility and pheromone at the i^{th} step and the path selection result $j \in \{0, 1\}$.

The visibility can be shown as formula (4.17)

$$\eta_{m,n}^{1}(t) = \begin{cases} \left(1 + \sum_{i=1}^{n} s_{m,i}\right) & \sum_{i=1}^{n} s_{m,i} < E \\ 0 & \sum_{i=1}^{n} s_{m,i} \geq E \end{cases} \tag{4.17}$$

where $\eta_{n1}^{(m)}(t)$ is the calculation of the visibility. As each target needs to be monitored by E sensors, the visibility is higher when the number of sensors for target m is close to E. The visibility is set to zero when the number of sensors for target m is above E as each target only needs to be monitored by E sensor nodes. In the way, we have $\eta_{m,n}^{0}(t) = E - \eta_{m,n}^{1}(t)$.

The fitness of the binary ant is calculated according to formula (4.5) and fitness of the elite binary ant is updated if the fitness value of the binary ant is higher. The pheromone trail intensity is updated according to the fitness value of the elite binary ant.

The pheromone update is to add the pheromone on edges of the elite binary ant and to evaporate pheromone on the other edges. The evaporation increases the diversity of the ant colony. The pheromone is updated according to

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau^{best} \tag{4.18}$$

where τ_{ij} is the pheromone on path j at the i^{th} step and $\rho \in (0, 1)$ is the evaporation coefficient. The increment of pheromone $\Delta\tau^{best}$ is given to those paths belonging to the elite binary ant which can be shown as

$$\Delta\tau^{best} = \varphi \sum_{m=1}^{M} w_m \tag{4.19}$$

where $\sum_{m=1}^{M} w_m$ is the fitness value of Z_{best} and φ is a constant parameter.

Table 4.1: The sign of quantum rotation angle

z_l	z_{best}^{old}	$f(z_l) > f(z_{best}^{old})$	$\Delta\theta_l$	$s(\alpha_l\beta_l)$ $\alpha\beta > 0$	$\alpha\beta < 0$	$\alpha = 0$	$\beta = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0.05π	+1	-1	0	±1
0	1	True	0.05π	-1	+1	±1	0
1	0	False	0.01π	-1	+1	±1	0
1	0	True	0.025π	+1	-1	0	±1
1	1	False	0.005π	+1	-1	0	±1
1	1	True	0.025π	+1	-1	0	±1

4.3.6 Quantum Rotation Operation

The quantum gate of QACEA is a rotation gate and can be represented as

$$\begin{bmatrix} \alpha_l^{new} \\ \beta_l^{new} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \alpha_l \\ \beta_l \end{bmatrix} \tag{4.20}$$

where α_l and β_l are complex numbers that specify the probability amplitudes of the $|0\rangle$ state and $|1\rangle$ state on the l^{th} quantum bit and α_l^{new} and β_l^{new} are renewed probability amplitudes. The value of $\theta_l = s(\alpha_l\beta_l)\Delta\theta_l$ is updated through a lookup Table 4.1, where z_{best}^{old} is the elite binary ant in the last iteration and $s()$ is the sign operation. The only exception occurs in the first iteration where z_{best}^{old} is the elite binary ant in the current iteration.

4.3.7 Quantum Mutation Operation

The quantum mutation operation can be represented as

$$\begin{bmatrix} \beta_l \\ \alpha_l \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_l \\ \beta_l \end{bmatrix} \tag{4.21}$$

where the probability amplitudes are updated with a quantum NOT gate with a small probability. The smaller the fitness value of the observation result, the higher the mutation rate.

Algorithm 2 The quantum ant colony evolutionary algorithm based target coverage method

1: **Begin**
2: Generate the initial binary ants and the quantum ants
3: **while** Algorithm has not reached designated iterations **do**
4: Quantum state observation for the quantum ants
5: Evaluate the fitness value for the observation result
6: Update the elite binary ant Z_{best}
7: Path selection to generate binary ants
8: Evaluate the fitness value for the binary ants
9: Update the elite binary ant Z_{best}
10: Pheromone update
11: Quantum rotation operation
12: Quantum mutation operation
13: **end while**
14: Output the fitness value of elite binary ant Z_{best}
15: **End**

4.3.8 Basic Steps

The basic steps of MCCSFLA are described in Algorithm 2:

4.4 Simulation Results and Discussion

In the simulation process, the monitoring area is set to 600 × 600m and there are 200 targets in the area. The sensor nodes and targets are distributed uniformly and randomly in it. In the monitoring process each sensor can simultaneously monitor 5 targets within its monitoring range and each target needs at least 3 sensors to monitor it simultaneously. We use the typical parameters. In QACEA, the number of ants is 40, the pheromone evaporation coefficient is $\rho = 0.9$, μ and λ are set to 2. In SA, the initial temperature is set to 300 degrees and the annealing temperature coefficient is set to 0.85. In GA, the chromosome number is 40, crossover probability is set to 0.7, mutation probability is set to 0.05 and the generation gap is set to 0.9. The maximum iteration of all the three algorithms is set to 100.

Fig.4.1 to Fig.4.4 show the target coverage rate of the self-organizing wireless sensor networks by iteration using QACEA, GA and SA with the sensing radius 50, 60, 70 and 80 respectively. The number of sensor nodes is 150. According to these comparisons, the proposed coverage scheme based on QACEA has a higher target coverage rate than the schemes based on GA or SA. As it may be observed in the figures, in the beginning, the target coverage rate of self-organizing wireless sensor networks of all the three algorithms increased. However, when the iteration number is increased to 60, the SA performance degrades due to its premature convergence. Similarly, since falling into local optimal solutions is an inevitable phenomenon in GA, it tends to slow down the convergence speed as well after 60 iterations. As shown in the figures, QACEA still has a fast convergence rate after 60 iterations which means that the algorithm is prevented from premature

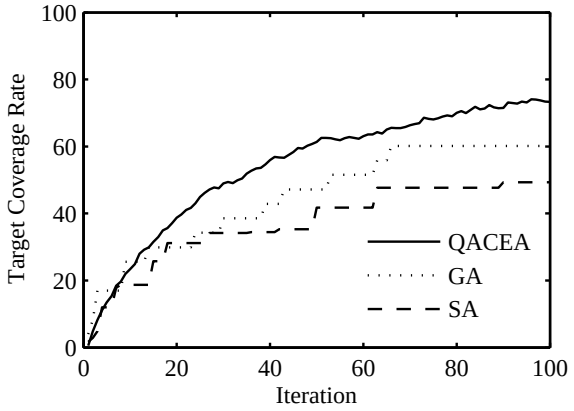


Figure 4.1: Target coverage rate (%) by iterations with sensing radius 50m.

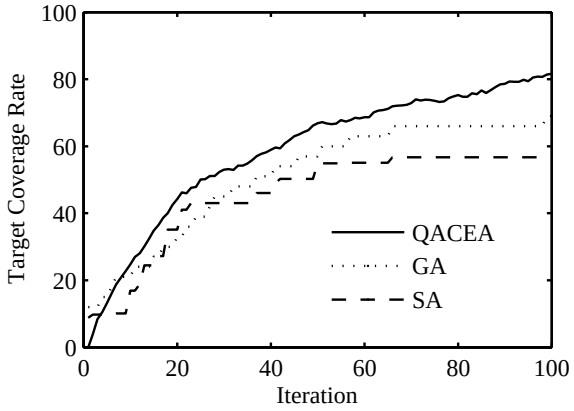


Figure 4.2: Target coverage rate (%) by iterations with sensing radius 60m.

convergence. In other words, it achieves a better convergence rate and a higher target coverage rate than GA and SA. The results show that the target coverage rate of the proposed QACEA method is about 10 percentage and 20 percentage higher than that of GA and SA respectively with different sensing radius.

A summary of the number of detected targets obtained for QACEA, GA and SA with different numbers of sensor nodes and sensing radii is illustrated in Fig.4.5 to Fig.4.8. As the figures show, QACEA yielded much better results compared to GA and SA. It can

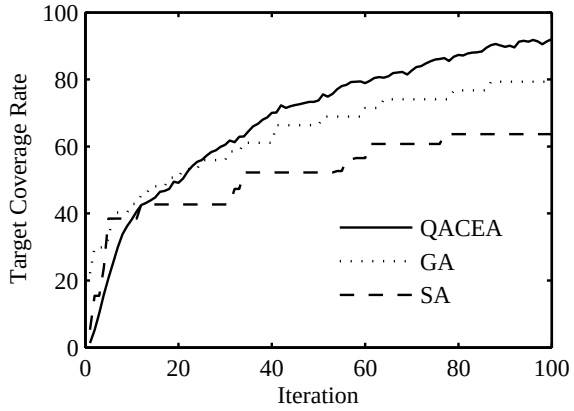


Figure 4.3: Target coverage rate (%) by iterations with sensing radius 70m.

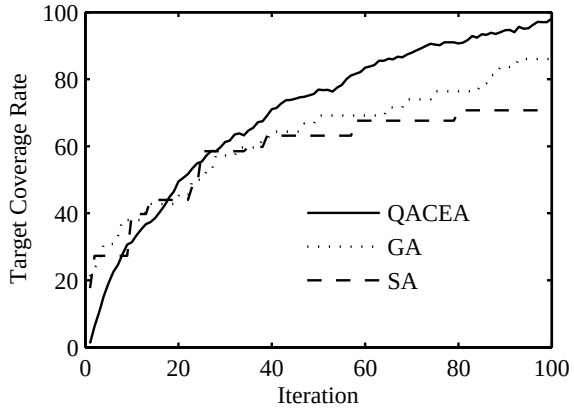


Figure 4.4: Target coverage rate (%) by iterations with sensing radius 80m.

be seen that the number of detected targets by QACEA is 6.90% to 19.09% higher than that of GA and 32.67% to 54.27% higher than that of SA which indicates a higher target coverage rate.

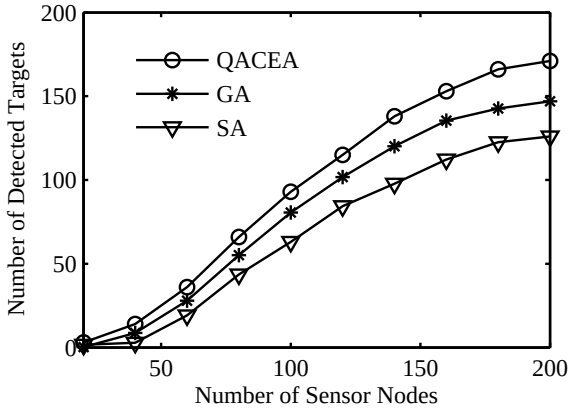


Figure 4.5: Number of successful detected target by sensor number with sensing radius 50m.

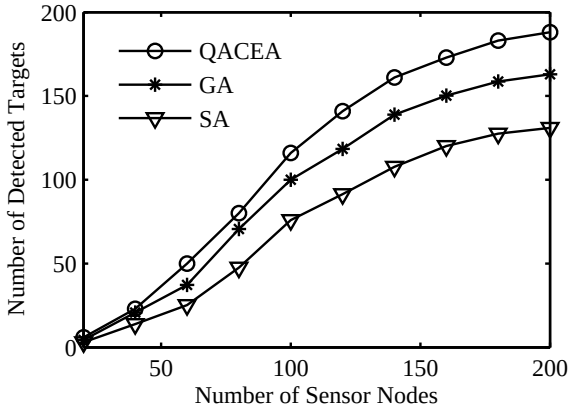


Figure 4.6: Number of successful detected target by sensor number with sensing radius 60m.

4.5 Summary

In this chapter, a target coverage scheme based on the quantum ant colony evolutionary algorithm (QACEA) was proposed to increase the target coverage rate of the self-organizing wireless sensor networks. The system model for the target coverage problem was established while forming the objective function. The simulation results demonstrated that

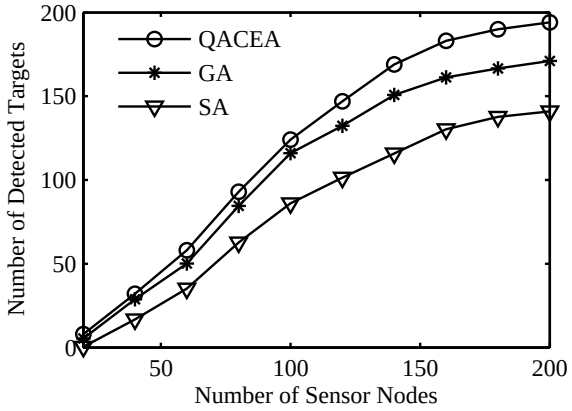


Figure 4.7: Number of successful detected target by sensor number with sensing radius 70m.

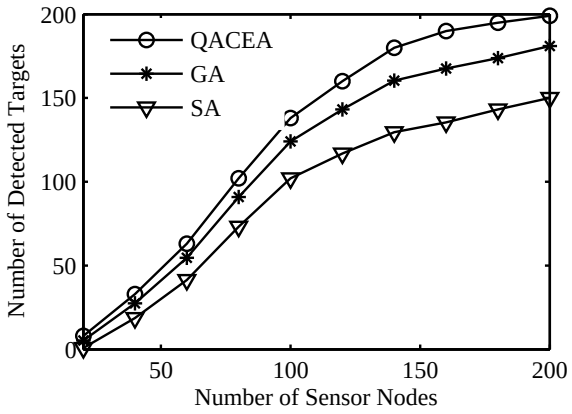


Figure 4.8: Number of successful detected target by sensor number with sensing radius 80m.

QACEA solves the target coverage problem with a higher number of detected targets and a higher target coverage rate than GA and SA.

Chapter 5

Energy Efficient Duty Cycle Design based on QICEA in WSNs

5.1 Introduction

The recent advances in micro-sensor and technology growth in the field of wireless communications has facilitated the deployment of more capable WSNs. A WSN generally consists of a large number of sensor nodes for exact sensing. Limited by their size, small wireless sensors are generally low-cost, low-power devices with a restricted power source and limited sensing range.

As the energy of a battery is limited, we can schedule the sensor nodes activity to enhance the lifetime of WSNs. In other words, each node in a WSN can switch between the active mode and sleep mode in a certain order. If the time unit of the mode switching is the cycle, the active orders of all sensor nodes in a WSN can be expressed as a duty cycle. Due to the limited power source, lifetime optimization through duty cycle design has always been an important issue of WSNs. Careful duty cycle design can be an effective optimization means for achieving maximum network lifetime with full coverage

constraints. However, the computational complexity of the duty cycle design problem increases exponentially with the increment of the number and the lifetime of the nodes. Thus, the duty cycle design problem can be formulated as a nonlinear combinational optimization problem and its computational complexity is too high for an exhaustive search algorithm for practical implementation [120].

Fortunately, heuristic algorithms provide a solution to the nonlinear combinational optimization problem, which include PSO, GA and the SA. A research effort on the duty cycle design problem based on the greedy algorithm can be found in [123]. Their work focused on improving the scheduling scheme for sensor sets. The greedy algorithm is simple and fast but it usually yields a shorter network lifetime than evolutionary algorithms. In [124], an improved GA was proposed for optimizing node scheduling in wireless sensor networks. Their algorithms have shown to perform well on sensor networks with a limited number of nodes. But it suffers from a premature convergence and low convergence rate when the number of nodes is high. A duty cycle design scheme is proposed in [125] and PSO is used to solve the problem. However, in practice, it gets stuck into local optima and falls into evolutionary stagnation.

Based on the quantum theory, we propose a quantum immune clonal evolutionary algorithm (QICEA) for duty cycle design in WSNs which can achieve a longer network lifetime while maintaining full coverage in the monitoring area. QICEA combines quantum computation and an artificial immune clonal algorithm where the elements of the antibody population are Q-bits. Simulation results show that this QICEA for the duty cycle design problem in WSNs enjoys a longer network lifetime compared to the conventional GA and SA methods while maintaining full coverage in the monitoring area.

5.2 The System Model

This section describes the system model of the duty cycle design problem with respect to the limited sensing range, the restricted node energy and the full coverage constraint. Cardei has used a similar model to demonstrate that the duty cycle design problem in sensor networks is NP hard [123]. In this work, a network lifetime system based on round units is employed to model the duty cycle design problem.

Consider a WSN with M targets and N sensor nodes distributed over the monitoring area. The sensing relationship can be shown as

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,M-1} & r_{1,M} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,M-1} & r_{2,M} \\ \vdots & & r_{n,m} & & \vdots \\ r_{N-1,1} & r_{N-1,2} & \cdots & r_{N-1,M-1} & r_{N-1,M} \\ r_{N,1} & r_{N,2} & \cdots & r_{N,M-1} & r_{N,M} \end{bmatrix} \quad (r_{n,m} \in \{0,1\}) \tag{5.1}$$

where $r_{n,m}$ is the sensing relationship between the m-th target and n-th sensor, $r_{n,m}$=1 indicates that the m-th target is within the sensing range of the n-th sensor and $r_{n,m}$= 0 otherwise.

Assume that the unit of lifetime is a round and the maximum lifetime of each sensor node is D rounds. The duty cycle of the WSN can be represented as

$$L = \begin{bmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,N-1} & l_{1,N} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,N-1} & l_{2,N} \\ \vdots & & l_{i,n} & & \vdots \\ l_{ND-1,1} & l_{ND-1,2} & \cdots & l_{ND-1,N-1} & l_{ND-1,N} \\ l_{ND,1} & l_{ND,2} & \cdots & l_{ND,N-1} & l_{ND,N} \end{bmatrix} \quad (l_{i,n} \in \{0,1\}) \tag{5.2}$$

where $l_{i,n} = 1$ represents that the n-th sensor is in the active mode in round i and $l_{i,n} = 0$ represents that the n-th sensor is in the sleep mode in round i. The monitoring cycle can be formulated as the form of matrix multiplication, which can be represented as

$$LR = \begin{bmatrix} \sum_{n=1}^{N} l_{1,n} r_{n,1} & \cdots & \sum_{n=1}^{N} l_{1,n} r_{n,M} \\ \vdots & \sum_{n=1}^{N} l_{i,n} r_{n,m} & \vdots \\ \sum_{n=1}^{N} l_{ND,n} r_{n,1} & \cdots & \sum_{n=1}^{N} l_{ND,n} r_{n,M} \end{bmatrix} \tag{5.3}$$

where $\sum_{n=1}^{N} l_{i,n} r_{n,m} > 0$ represents that the m-th target is monitored by at least one sensor in round i, and $\sum_{n=1}^{N} l_{i,n} r_{n,m} = 0$ otherwise. In order to maintain full coverage in the monitoring area in the i-th round, all the elements in the i-th row must be positive. Thus, from top to bottom, if all the elements from row 1 to row i are positive, the network lifetime is longer than i round. So the system model of the duty cycle design problem can be represented as

Objective:

$$f(L) = row\_zero(LR) - 1 \tag{5.4}$$

Subject to:

$$\sum_{i=1}^{ND} l_{i,n} \leq D, n = 1 \ldots N \tag{5.5}$$

where row_zero is the row number for the first zero element that occurs from top to bottom in matrix LR. The constraint $\sum_{i=1}^{ND} l_{i,n} \leq D, n = 1 \ldots N$ represents that the maximum lifetime of each sensor node is D rounds.

From the system model, we can see that the number of possible duty cycle combinations increases exponentially with the sensor number and the maximum lifetime of nodes. So QICEA is proposed to solve this problem.

5.3 Duty Cycle Design based on QICEA

Based on the quantum theory and the clonal selection theory, we introduce QICEA to enhance performance of the traditional evolutionary algorithms. Compared with the traditional iterative optimization methods, QICEA with quantum bit representation can explore the search space with a linear superposition of states. The key steps of QICEA include antibody encoding, initial antibody generation, affinity calculation, antibody selection and clonal mutation operation.

5.3.1 Antibody Encoding

In QICEA, a Q-bit is a kind of probabilistic representation. It can be in the basis state $|0\rangle$, in the basis state $|1\rangle$ or in any superposition of the two states. As the quantum bit can represent a linear superposition of all the possible solutions, its population diversity is higher than other representations. It can be defined as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{5.6}$$

where α and β denote the probability amplitudes of the $|0\rangle$ state and the $|1\rangle$ state which satisfy

$$|\alpha|^2+|\beta|^2=1 \tag{5.7}$$

$|\alpha|^2$ and $|\beta|^2$ are the probabilities that specify the probability amplitudes of the $|0\rangle$ state and the $|1\rangle$ state.

A quantum antibody, which is a form of the probabilities of states, is a Sine and Cosine representation of all the binary solution states and is more straightforward than the traditional Q-bits representation. A quantum antibody with N sensors and maximum lifetime of ND rounds can be represented as

$$Y = \left[\begin{array}{c|c|c|c} \cos(\theta_{1,1}) & \cos(\theta_{1,2}) & \cdots & \cos(\theta_{ND,N}) \\ \sin(\theta_{1,1}) & \sin(\theta_{1,2}) & \cdots & \sin(\theta_{ND,N}) \end{array}\right] \tag{5.8}$$

where θ give the probabilities that the quantum bit exists in states $|0\rangle$ and $|1\rangle$ respectively, which must satisfy

$$\begin{cases} \cos^2(\theta) = |\alpha|^2 \\ \sin^2(\theta) = |\beta|^2 \end{cases} \tag{5.9}$$

5.3.2 Initial Antibody Generation

In QICEA, all the angles of a quantum antibody are initialized with the Chebyshev map which can be represented as

$$\theta_j = 2\pi \times |\chi_j| \tag{5.10}$$

where $\chi_j \in (-1, 1)$is the Chebyshev chaotic random number which can be generated as

$$\chi_{j+1} = \cos(\phi \arccos \chi_j) \tag{5.11}$$

where $\phi \geq 2$ is a constant parameter. It means that all possible angles appear with a random probability at the beginning.

5.3.3 Affinity Calculation

For the duty cycle design problem, the quantum antibody should be converted to a binary solution for evaluation.

$$p_{i,n} = \begin{cases} 0 & random[0,1] \leq \cos^2(\theta_{i,n}) \\ 1 & random[0,1] > \cos^2(\theta_{i,n}) \end{cases} \tag{5.12}$$

The binary result is calculated for the affinity with formula (5.4). When the binary solution does not satisfy the constraint (5.5), we need to reduce the number of working rounds via changing some 1 to 0 for the corresponding node in the binary solution.

5.3.4 Antibody Selection and Clonal Mutation Operation

Unlike the general QEA, a clonal selection is applied in QICEA to determine the antibody with the highest affinity. We assume there are S antibodies in the population. First all the antibodies are ranked in a decreasing order of affinity. Then the first r antibodies with a higher affinity are selected for cloning. The number of clone copies can be represented as

$$Z_k = \left\lceil \frac{\rho S}{k} \right\rceil \tag{5.13}$$

where Z_k is the number of copies that are cloned by the k-th antibody, $\lceil\rceil$ is a ceiling rounding operation and ρ is a multiplying factor that is chosen empirically. In order to enlarge the population size, the total number of cloned antibodies Num must satisfy

$$Num = \sum_{q=1}^{r} Z_k \geq S \tag{5.14}$$

Then the clonal mutation is performed on the cloned antibody population with a quantum gate. In QICEA, the state of Num quantum antibodies can be randomly converted by a rotation gate, which can be represented as

$$\begin{bmatrix} \cos(\theta_{i,n} + \Delta\theta_j) \\ \sin(\theta_{i,n} + \Delta\theta_j) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_j) & -\sin(\Delta\theta_j) \\ \sin(\Delta\theta_j) & \cos(\Delta\theta_j) \end{bmatrix} \begin{bmatrix} \cos\theta_{i,n} \\ \sin\theta_{i,n} \end{bmatrix} \tag{5.15}$$

where $\Delta\theta_j$ is a random rotation angle for the (i, n) angle in the mutation operation. In order to ensure randomness of $\Delta\theta_j$, we use a Logistic map to generate $\Delta\theta_j$ which can be represented as

$$\Delta\theta_j = \lambda(2x_{j+1} - 1) \tag{5.16}$$

where λ is a constant parameter that controls the rotation angle and x_{j+1} is generated by the Logistic map which can be represented as

$$x_{j+1} = \mu x_j(1 - x_j), \;\; j = 0, 1, 2, \cdots, ND \times N \tag{5.17}$$

where μ=4 is a constant parameter of the Logistic map. We use a quantum NOT gate to update the quantum antibody with a small probability, which can be represented as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} = \begin{bmatrix} \sin\theta_i \\ \cos\theta_i \end{bmatrix} \tag{5.18}$$

We can see that the probability amplitudes of $|0\rangle$ state and $|1\rangle$ state exchanged. After the clonal mutation operation, QICEA selects $S - r$ antibodies from Num quantum antibodies. The combination of these $S - r$ antibodies and the original r antibodies will form a new population with S antibodies.

5.3.5 Basic Steps

The basic steps of QICEA for energy efficient duty cycle design are described in Algorithm 3:

Algorithm 3 The QICEA based energy efficient duty cycle design

1: **Begin**
2: Generate the initial antibodies
3: Quantum state observation for the initial antibodies
4: Evaluate the affinity for the initial antibodies
5: **while** Algorithm has not reached designated iterations **do**
6: Antibody selection
7: Clonal mutation
8: Quantum state observation for the antibodies
9: Evaluate the affinity for the antibodies
10: Form a new population with high-affinity antibodies
11: **end while**
12: Output the antibody with highest affinity
13: **End**

5.4 Simulation Results and discussion

In this section, the proposed QICEA is tested with different sensors and targets deployments for the energy efficient duty cycle design problem. The monitoring area is set to 500×500m and the sensor nodes and targets are distributed uniformly and randomly in it. The sensing radius for each sensor is 300m and the maximum lifetime of each sensor node is 10 rounds. In QICEA, the number of antibodies is 40. Comparisons are made with GA and SA. In GA, we set the chromosome number to 40, the crossover probability to 0.9, the generation gap to 0.9 and the mutation probability to 0.05. In SA, the initial temperature is set to 500 degrees and the annealing temperature coefficient is set to 0.98. The maximum number of iterations of all the three algorithms is set to 100.

Fig. 5.1 to Fig. 5.4 show the network lifetime of QICEA, GA and SA on the energy efficient duty cycle design problem with 30 targets, 75 sensors and 40 targets, 100 sensors, respectively. From the figures, we can see that our algorithm produces a duty cycle with a longer network lifetime in both conditions after 100 iterations. In the beginning, the convergence rates of all the three algorithms increased. However, SA gets stuck in a local optima and is not able to escape it. We can see that the convergence rate is very slow after 20 iterations which means that the algorithm falls into premature convergence. The performance of GA is shown to outperform SA most of the time in the convergence rate as well as the final network lifetime. It has a fast convergence rate at the beginning but then stabilizes after 50 iterations. It can be seen that QICEA finds a high-quality duty cycle scheme much faster than SA and GA. It also searches for solutions more efficiently. The simulation results show that in Fig.1 to Fig.2, the network lifetime of the proposed QICEA method is 4.74% and 5.67% longer than that of GA and 6.04% and 6.74% longer than that of SA which means QICEA has a higher energy efficiency.

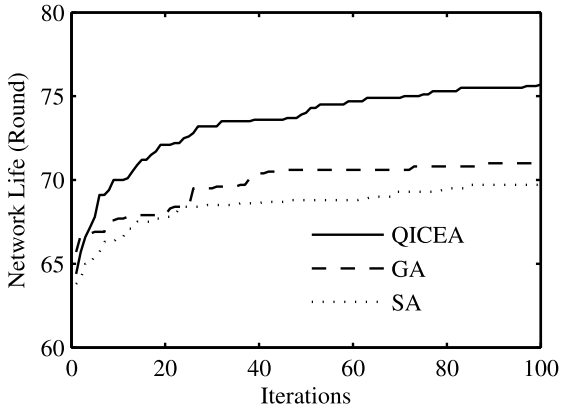


Figure 5.1: Network lifetime with 10 targets and 55 sensors.

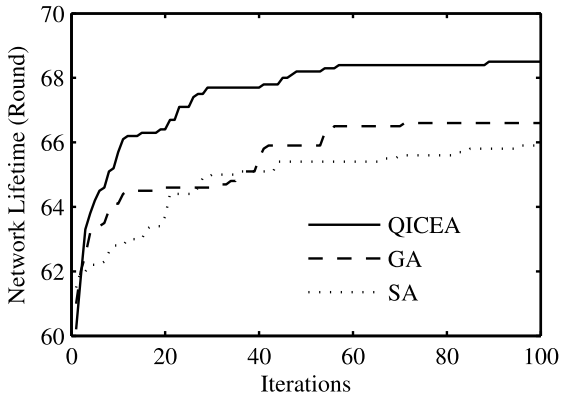


Figure 5.2: Network lifetime with 20 targets and 55 sensors.

5.5 Summary

Duty cycle design is a key problem in the operation of wireless sensor networks. In this chapter, we proposed QICEA for duty cycle design while maintaining full coverage in the monitoring area. Simulation results show that compared to SA and GA, the proposed QICEA can extend the lifetime of wireless sensor networks and enhance the energy efficiency effectively.

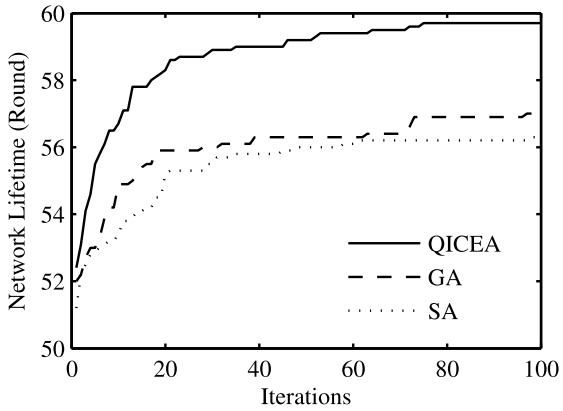


Figure 5.3: Network lifetime with 30 targets and 75 sensors.

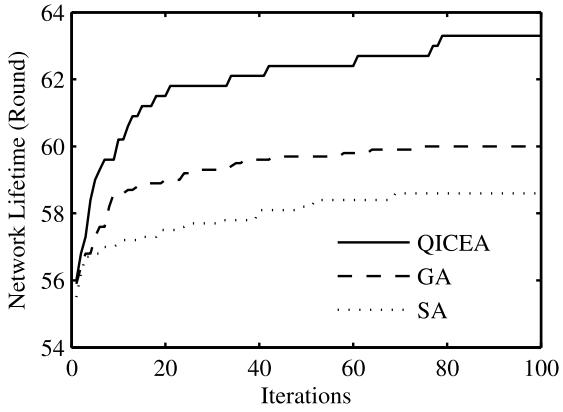


Figure 5.4: Network lifetime with 40 targets and 100 sensors.

Chapter 6

Low Energy Clustering in High-density WSN Based on FSEC

6.1 Introduction

With the continuous development of wireless sensor technology, more and more researchers pay attention to high-density wireless sensor networks. High-density wireless sensor networks have a wide range of applications in battlefield information collection, security systems, office automation and target localization, etc [126]. With the improvement of intelligence and the decrease of production costs, the number of sensors in high-density wireless sensor networks grows exponentially. In some networks, there are hundreds to thousands sensors working simultaneously. In these studies and applications, how to reduce the communication energy consumption by clustering is a key issue [127].

The most common clustering protocols for wireless sensor networks include LEACH and HEED. LEACH (Low Energy Adaptive Clustering Hierarchy) [128] is a low energy consumption clustering scheme that significantly improves network lifetime. The LEACH method uses a duty cycle cluster head selection method instead of fixed cluster heads.

HEED (Hybrid Energy-Efficient Distributed clustering approach) was first introduced in [129]. In HEED, the reserved energy and density of nodes were considered in cluster head selection. Both methods focus on maximum network lifetime instead of the minimum energy consumption of the whole network. These methods are generally more concerned about the reserved energy of nodes and the death time of the first battery-powered sensor node in the network. Therefore, the energy consumption is nearly uniformly distributed to all sensor nodes if the initial energy of all nodes are equal. With these methods, in most cases, the total energy consumption of the whole network is not minimized. Instead, the network lifetime is maximized as the nodes with more reserved energy have a greater chance of being selected as the cluster head.

In some high-density sensor networks, nodes are powered by wired power, although the communication is wireless. In this case, people are more concerned about the total energy consumption of the whole network per unit time. In other words, it is possible to reduce the power consumption and electricity bills.

In some high-density wireless sensor networks, there are hundreds of sensor nodes in the whole network. The energy consumption per unit of time for the whole network is high. How to select a certain percentage of cluster heads with the minimum energy consumption for data collection is an important issue. In [130], a method based on a simple genetic algorithm for data collection is proposed for minimum data collection energy consumption. However, the simple genetic algorithm easily falls into local optima and it often gets high energy consumption clustering results.

Based on the above studies, a Low Energy Clustering method of High-density wireless sensor networks based on fuzzy simulated evolutionary computation is proposed in this chapter. In order to reduce communication energy consumption, we also designed a fuzzy controller to dynamically adjust the crossover and mutation probability. Simulations are conducted by using the proposed method, the clustering methods based on particle swarm

optimization and the method based on quantum evolutionary algorithm. Results show that the energy consumption of the proposed method decreased compared to the other two methods which means that the proposed method improves the energy efficiency.

6.2 System Model

This chapter studies a class of high-density sensor networks, in which nodes are randomly distributed in the monitoring area. The sensor nodes within the monitoring area are divided into a number of clusters and there is a single cluster head node within each cluster.

In the uplink transmission phase, the randomly distributed sensor nodes observe and check the targets within the monitoring area and then send the perceived data to the cluster head node. The cluster head node collects the data from sensor nodes and then uploads the data to the gateway node directly. The gateway node collects the data from the cluster head nodes and transfers the data to the user for further analysis and processing.

In the downlink phase, users allocate monitoring tasks to the cluster head nodes through the gateway node and then the cluster head nodes assign the tasks to the sensor nodes within the clusters to complete the monitoring tasks. Generally, the amount of uplink data is much larger than the downlink data in monitoring tasks.

In order to reduce the energy consumption with the node communication distance limitation, we must develop an efficient clustering scheme for high-density wireless sensor networks. The energy consumption of a wireless sensor network consists of the communication energy, the sensing energy and the microprocessor energy. Studies show that the transmission radio energy and receive energy consumption accounted for more than half of the total energy consumption of wireless sensor networks. Meanwhile, the sensing en-

ergy consumption and the microprocessor energy consumption is relatively fixed which is difficult to reduce through optimization. So we focus on how to reduce the communication energy consumption via optimal clustering in high density sensor networks.

In high density sensor networks, the transmission radio energy can be shown as[131]:

$$cost_s(k,d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^n \tag{6.1}$$

where $cost_s(k,d)$ is the transmission radio energy, d is the distance between two nodes, k is the length of sending bits, n is the path-loss exponent, ε_{amp} is the power amplification parameter and E_{elec} is the electronics energy parameter. Depending on the communications environment, the general value of n is between 2-4. The value of n is higher when the communication environment is worse.

The receive energy of k bits data can be shown as:

$$cost_r(k) = E_{elec} \cdot k \tag{6.2}$$

where $cost_r(k)$ is the receiver dissipated energy for receiving k bits. In the actual system, E_{elec} for sending and receiving are not exactly the same. This thesis uses a simplified model and we assume that the E_{elec} for sending and receiving are the same.

In this chapter, we use k=1Mbits, ε_{amp} = 100pJ/bit/m^2, E_{elec}= 50nJ/bit and $n = 3$ [131].

6.3 Fuzzy Simulated Evolutionary Computation

Simulated evolutionary computation specifically includes evolution programming, evolution strategies, genetic programming and genetic algorithms. As genetic algorithms are most widely used among these numerical optimization methods, in this chapter, we use a fuzzy based genetic algorithm to develop an efficient clustering scheme for high density wireless sensor networks.

6.3.1 Adjusting the Algorithm Parameter

In fuzzy simulated evolutionary computation, the crossover probability and mutation probablity are key parameters that have a significant impact on the algorithm behavior and performance. When the crossover probability is greater, the algorithm would generate new individuals faster. However, when the crossover probability is too high, the structure of outstanding individuals in the population will be destroyed which leads to a slow algorithm convergence rate. If the crossover probability is too low, the search speed of the algorithm will become very slow. In this way, the individuals structure will be hard to destroy which will lead to evolutionary stagnation.

Similarly, when the mutation probability is greater, the algorithm tends to destroy the original structure of individuals in the population and generate new individuals. However, when the mutation probability is too high, the structure of outstanding individuals in the population will be destroyed which leads to convergence stagnation. If the mutation probability is too low, the search speed of the algorithm will become very slow. In this way, the individuals structure will be hard to destroy, which will lead to evolutionary stagnation.

Previous studies show that the adaptively adjustable crossover probability can significantly improve the diversity of the population and improve the convergence rate. To solve this problem, Srinvivas proposed an adaptive genetic algorithm in [132]. The adaptive genetic algorithm adaptively adjusts crossover and mutation probabilities with the individual fitness. The adjusting method is simple, however the accelerating effect is not obvious.

In this chapter, a fuzzy simulated evolutionary computation is proposed to automatically adjust the crossover probability and mutation probability. We also design a fuzzy controller based on the fuzzy control concept. In fuzzy simulated evolutionary computation, when the average fitness of the population is too high, the fuzzy controller adjusts

the crossover probability and mutation probability to a low value to avoid population diversity decline. Similarly, when the average fitness of the population is low, the fuzzy controller adjust crossover probability and mutation probability to a high value to increase the population diversity.

Meanwhile, for excellent individuals, the fuzzy controller adjusts the crossover probability and mutation probability to a lower value to keep the individual structure for the next iteration. Similarly, for low fitness individuals, the fuzzy controller will increase the crossover probability and mutation probability to improve the fitness of the individuals.

6.3.2 The Input and Output of the Fuzzy Controller

In the design of the fuzzy controller that adjusts the crossover probability and mutation probability, first we need to determine the input and output of the fuzzy controller and design the membership function for the fuzzy input and output.

In order to adjust the crossover probability and mutation probability, first we need to do normalizations for the average fitness of the population $\overline{f}$, the individual for mutation f_t and the individual with a smaller fitness value in the crossover operation f_b. The normalizations can be shown as

$$f_c = \frac{f_{\max} - f_b}{f_{\max} - f_{\min}} \tag{6.3}$$

$$f_m = \frac{f_{\max} - f_t}{f_{\max} - f_{\min}} \tag{6.4}$$

$$f_a = \frac{f_{\max} - \overline{f}}{f_{\max} - f_{\min}} \tag{6.5}$$

where f_c is the normalized fitness of the individual with a smaller fitness value in the crossover operation and its value range is $f_c \in [0, 1]$, f_m is the normalized fitness of the individual for mutation and its value range is $f_m \in [0, 1]$. f_a is the normalized average fitness of the population and its value range is $f_a \in [0, 1]$.

$f_{\max}$ is the fitness of the individual with the largest fitness value in the population and $f_{\min}$ is the fitness of the individual with the smallest fitness value in the population.

In the high-density sensor network clustering, the individual fitness is smaller when the network communication energy is smaller. For the controller of crossover probability P_c, the fuzzy controller input parameters includes the individual with a small fitness value in the crossover operation f_c and the normalized average fitness of the population f_a.

When f_c is greater, it indicates that the individual with a smaller fitness value in the crossover operation is better. Similarly, when f_a is greater, it indicates that the average fitness of individuals in the population $\overline{f}$ is smaller. According to the author's previous experience, the output value range of the controller of crossover probability is $P_c \in [0.65, 0.95]$.

For the controller of crossover probability P_m, the fuzzy controller input parameters include the individual in the mutation operation f_m and the normalized average fitness of the population f_a. When f_m is greater, it indicates that the individual in the mutation operation is better. Similarly, when f_a is greater, it indicates that the average fitness of individuals in the population $\overline{f}$ is smaller. According to previous experience, the output value range of the controller for the crossover probability is $P_m \in [0.05, 0.26]$.

6.3.3 The Membership Function

For crossover probability P_c, we use the triangular membership function and the Mamdani membership function for input value fuzzification. In order to design the triangular membership functions, we follow the specific design criteria presented in [133]. For the design of Mamdani membership functions, we follow the design criteria presented in [134].

For the normalized fitness of the individual with a smaller fitness value in the crossover operation f_c, we choose the triangular membership function for fuzzification and select six fuzzy sets, including very small (FVS), small (FS), relatively small (FRS), relatively

big (FRB), big (FB) and very big (FVB) for the input value range $f_c \in [0,1]$. The first letter F is used to distinguish different kinds of normalized fitness of the individuals.

$$
\begin{aligned}
FVS(f_c) &= \begin{cases} -5f_c + 1 & 0 \le f_c \le 0.2 \\ 0 & f_c > 0.2 \end{cases} \\
FS(f_c) &= \begin{cases} 5f_c & 0 \le f_c \le 0.2 \\ -5f_c + 2 & 0.2 < f_c \le 0.4 \\ 0 & f_c > 0.4 \end{cases} \\
FRS(f_c) &= \begin{cases} 0 & f_c < 0.2 \\ 5f_c - 1 & 0.2 \le f_c \le 0.4 \\ -5f_c + 3 & 0.4 < f_c \le 0.6 \\ 0 & f_c > 0.6 \end{cases} \\
FRB(f_c) &= \begin{cases} 0 & f_c < 0.4 \\ 5f_c - 2 & 0.4 \le f_c \le 0.6 \\ -5f_c + 4 & 0.6 < f_c \le 0.8 \\ 0 & f_c > 0.8 \end{cases} \\
FB(f_c) &= \begin{cases} 0 & f_c < 0.6 \\ 5f_c - 3 & 0.6 \le f_c \le 0.8 \\ -5f_c + 5 & 0.8 < f_c \le 1 \end{cases} \\
FVB(f_c) &= \begin{cases} 5f_c - 4 & 0.8 \le f_c \le 1 \\ 0 & f_c < 0.8 \end{cases}
\end{aligned}
\tag{6.6}
$$

.

The triangular membership functions of these fuzzy sets are distributed as shown in Fig.6.1.

For the normalized average fitness of the population f_a, we choose the Mamdani

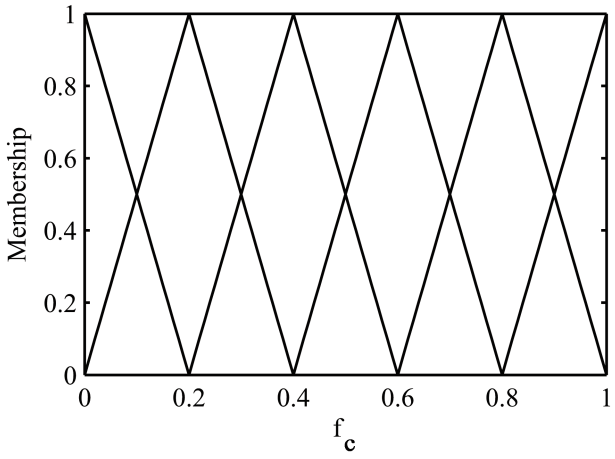


Figure 6.1: The triangular membership functions for f_c

membership function for fuzzification and select six fuzzy sets, including very small (AVS), small (AS), relatively small (ARS), relatively big(ARB), big (AB) and very big (AVB) for the of input value range $f_a \in [0, 1]$. For the Mamdani membership function, if the membership value is great than 1, the membership value equals to 1. The first letter A is used to distinguish different kinds of normalized fitness of the individuals.

$$\begin{aligned}
AVS(f_a) &= 1 - e^{-\frac{0.5}{|10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1 \\
AS(f_a) &= 1 - e^{-\frac{0.5}{|2 - 10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1 \\
ARS(f_a) &= 1 - e^{-\frac{0.5}{|4 - 10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1 \\
ARB(f_a) &= 1 - e^{-\frac{0.5}{|6 - 10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1 \\
AB(f_a) &= 1 - e^{-\frac{0.5}{|8 - 10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1 \\
AVB(f_a) &= 1 - e^{-\frac{0.5}{|10 - 10 f_a|^{2.5}}} \qquad 0 \le f_a \le 1
\end{aligned} \tag{6.7}$$

.

The Mamdani membership functions of these fuzzy sets are distributed as shown in Fig.6.2.

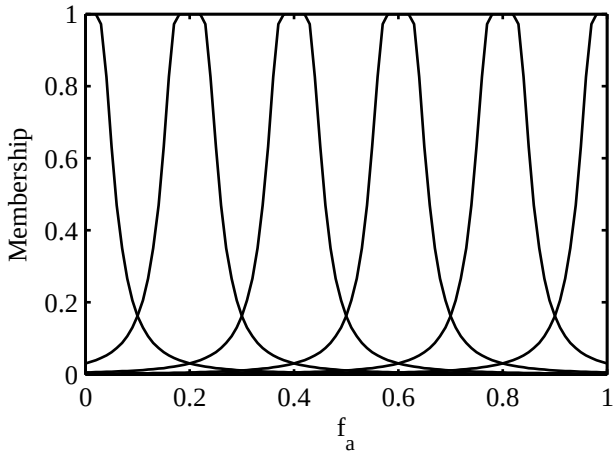


Figure 6.2: The Mamdani membership functions for f_a

For the output of the fuzzy controller that adjusts P_c, we use a Gaussian fuzzy membership function for the defuzzification. The defuzzification includes seven fuzzy sets: very small (GVS), small (GS), relatively small (GRS), medium (GM), relatively big(GRB), big (GB) and very big (GVB) for the of input value range $P_c \in [0.65, 0.95]$.The first letter G is used to distinguish different kinds of normalized fitness of the individuals.

$$
\begin{aligned}
GVS(P_c) &= e^{-\frac{(P_c-0.65)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GS(P_c) &= e^{-\frac{(P_c-0.7)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GRS(P_c) &= e^{-\frac{(P_c-0.75)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GM(P_c) &= e^{-\frac{(P_c-0.8)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GRB(P_c) &= e^{-\frac{(P_c-0.85)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GB(P_c) &= e^{-\frac{(P_c-0.9)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95 \\
GVB(P_c) &= e^{-\frac{(P_c-0.95)^2}{2\times 0.02^2}} && 0.65 \le P_c \le 0.95
\end{aligned}
\tag{6.8}
$$

.

The Gaussian membership functions of these fuzzy sets are distributed as shown in

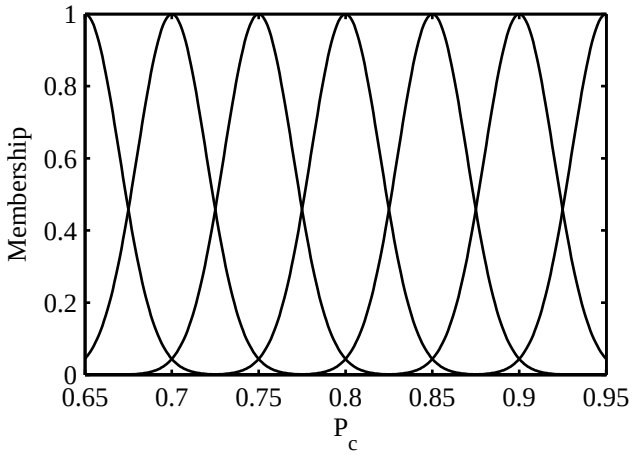


Figure 6.3: The Gaussian membership functions for P_c

Fig.6.3.

For mutation probability P_m, we also use the triangular membership function and the Mamdani membership function for input value fuzzification.

For the normalized fitness of the individual in the mutation operation f_m, we choose the triangular membership function for fuzzification, and select six fuzzy sets, including very small (DVS), small (DS), relatively small (DRS), relatively big(DRB), big (DB) and very big (DVB) for the input value range $f_c \in [0, 1]$. The first letter D is used to distinguish different kinds of normalized fitness of the individuals. The fuzzification process for f_a is the same as the process in the controller for crossover probability P_c.

$$
\begin{aligned}
DVS(f_m) &= \begin{cases} -5f_m + 1 & 0 \le f_m \le 0.2 \\ 0 & f_m > 0.2 \end{cases} \\
DS(f_m) &= \begin{cases} 5f_m & 0 \le f_m \le 0.2 \\ -5f_m + 2 & 0.2 < f_m \le 0.4 \\ 0 & f_m > 0.4 \end{cases} \\
DRS(f_m) &= \begin{cases} 0 & f_m < 0.2 \\ 5f_m - 1 & 0.2 \le f_m \le 0.4 \\ -5f_m + 3 & 0.4 < f_m \le 0.6 \\ 0 & f_m > 0.6 \end{cases} \\
DRB(f_m) &= \begin{cases} 0 & f_m < 0.4 \\ 5f_m - 2 & 0.4 \le f_m \le 0.6 \\ -5f_m + 4 & 0.6 < f_m \le 0.8 \\ 0 & f_m > 0.8 \end{cases} \\
DB(f_m) &= \begin{cases} 0 & f_m < 0.6 \\ 5f_m - 3 & 0.6 \le f_m \le 0.8 \\ -5f_m + 5 & 0.8 < f_m \le 1 \end{cases} \\
DVB(f_m) &= \begin{cases} 5f_m - 4 & 0.8 \le f_m \le 1 \\ 0 & f_m < 0.8 \end{cases}
\end{aligned}
\tag{6.9}
$$

.

The triangular membership functions of these fuzzy sets are distributed as shown in Fig.6.4.

For the output of the fuzzy controller that adjusts P_m, we use a bell-shaped membership function for the defuzzification. The defuzzification includes eight fuzzy sets: extremely small (OES), very small (OVS), small (OS), relatively small (ORS), medium

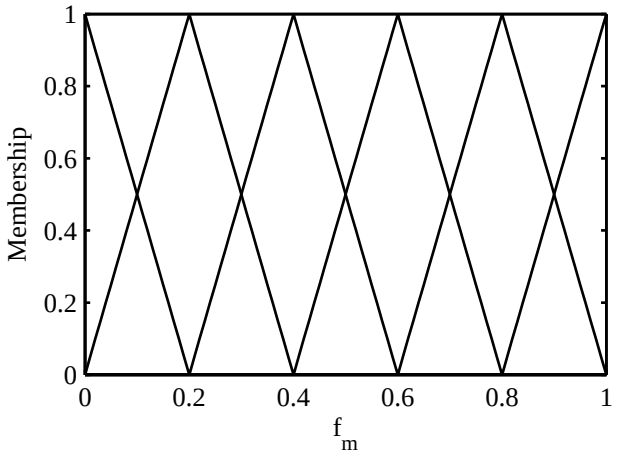


Figure 6.4: The triangular membership functions for f_m

(OM), relatively big (ORB), big (OB) and very big (OVB) for the of input value range $P_m \in [0.05, 0.26]$. The first letter O is used to distinguish different kinds of normalized fitness of the individuals.

$$
\begin{aligned}
OES(P_m) &= \frac{1}{1+\left|\frac{10P_m-0.5}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
OVS(P_m) &= \frac{1}{1+\left|\frac{10P_m-0.8}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
OS(P_m) &= \frac{1}{1+\left|\frac{10P_m-1}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
ORS(P_m) &= \frac{1}{1+\left|\frac{10P_m-1.4}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
OM(P_m) &= \frac{1}{1+\left|\frac{10P_m-1.7}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
ORB(P_m) &= \frac{1}{1+\left|\frac{10P_m-2}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
OB(P_m) &= \frac{1}{1+\left|\frac{10P_m-2.3}{0.1}\right|^2} && 0.05 \le P_m \le 0.26 \\
OVB(P_m) &= \frac{1}{1+\left|\frac{10P_m-2.6}{0.1}\right|^2} && 0.05 \le P_m \le 0.26
\end{aligned}
\tag{6.10}
$$

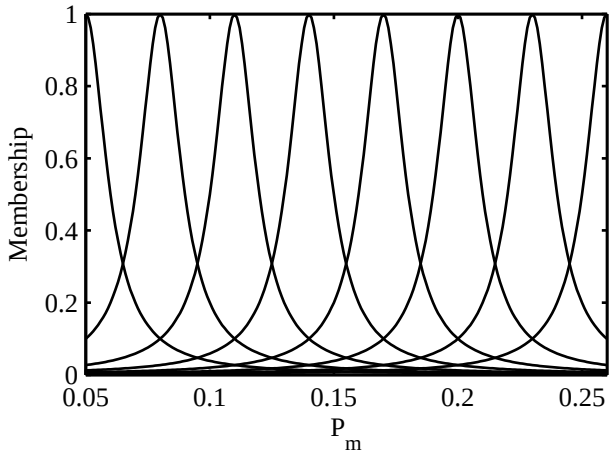


Figure 6.5: The bell-shaped membership functions for P_m

.

The bell-shaped membership functions of these fuzzy sets are distributed as shown in Fig.6.5.

6.3.4 The Fuzzy Rules and Fuzzy Implication

The previous experience for the parameter adjustment can be summarized as the following fuzzy rules [135]:

(I) " "IF f_c is bigger, THEN the crossover probability P_c is smaller"

(II) "IF f_m is bigger, THEN the crossover probability P_m is smaller"

(III) "IF f_a is bigger, THEN the crossover probability P_c is bigger"

(IV) "IF f_a is bigger, THEN the crossover probability P_m is bigger"

According to the fuzzy rule (I) and (III), we designed a double input-single output (DISO) fuzzy rule table, which can be shown in Table 6.1. As the number of input fuzzy set for both f_c and f_a is 6, the number of fuzzy rules in Table 6.1 increases up to 36 for the output P_c.

The numbers in the table are used to indicate the order of the rules. To assess the influence of each rules, we use the fuzzy implication $R_i(i = 1, 2, \cdots, 36)$ to express the i-th rule for adjusting the crossover probability P_c. Once all the contributions of the 36 implications are determined we can aggregate these 36 fuzzy implications into one output fuzzy implication R, namely

$$R = R_1 \bigcup R_2 \bigcup \cdots R_{36} = \bigcup_{i=1}^{36} R_i \tag{6.11}$$

.

Each fuzzy implication can be expressed by the inference "IF f_c is A and f_a is B, THEN P_c is U" that is activated by the input values f_c and f_a, namely $A(f_c) \wedge B(f_a) \rightarrow U(P_c)$.

As we apply the minimum (Mamdani) implication for each $A(f_c) \wedge B(f_a) \rightarrow U(P_c)$, the text form and the corresponding activation degrees of the consequent parts of these 36 rules become:

(1) IF f_c is FVB and f_a is AVS, THEN P_c is GVS, $R_1(f_c, f_a, P_c) = FVB(f_c) \wedge AVS(f_a) \wedge GVS(P_c)$

(2) IF f_c is FB and f_a is AVS, THEN P_c is GVS, $R_2(f_c, f_a, P_c) = FB(f_c) \wedge AVS(f_a) \wedge GVS(P_c)$

Table 6.1: Fuzzy rules for adjusting the crossover probability P_c

	f_c					
f_a	FVB	FB	FRB	FRS	FS	FVS
AVS	(1)GVS	(2)GVS	(3)GS	(4)GRS	(5)GM	(6)GRB
AS	(7)GVS	(8)GS	(9)GRS	(10)GM	(11)GRB	(12)GB
ARS	(13)GS	(14)GS	(15)GRS	(16)GM	(17)GRB	(18)GB
ARB	(19)GS	(20)GRS	(21)GM	(22)GRB	(23)GB	(24)GB
AB	(25)GS	(26)GRS	(27)GM	(28)GRB	(29)GB	(30)GVB
AVB	(31)GRS	(32)GM	(33)GRB	(34)GB	(35)GVB	(36)GVB

...

(36) IF f_c is FVS and f_a is AVB, THEN P_c is GVB, $R_{36}(f_c, f_a, P_c) = FVS(f_c) \wedge AVB(f_a) \wedge GVB(P_c)$

According to the fuzzy rules (II) and (IV), we design a double input-single output (DISO) fuzzy rule table which can be shown in Table 6.2. As the number of input fuzzy set for both f_m and f_a is 6, the number of the fuzzy rules in Table 6.2 increases up to 36 for the output P_m.

Table 6.2: Fuzzy rules for adjusting the crossover probability P_m

	f_m					
f_a	DVB	DB	DRB	DRS	DS	DVS
AVS	(1)OES	(2)OES	(3)OVS	(4)OS	(5)ORS	(6)OM
AS	(7)OES	(8)OVS	(9)OS	(10)ORS	(11)OM	(12)ORB
ARS	(13)OVS	(14)OS	(15)ORS	(16)OM	(17)ORB	(18)OB
ARB	(19)OVS	(20)OS	(21)OM	(22)ORB	(23)ORB	(24)OB
AB	(25)OS	(26)ORS	(27)OM	(28)ORB	(29)OB	(30)OVB
AVB	(31)ORS	(32)OM	(33)ORB	(34)OB	(35)OVB	(36)OVB

To assess the influence of each rule $G_i(i = 1, 2, \cdots, 36)$, we use the fuzzy implication $G_i(i = 1, 2, \cdots, 36)$ to express the i-th rule for adjusting the mutation probability P_m. Once all the contributions of the 36 implications are determined we can aggregate these 36 fuzzy implications into one output fuzzy implication G, namely

$$G = G_1 \bigcup G_2 \bigcup \cdots G_{36} = \bigcup_{i=1}^{36} G_i \tag{6.12}$$

.

Each fuzzy implication can be expressed by the inference "IF f_m is A and f_a is B, THEN P_m is U that is activated by the input value f_m and f_a, namely $A(f_m) \wedge B(f_a) \rightarrow U(P_m)$. As we apply the minimum (Mamdani) implication for each $A(f_m) \wedge B(f_a) \rightarrow U(P_m)$, the text form and the corresponding activation degrees of the consequent parts of these 36 rules become:

(1)IF f_m is DVB and f_a is AVS, THEN P_m is OES, $G_1(f_m, f_a, P_m) = DVB(f_m) \wedge AVS(f_a) \wedge OES(P_m)$

(2)IF f_m is DB and f_a is AVS, THEN P_m is OES, $G_2(f_m, f_a, P_m) = DB(f_m) \wedge AVS(f_a) \wedge OES(P_m)$

...

(36)IF f_m is DVS and f_a is AVB, THEN P_m is OVB, $G_{36}(f_m, f_a, P_m) = DVS(f_m) \wedge AVB(f_a) \wedge OVB(P_m)$

6.3.5 Defuzzification

According to Zadeh [137], the fuzzy interpretation of the i-th rule for P_c can be presented as

(1) $U_1(P_c) = (FVB(f_{c0}) \wedge AVS(f_{a0})) \circ R_1(f_{c0}, f_{a0}, P_c)$

(2) $U_2(P_c) = (FB(f_{c0}) \wedge AVS(f_{a0})) \circ R_2(f_{c0}, f_{a0}, P_c)$

...

(36) $U_{36}(P_c) = (FVS(f_{c0}) \wedge AVB(f_{a0})) \circ R_{36}(f_{c0}, f_{a0}, P_c)$

where the operator $\circ$ indicate the minimum implication for the corresponding elements and R_i is defined in Section 6.3.4. We use the max-min aggregation operation to get the output fuzzy set of P_c, which can be shown as

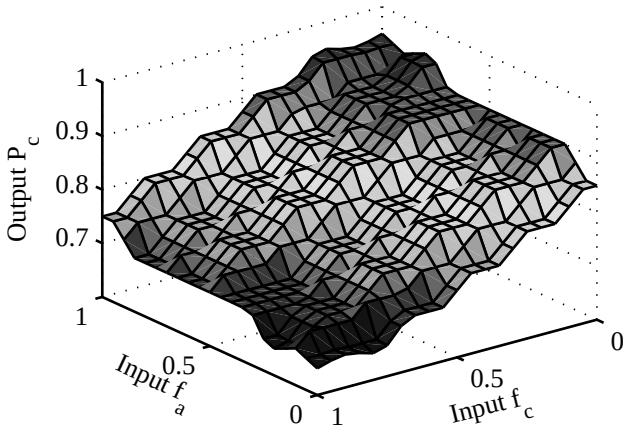


Figure 6.6: The output P_c of the fuzzy controller

$$U^*(P_c) = \bigcup_{i=1}^{36} U_i(P_c) \tag{6.13}$$

where U^* is the membership function of the resultant output fuzzy set of P_c.

In the defuzzification we use the Mean of Maximum (MOM) method to calculate the control output of P_c and the result can be shown as Fig 6.6.

From the figure we can see that in the input value range $f_c \in [0,1]$ and $f_a \in [0,1]$, the output P_c meets the previous experience (I) and (III) for the parameter adjustment, which are "IF f_m is bigger, THEN the crossover probability P_m is smaller" and "IF f_a is bigger, THEN the crossover probability P_c is bigger".

Similarly, the fuzzy interpretation of the i-th rule for P_m can be presented as

(1) $U_1(P_m) = (DVB(f_{m0}) \wedge AVS(f_{a0})) \circ G_1(f_{m0}, f_{a0}, P_m)$

(2) $U_2(P_m) = (DB(f_{m0}) \wedge AVS(f_{a0})) \circ G_2(f_{m0}, f_{a0}, P_m)$

...

(36) $U_{36}(P_m) = (DVS(f_{m0}) \wedge AVB(f_{a0})) \circ G_{36}(f_{m0}, f_{a0}, P_m)$

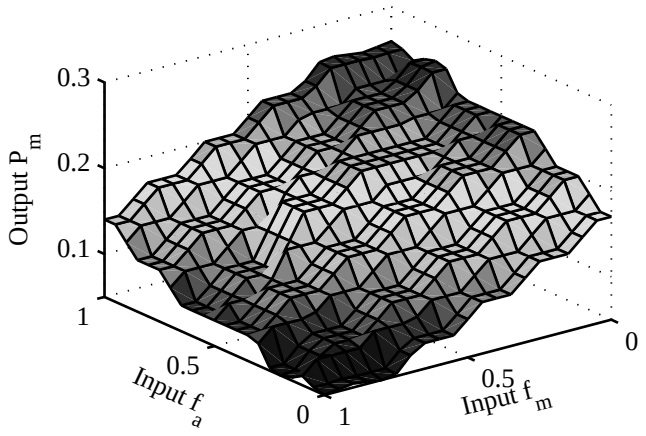


Figure 6.7: The output P_m of the fuzzy controller

where G_i is defined in Section 6.3.4. We use the max-min aggregation operation to get the output fuzzy set of P_m, which can be shown as

$$U^*(P_m) = \bigcup_{i=1}^{36} U_i(P_m) \tag{6.14}$$

where U^* is the membership function of the resultant output fuzzy set of P_m.

In the defuzzification we use the Mean of Maximum (MOM) method to calculate the control output of P_m and the result can be shown as Fig 6.7.

From the figure we can see that in the input value range $f_c \in [0, 1]$ and $f_a \in [0, 1]$, the output P_m meets the previous experience (II) and (IV) for the parameter adjustment, which are "IF f_m is bigger, THEN the crossover probability P_m is smaller" and "IF f_a is bigger, THEN the crossover probability P_m is bigger".

6.3.6 Computational Complexity

For the computational complexity analysis, the two fuzzy controllers are equivalent to two dual-input single-output functions $g_1(f_c, f_a) = P_c$ and $g_2(f_m, f_a) = P_m$. The output

P_c is calculated in advance and stored in a table for each input (f_c, f_a) that is rounded to a certain precision. Similarly, the output P_m is calculated in advance and stored in a table for each input (f_m, f_a) that is rounded to a certain precision. In this way the computational complexity of both controller are $O(1)$.

6.4 Low Energy Clustering Based on Fuzzy Simulated Evolutionary Computation in High-density WSNs

In a high-density WSN, the traditional clustering methods lack an overall consideration so there are some sensors far from the cluster head which result in energy waste. The clustering method that is proposed in this chapter establishes a global unity mechanism for minimizing communication energy consumption which considers the position of the cluster heads and the sensor nodes. The main steps of our methods are: individual encoding and initial population generation, design of the fitness function, selection, crossover and mutation and adaptive adjustment of parameters based on the fuzzy controller.

6.4.1 Population Encoding and Initialization

In Fuzzy Simulated Evolutionary Computation, the encoded solutions are represented by chromosomes. First, we conduct a natural number coding for all the nodes within the region with natural numbers 1 to L. In this way, each individual can be encoded as a vector of binary numbers, with the same length as the number of nodes L. Each bit of the individual is composed of a Boolean variable stating whether the corresponding sensor node is selected as the cluster head node or not. “1” represents that the node is selected as a cluster head node in the corresponding position and “0” represents that the node is

selected as a sensor node in the corresponding position. Each sensor node only clusters with the nearest cluster head. For example, if there are 8 nodes in the region and No.3, No.5 and No.7 nodes are selected as clusterhead nodes, the individuals can be represented as 00101010.

We use a population with a fixed number of individuals. The population can be shown as

$$\begin{aligned} P &= \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,L-1} & e_{1,L} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,L-1} & e_{2,L} \\ \vdots & & e_{n,l} & & \vdots \\ e_{N-1,1} & e_{N-1,2} & \cdots & e_{N-1,L-1} & e_{N-1,L} \\ e_{N,1} & e_{N,2} & \cdots & e_{N,L-1} & e_{N,L} \end{bmatrix} \\ &= \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{N-1} \\ E_N \end{bmatrix} \qquad (e_{n,l} \in \{0,1\}) \end{aligned} \tag{6.15}$$

.

where L is the number of nodes in the sensor networks, N is the number of individuals in the population, M is the number of clusterheads. $e_{n,l}$=1 represents that the n-th node of the l-th individual is the clusterhead node and $e_{n,l}$= 0 otherwise.

We assume that there are a fixed number of clusterhead nodes in the sensor network so the constraint can be shown as

$$\sum_{l=1}^{L} e_{n,l}=M \quad (n \in \{1,2,\cdots,N\}) \tag{6.16}$$

.

where M is the number of clusterhead nodes.

6.4.2 Fitness Function

Each individual in the population is awarded a score depending on the communications energy consumption. The fitness function of a individual is defined as

$$Fit(E) = \sum_{l=1}^{L} (cost_s + cost_r) \tag{6.17}$$

.

where $E_n = \begin{bmatrix} e_{n,1} & e_{n,2} & \cdots & e_{n,L-1} & e_{n,L} \end{bmatrix}$ is the fitness of the n-th individual of the populations, and the fitness equals to the communication energy consumption of the n-th clustering scheme. By this definition, the individual with a smaller fitness in the population is the better one. The individuals with the lower fitness value are more likely to be parents in the next generation.

6.4.3 Selection

We use the roulette wheel proportionate selection strategy for the selection operator. The selection operator selects an individual from the current population for the next population with the probability inversely proportional to its fitness value which can be shown as

$$P_{SELECT}(E_n) = \frac{\frac{1}{Fit(E_n)}}{\sum_{n=1}^{N} \frac{1}{Fit(E_n)}} \tag{6.18}$$

.

In this way, the individuals that have a lower communication energy consumption will have a higher probability to be selected.

6.4.4 Crossover

In the crossover operation, two new offspring individuals are generated from each pair of selected parent individuals. In order to keep the number of clusterhead nodes fixed, we design a crossover operator Based on a Boolean operation.

Firstly, we apply the logical AND operation to each pair of selected parent individuals and obtain an intermediate binary vector E'. The length of binary vector E' is equal to the number of sensor nodes L. For example, when the pair of selected parent individuals is E_1= [00101010] and E_2= [00100101], the intermediate binary vector is E' = [00100000].

Secondly, we apply the logical XOR operation to each pair of selected parent individuals and obtain another intermediate binary vector E''. The length of binary vector E'' is equal to the number of sensor nodes L. For example, when the pair of the selected parent individuals is E_1= [00101010] and E_2= [00100101], the other intermediate binary vector is E''= [00001111].

Finally, we "average random" assign "1" in individual E'' to the corresponding position of E' to generate two new offspring individuals. "Average" means that the number of "1"s assigned to each individual is equal. For example, when E' = [00100000] and E''= [00001111], the possible two new offspring individuals can be E_1^{new}= [00101100] and E_2^{new}= [00100011].

6.4.5 Mutation Operation

In order to add some diversity to the population and keep the number of the cluster head nodes fixed some random mutations are applied to the individuals in the population. The mutation operation is done by randomly exchanging the position of "1" and "0" in an individual with a certain probability. For example, the individual E= [00101100] can be mutated to E= [00101001], which means that "1" in the 6th bit and 0 in 8th bit are

exchanged.

6.4.6 Fuzzy Adjustment of the Algorithm Parameters

In the traditional genetic algorithm, crossover and mutation probabilities of selection have a great influence on the performance of the algorithm. We use the adjusting method presented in Section 6.3.1.

6.5 Simulation Results and Discussion

In this section the proposed fuzzy simulated evolutionary computation is tested with different sensor nodes and clusterhead proportions for the low energy clustering problem in high-density WSNs. Simulation experiments were conducted to verify the communication energy reduction of the proposed clustering method. All the cases are run in MATLAB 7.0 by a computer with 2.4GHz Intel Core i5 CPU and 2GB RAM. In the simulations, the monitoring area is $100m \times 100m$, and the nodes in a high-density sensor network are randomly distribute within the area. The gateway node is located at $(50, 50)$. As the communications energy consumption of uplink is much larger than downlink in most WSNs, we only consider the uplink communications energy consumption. In the simulations, we set $k = 1$Mbps, $n = 3$, $E_{elec} = 50$nJ/bit, $\varepsilon_{amp} = 100$pJ/bit/m^2.

Comparisons are made with PSO and QEA by only considering the best solution in the population. The fitness function in Equation 6.17 is used to evaluate their performance. The parameters employed for the fuzzy simulated evolutionary computation, including the number of iterations and the population size, are the same as those of PSO and QEA. In the fuzzy simulated evolutionary computation, the output value range of the fuzzy controller of the crossover probability is $P_c \in [0.65, 0.95]$ and the output value range of the controller of the crossover probability is $P_m \in [0.05, 0.26]$. In QEA, we use the same

lookup table as in [136]. In PSO, the cognitive and social parameters are set to 2, and the maximum velocity is set to 6.

Fig.6.8 to Fig. 6.11 show the communication energy consumption of a high density WSN using the fuzzy simulated evolutionary computation, PSO and QEA with 100, 200, 300 and 400 nodes respectively, where the energy consumption is obtained directly from Formulas 6.1 and 6.2. The results are averaged over 10 runs with different random nodes distributions. According to these comparisons, the proposed fuzzy simulated evolutionary computation has a lower communication energy consumption than PSO and QEA with different numbers of nodes after 100 iterations. As it may be observed in the figures, in the beginning, the communication energy consumption of all the algorithms decreased. However, after only a few iterations QEA falls into an evolutionary stagnation. From the figures we can also see that PSO is much slower in convergence compared with the fuzzy simulated evolutionary computation. By comparing the energy consumption used for communications the energy consumption of the fuzzy simulated evolutionary computation is much less than that of PSO and QEA. More specifically, the communications energy consumption of the fuzzy simulated evolutionary computation is 4.6% to 9.8% less than that of PSO and 20.6% to 29.7% less than that of QEA for different numbers of nodes.

Fig. 6.12 to Fig. 6.15 show the communication energy consumption of a high density WSN using the fuzzy simulated evolutionary computation, PSO and QEA with the clusterhead proportion of 5%, 10%, 15% and 20%, respectively where the number of the nodes increased from 200 to 1200. The results are the average energy consumption of 10 different random nodes positions. According to these comparisons the proposed fuzzy simulated evolutionary computation has a lower communications energy consumption than PSO and QEA with different clusterhead proportions and node numbers. More specifically, the communications energy consumption of the fuzzy simulated evolutionary computation is 2.34% to 36.02% lower than that of PSO and 18.41% to 61.31% lower than that

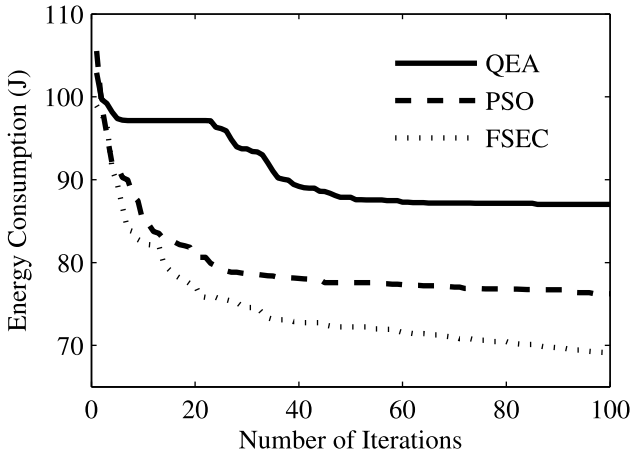


Figure 6.8: Communication energy consumption by iteration with 100 nodes and cluster head proportion of 10%

of QEA for different clusterhead proportions and node numbers.

Fig. 6.16 to Fig. 6.19 show the average node communication energy consumption of the high-density sensor network using the fuzzy simulated evolutionary computation, PSO and QEA with clusterhead proportions of 5%, 10%, 15% and 20% respectively where the number of the nodes increased from 200 to 1200. The results are the average energy consumption over 10 moves with different random nodes positions. According to these comparisons, the proposed fuzzy simulated evolutionary computation has a lower average node communications energy consumption than PSO and QEA with different clusterhead proportions and node numbers.

6.6 Conclusion

In this chapter, we proposed a fuzzy simulated evolutionary computation clustering method to reduce the communications energy consumption for high-density sensor networks. The clustering method establishes a global mechanism for minimizing communications energy

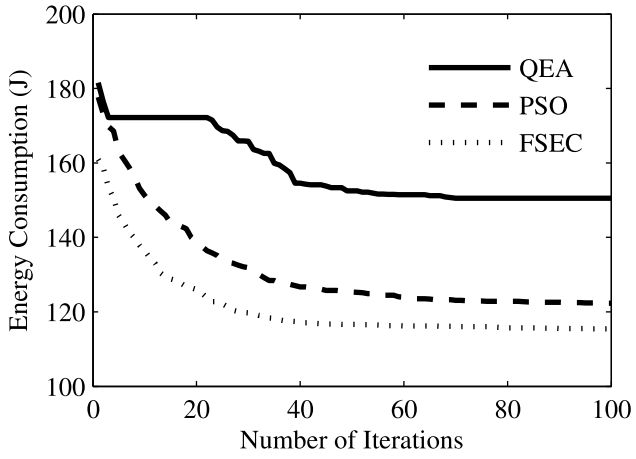


Figure 6.9: Communication energy consumption by iteration with 200 nodes and cluster head proportion of 10%

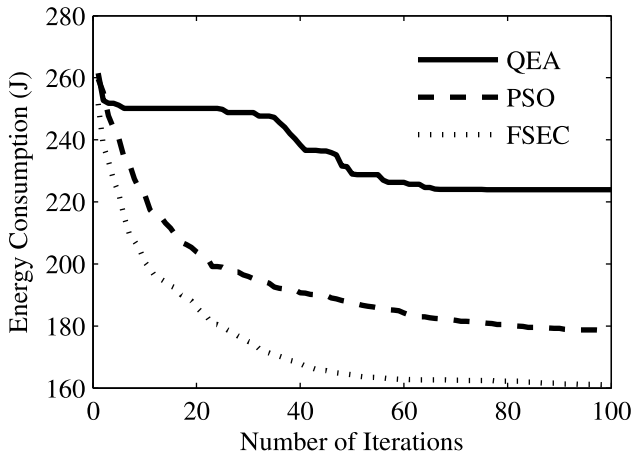


Figure 6.10: Communication energy consumption by iteration with 300 nodes and cluster head proportion of 10%

consumption which considers the position of the clusterheads and the sensor nodes. In order to reduce the communications energy consumption we also designed a fuzzy controller to dynamically adjust the crossover and mutation probabilities. Our simulation results show that the communications energy consumption of the proposed method decreased

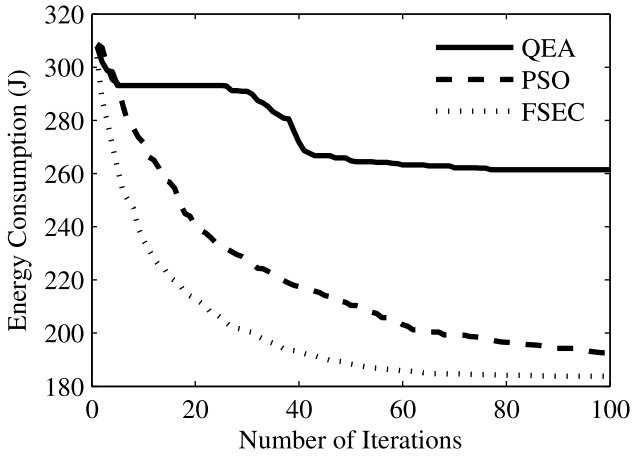


Figure 6.11: Communication energy consumption by iteration with 400 nodes and cluster head proportion of 10%

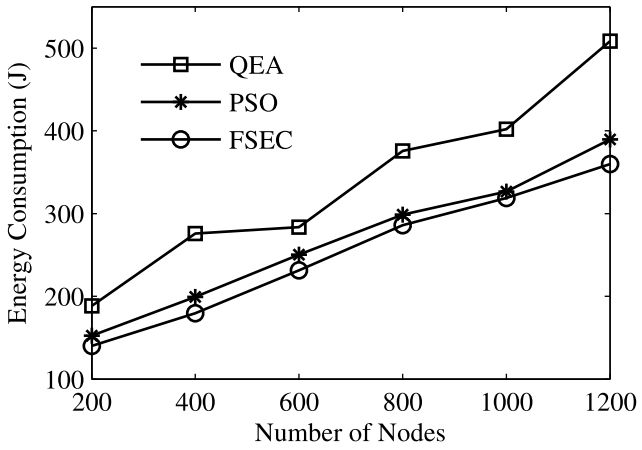


Figure 6.12: Communication energy consumption with clusterhead proportion of 5%.

compared to the other two methods which means that the proposed method significantly improves energy efficiency.

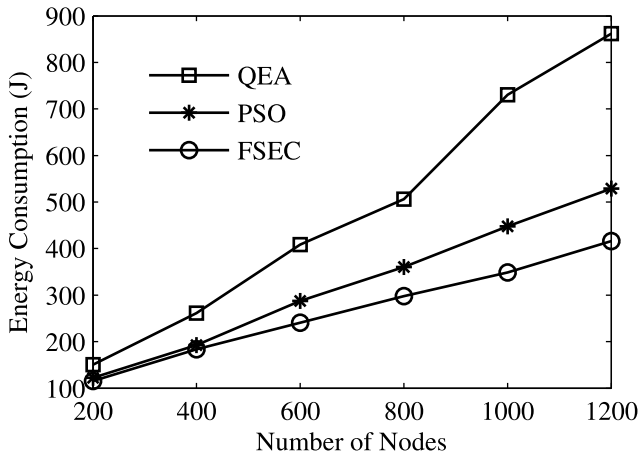


Figure 6.13: Communication energy consumption with clusterhead proportion of 10%.

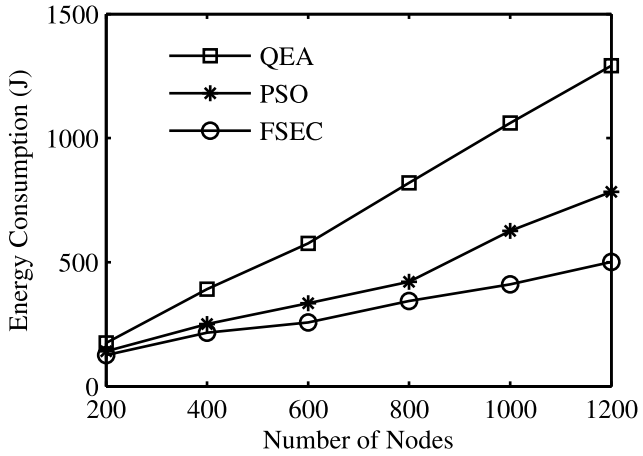


Figure 6.14: Communication energy consumption with clusterhead proportion of 15%.

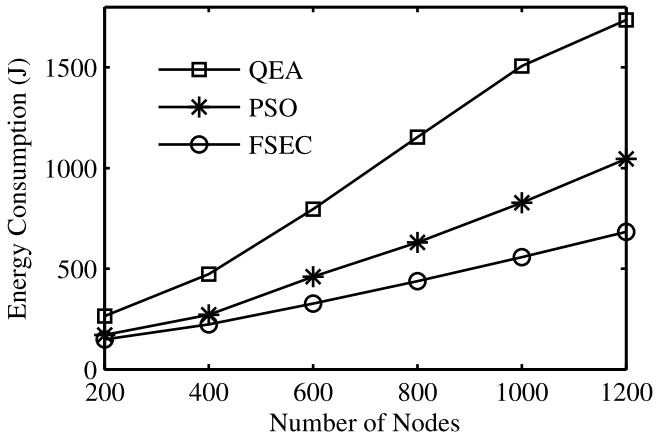


Figure 6.15: Communication energy consumption with clusterhead proportion of 20%.

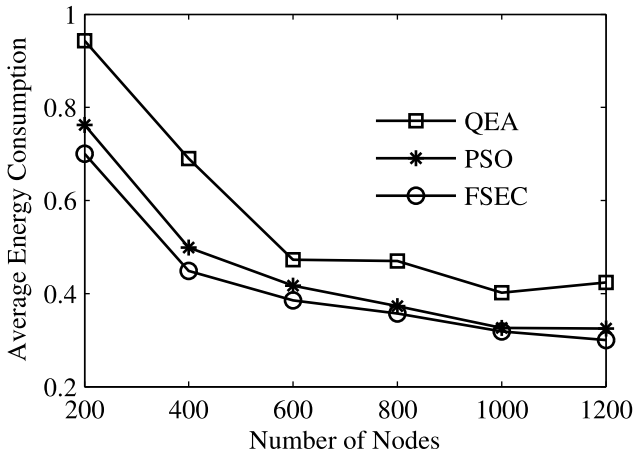


Figure 6.16: Average node communication energy consumption with clusterhead proportion of 5%.

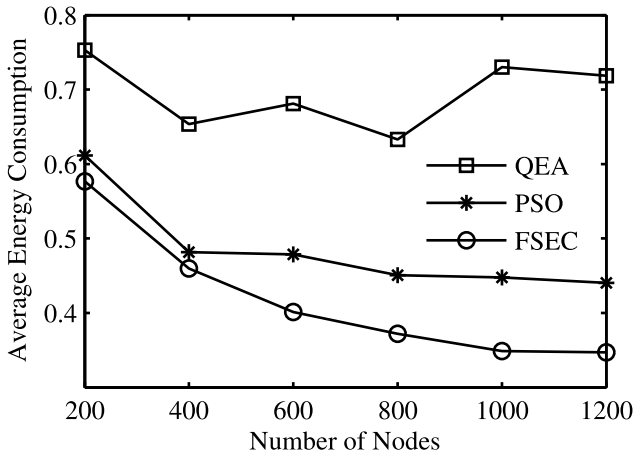


Figure 6.17: Average node communication energy consumption with clusterhead proportion of 10%.

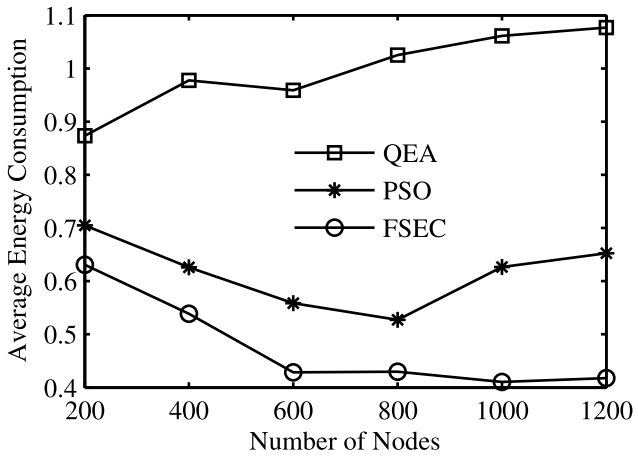


Figure 6.18: Average node communication energy consumption with clusterhead proportion of 15%.

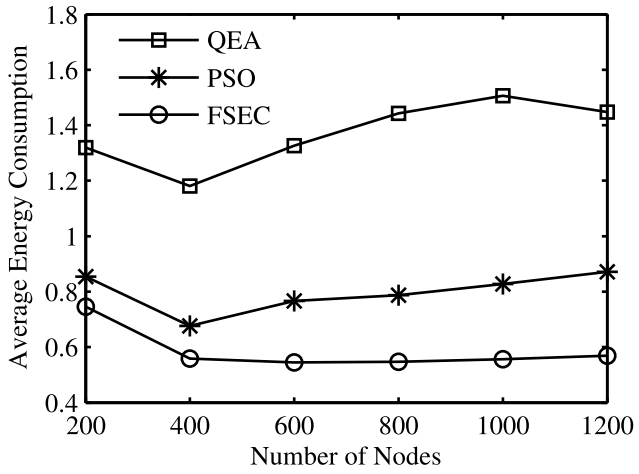


Figure 6.19: Average node communication energy consumption with clusterhead proportion of 20%.

Chapter 7

QoS Routing Based on Parallel Elite Clonal Quantum Evolution for Multimedia Wireless Sensor Networks

7.1 Introduction

With the advances in multimedia WSNs, multi-constraints QoS routing algorithms are becoming a hot topic in both research and application areas [10]. As there are many QoS constraints in WSNs, the "Best-Effort" mechanism cannot meet the needs of multimedia applications and the traditional Shortest Path First Algorithm is no longer applicable.

Many QoS based routing algorithms have been proposed, including delay constraint routing [138]-[139] and energy efficient routing [140]-[142]. In the multi-constraints QoS unicast routing problem, an optimal path with minimum energy consumption and multiple QoS constraints is selected that is able to lower the overall network energy consumption

while guaranteeing the communication quality of multimedia sensor nodes. However, when the number of nodes in multimedia WSNs is large, the complexity of the multi-constraints QoS unicast routing problem increases exponentially and the problem becomes an NP-hard nonlinear mixed integer combinatorial optimization problem [143].

In previous research, a number of heuristic optimization methods have been proposed for the QoS routing problem in multimedia WSNs, such as genetic algorithms [53][144], ant colony optimization [145], random walk routing [146] and opportunistic routing [147]. In [53] a GA based routing protocol was investigated for the routing problem with energy harvesting constraints. The GA based technique performs well at the first few generations. However, it suffers from stagnation after only a few generations. In [145] an ACO (Ant Colony Optimization) algorithm for path selection was proposed. This work focused on optimizing the user utility and the network provider utility with network QoS constraints. The ACO approach is simple and fast but suffers from premature convergence. A random walk routing method was introduced in [146] to minimize the cost and delay. The proposed method has shown good results in terms of energy saving and latency reduction. However, the reference does not consider factors such as packet loss, delay jitter and bandwidth constraints.

Moreover, for multimedia sensor networks, the real-time and bandwidth requirements for multimedia service delivery are more challenging than for ordinary sensor networks with best effort service. QoS constraints, such as packet loss, delay jitter and bandwidth constraints must be satisfied for multimedia services.

Recently, the quantum evolutionary algorithm [148] has been proposed as a novel algorithm for solving global multidimensional optimization problems. QEA combines an evolutionary algorithm and quantum computing. It employs the principles of quantum algorithms and mechanisms, such as Q-bits, superposition of quantum states, interference and uncertainty. Compared with the other heuristic optimization algorithms, QEA

shows advantages in computational complexity. However, the simple QEA has only one population which leads to a low convergence speed.

In this chapter, we propose a novel parallel elite clonal quantum evolutionary algorithm for QoS routing in multimedia WSNs. We first formulate our objective function to minimize path energy consumption under multiple constraits. We then design PECQEA with multiple populations to solve the multi-constrained QoS routing problem. PECQEA combines the merits of both the clonal selection algorithm and quantum evolutionary algorithm. It has fast search speed and global search ability in complex search spaces which is suitable for combinatorial optimization problems. Moreover, it can avoid local optima and converge rapidly towards the optimal region during the search process. Besides that, PECQEA is adaptive and robust when searching in a large space, as it can maintain individual diversity in different populations

The Markov chain theory is then used to analyze the convergence of PECQEA. First, we prove that the evolution process of PECQEA is a Markov chain. Then we model and analyze the transition matrix of the evolution process of PECQEA. The proof shows that PECQEA can converge to the global optimum within limited iterations.

To demonstrate the effectiveness of PECQEA, simulations are conducted for the multi-constrainted QoS unicast routing problem and performance comparisons are made with GA and ACO. Simulation results show that PECQEA achieves faster convergence and lower energy consumption than existing algorithms.

The rest of the chapter is organized as follows. The system model is described in Section 7.2. In Section 7.3, we present the design of PECQEA for QoS routing in detail. We analyze the convergence of PECQEA in Appendix A. Simulation results and performance comparisons of PECQEA, GA and ACO are discussed in Section 7.4. Finally, Section 7.5 concludes the chapter.

7.2 System Model

This section describes the mathematical model of the multi-constrainted QoS unicast routing problem for multimedia WSNs with respect to restrictions on communication capabilities. We use a directed graph $G(V, E)$, in which V is a set of nodes and E is a collection of links. Nodes include a source node, a destination node and a number of relay nodes. We denote the source node as No.1, the relay nodes as ascending numbers starting from No.2 to No.$(n-1)$ and the destination nodes serial numbers as No.n. So a directed route that starts from source node No.1 to the destination node No.n can be formulated as a series of ordered nodes $p(v_1, v_n) = \{v_1 \to \cdots \to v_i \to \cdots \to v_n\}$.

Each directed link e can be expressed by a pair of nodes $e = \{v_i \to v_j\}(i \neq j)$, and each link $e \in E$ has five QoS parameters: cost, delay, bandwidth, packet loss and delay jitter. The cost parameter of the link e can be formulated as a real number $cost(e) : E \to R_+$. Similarly, the delay, bandwidth, delay jitter and packet loss can be denoted by real numbers $delay(e)$, $bandwidth(e)$, $delay\_jitter(e)$ and $packet\_loss(e)$ respectively.

7.2.1 Radio Energy Model

In this chapter, we use the radio energy model proposed in [128]. On each link e, the cost energy consists of two parts, the transmission radio energy and the receive energy

$$cost(e) = cost_s + cost_r \tag{7.1}$$

In order to transmit k bits to another node in distance d, the transmission radio energy can be shown as:

$$cost_s(k, d) = \begin{cases} E_{elec} \cdot k + \varepsilon_{amp1} \cdot k \cdot d^2 & d \leq d_0 \\ E_{elec} \cdot k + \varepsilon_{amp2} \cdot k \cdot d^4 & d > d_0 \end{cases} \tag{7.2}$$

where $cost_s$ is the transmitter dissipated energy, d_0 is the transmission distance threshold and E_{elec} is the electronics energy parameter. The amplifier energy depends on the power

amplification parameter in free space ε_{amp1} and the power amplification parameter for multipath fading ε_{amp2}, the length of bits k and the distance between two nodes d. The receive energy can be shown as:

$$cost_r(k) = E_{elec} \cdot k \tag{7.3}$$

where $cost_r$ is the receiver dissipated energy. Since the energy consumed by computing and other activity is relatively fixed, in this work only the transmitter dissipated energy cost and the receiver dissipated energy cost are considered.

7.2.2 Route Functions

Route Cost Function

The energy cost of route $p(v_1, v_n)$ can be expressed as:

$$route\_cost(p(v_1, v_n)) = \sum_{e \in p(v_1, v_n)} cost(e) \tag{7.4}$$

where $route\_cost(p(v_1, v_n))$ is the energy cost of route $p(v_1, v_n)$, $e \in E$ is a link on route $p(v_1, v_n)$ and $cost(e)$ is the energy cost of the link e.

Route Delay Function

The total delay of the route can be formulated as:

$$route\_delay(p(v_1, v_n)) = \sum_{e \in p(v_1, v_n)} delay(e) \tag{7.5}$$

where $route\_delay(p(v_1, v_n))$ is the total delay time of route $p(v_1, v_n)$, $e \in E$ is a link on route $p(v_1, v_n)$ and the time delay of the link e is expressed as $delay(e)$.

Route Bandwidth Function

The bandwidth of route $p(v_1, v_n)$ can be expressed as:

$$route\_bandwidth(p(v_1, v_n)) = \min\{bandwidth(e)\} \tag{7.6}$$

where $bandwidth(p(v_1, v_n))$ is the bottleneck bandwidth of route $p(v_1, v_n)$, $e \in E$ is a link on route $p(v_1, v_n)$ and the bandwidth on link e can be represent as $bandwidth(e)$.

Delay Jitter Function

The delay jitter of route $p(v_1, v_n)$ can be expressed as:

$$route\_delay\_jitter(p(v_1, v_n)) = \sum_{e \in p(v_1, v_n)} delay\_jitter(e) \tag{7.7}$$

where $route\_delay\_jitter(p(v_1, v_n))$ is the total delay jitter of route $p(v_1, v_n)$, $e \in E$ is a link on route $p(v_1, v_n)$ and the delay jitter of link e can be represent as $delay\_jitter(e)$.

Packet Loss Function

The route also has a parameter representing the packet loss rate of $p(v_1, v_n)$ which can be expressed as:

$$\begin{aligned} &route\_packet\_loss(p(v_1, v_n)) = \\ &1 - \prod_{e \in p(v_1, v_n)} (1 - packet\_loss(e)) \end{aligned} \tag{7.8}$$

where $route\_packet\_loss(p(v_1, v_n))$ is the total packet loss of route $p(v_1, v_n)$, $e \in E$ is a link on route $p(v_1, v_n)$ and $packet\_loss(e)$ is the packet loss rate of link e.

7.2.3 Objective Function

Therefore, the multi-constrainted QoS unicast routing problem for multimedia services in WSNs can be represented as follows: Let $G(V, E)$ be a directed graph, each link $e \in E$ has five QoS parameters: cost $cost(e)$, delay $delay(e)$, bandwidth $bandwidth(e)$, delay jitter $delay\_jitter(e)$ and packet loss rate $packet\_loss(e)$. The objective is to search for a low energy cost route between the source and the destination, which can be expressed by:

$$E = \min route\_cost(p(v_1, v_n)) \tag{7.9}$$

7.2.4 Constraints

The aim of our algorithm is to search for a low energy cost route that starts from source v_1 to destination v_n, but as the multimedia service imposes QoS requirements, the route must also satisfy the constraints that can be formulated as:

$$route\_delay(p(v_1, v_n)) \leq D_{\max} \tag{7.10}$$

$$route\_bandwidth(p(v_1, v_n)) \geq B_{\min} \tag{7.11}$$

$$route\_delay\_jitter(p(v_1, v_n)) \leq J_{\max} \tag{7.12}$$

$$route\_packet\_loss(p(v_1, v_n)) \leq L_{\max} \tag{7.13}$$

where $D_{\max}$ is the acceptable maximum delay for multimedia service, $B_{\min}$ is the acceptable minimum bandwidth, $J_{\max}$ is the acceptable maximum delay jitter and $L_{\max}$ is the acceptable maximum packet loss rate.

7.3 QoS Routing Based On PECQEA

We present the design and computational procedure of PECQEA for multi-constrained QoS routing.

7.3.1 Brief Description of QEA

Based on the principles of quantum computation, Han and Kim first proposed QEA in [148] in order to solve a group of NP-hard optimization problems. QEA has a strong search capability which can effectively reduce the computational complexity and escape local optima. Analogous to the genetic and other evolutionary algorithms, QEA is a heuristic algorithm which also includes the concepts of population, individual and fitness function.

The theory of quantum computation was introduced in [149]. Quantum computing employs a Q-bit as the basic information unit which is similar to a bit in a classical computer. A Q-bit can take the value 0, expressed as the state vector $|0\rangle$, take the value 1, expressed as the state vector $|1\rangle$ or it can be in a superposition of the two. Quantum computing has better individual diversity than other single state representations, because it has a probabilistic representation of all the possible solutions to the problem.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{7.14}$$

In 7.14 α and β represent the probability of the two states. $|\alpha|^2$ and $|\beta|^2$ represent the probabilities that each bit takes a value of the two states after an observation. $|\alpha|^2$ and $|\beta|^2$ must satisfy $|\alpha|^2 + |\beta|^2 = 1$.

As the simple QEA only has one population which leads to a low convergence speed, we propose a novel PECQEA with multiple populations.

7.3.2 Encoding and Population Design

Each route $p(v_1, v_n) = \{v_1 \to \cdots \to v_i \to \cdots \to v_n\}$ can be represented as a series of path selections. We use out-degree to represent the number of edges directed out of a node. A quantum individual in PECQEA consists of n genes and the length of each gene equals to the rounded up binary logarithm of the maximum out-degree of all the nodes, which can be represented as $\lceil \log_2(MAX(outdegree)) \rceil$. The effective length of the binary code on link $e = \{v_i \to v_j\}(i \neq j)$ depends on the out-degree of the current node v_i. If the out-degree of node v_i is 1, we skip the gene, because there is only one path available. If the out-degree of node v_i is 2, we use the first binary bit of the gene to represent the path selection, one path as 0, the other path as 1. Similarly, if the out-degree of the node v_i is 4, we use two bits to represent the direction of the path; the four paths are encoded as 00, 01, 10 and 11 respectively. We convert binary bits on each gene into a natural number

and make a path selection according to this number. If any natural number exceeds the maximum value of the current node out-degree, the algorithm will take the remainder operator. If the path selection is finished before all the genes on the individual are used, the remaining genes will be ignored. In this way, each individual can be converted to a single route $p(v_1, v_n) = \{v_1 \rightarrow \cdots \rightarrow v_i \rightarrow \cdots \rightarrow v_n\}$. In PECQEA, a group of $O = n \times \lceil \log_2(MAX(outdegree)) \rceil$ quantum bits consisting of a quantum individual at a generation can be defined as:

$$q_m = \begin{bmatrix} \alpha_1 & \cdots & \alpha_i & \cdots & \alpha_O \\ \beta_1 & \cdots & \beta_i & \cdots & \beta_O \end{bmatrix}_{2\times O} \tag{7.15}$$

where n is the node number of the network, m is the counter of generation, and $|\alpha_i|^2 + |\beta_i|^2 = 1$.

As the quantum individual represents a probability of all the binary solutions at the same time, it has a better population diversity than the genetic algorithm during the evolutionary process.

For example, a Q-bit individual with a 3-Q-bit can be represented as:

$$q = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{2} & -\frac{1}{3} \\ -\frac{\sqrt{2}}{2} & \frac{1}{2} & \frac{2\sqrt{2}}{3} \end{bmatrix} \tag{7.16}$$

In another form:

$$\begin{aligned} q = &-\tfrac{\sqrt{6}}{12}|000\rangle + \tfrac{\sqrt{3}}{3}|001\rangle - \tfrac{\sqrt{2}}{12}|010\rangle + \tfrac{1}{3}|011\rangle \\ &+\tfrac{\sqrt{6}}{12}|100\rangle - \tfrac{\sqrt{3}}{3}|101\rangle + \tfrac{\sqrt{2}}{12}|110\rangle - \tfrac{1}{3}|111\rangle \end{aligned} \tag{7.17}$$

The above result means that the quantum individual contains the probabilities of all the states, which are 1/24, 1/3, 1/72, 1/9, 1/24, 1/3, 1/72, 1/9 for the eight different states from $|000\rangle$ to $|111\rangle$ respectively.

Unlike traditional QEA, individuals at generation m in PECQEA consist of multiple sub-populations, the i_{th} sub-population can be represented as:

$$Q_i(m) = \{\ q_m^1,\ \ q_m^2, \cdots, q_m^j\ \ , \cdots,\ \ q_m^L\ \} \tag{7.18}$$

where L is the size of each sub-population. Assume there are K sub-populations in the whole population Q, the whole population at generation m can be represented as $Q(m) = \left\{ \begin{array}{cccc} Q_1(m), & Q_2(m) & \cdots & Q_K(m) \end{array} \right\}$.

7.3.3 Fitness Function

The objective of PECQEA for the multi-constrainted QoS unicast routing problem for multimedia service delivery is to search for an optimal route from the source to the destination with the minimum cost. As delay, bandwidth, packet loss and delay jitter constraints are considered, the fitness function with the constraints can be formulated as:

$$fit(p(v_1, v_n)) = \sum_{e \in p(v_1, v_n)} cost(e) \tag{7.19}$$

where $fit()$ is the fitness function, $p(v_1, v_n)$ is the route encoded into the binary code. If (7.10), (7.11), (7.12) and (7.13) are satisfied, the cost is the total cost of the route. Otherwise, if any constraint is not satisfied, we set the fitness to a large constant as the penalty. The constant is much larger than the total energy cost of all links.

7.3.4 Clonal and Mutation Operators

In order to enhance the route search efficiency, a clonal operator is applied in each sub-population of PECQEA. The binary individual with the lowest energy consumption while satisfying all the constraints in each sub-population is chosen for cloning. First a fixed number of copies of the binary individual are generated, then the chaotic mutation operator is applied to each copy, which can effectively add diversity to the population. After that, we record the binary solution found in sub-population k as S_b^k, and use it in the individual update operation. With the clonal and the mutation operators, the algorithm can effectively avoid a local optimum.

7.3.5 Elite Operator

PECQEA keeps an elite binary solution S_e in the whole population. In each iteration, we use S_{best} to represent the solution with the lowest energy consumption. The elite binary solution will be replaced by S_{best} if S_{best} has a lower energy consumption. If S_{best} has higher energy consumption, the elite binary solution S_e will remain as in the previous generation.

7.3.6 Computational Procedure of PECQEA for Unicast Routing

The computational procedure of PECQEA is described as follows.

1) Parameters setting

Set the counter of the generation $t = 0$, and set the number of populations and quantum individuals in each population. The structure of a quantum individual is shown in (7.15).

2) Initialization

Initialize all the Q-bits of each individual to $1/\sqrt{2}$. This means that all the possible binary solutions have the same probability. In other words, when the number of the generation is $m = 0$, the j_{th} individual can be represented as

$$q_0^j = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \end{bmatrix}_{2\times O} \tag{7.20}$$

where the column of the individual q_0 is $O = \lceil \log_2(MAX(outdegree)) \rceil \times n$.

3) Observing

Start the while loop, observing the states of all the Q-bits in each individual. Each Q-bit collapses to a single state $|1\rangle$ or state $|0\rangle$ with the probability $|\beta|^2$ and $|\alpha|^2$, respectively. After doing so, each individual in the population collapses to a single binary vector solution S. The S can be defined as:

$$S_e = [\ s_1 \quad \cdots \quad s_i \quad \cdots \quad s_O\ ] \tag{7.21}$$

where $s_i \in \{0, 1\}$,$O = n \times \lceil \log_2(MAX(outdegree)) \rceil$.

4) Conversion

Convert the binary code to route $p(v_1, v_n)$ which is formulated as a series of ordered nodes.

5) Fitness Calculation

Calculate and evaluate the fitness value of each route with the fitness function (7.19).

6) Clonal and Mutation Operations

Clone the binary solution with lowest energy cost in the population and perform the mutation operator. Find the solution after mutation with the lowest energy cost and satisfy all constraints, and renew S_{best} with the solution.

7) Individual Update

Update each individual in the population by applying a quantum rotation gate. The quantum gate is a basic quantum operation in quantum computation which is similar to logic gates in conventional digital circuits. The update operation is given as:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos(\Delta\omega) & -\sin(\Delta\omega) \\ \sin(\Delta\omega) & \cos(\Delta\omega) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{7.22}$$

Table 7.1: The sign of quantum rotation angle

$s_{m,n}^{l,k}$	$s_{m,n}^{b,k}$	$f(S) > f(S_b^k)$	$r(\alpha\beta)$			
			$\alpha\beta > 0$	$\alpha\beta < 0$	$\alpha = 0$	$\beta = 0$
0	0	False	0	0	0	0
0	0	True	0	0	0	0
0	1	False	+1	-1	0	+1
0	1	True	-1	+1	+1	0
1	0	False	-1	+1	+1	0
1	0	True	+1	-1	0	+1
1	1	False	0	0	0	0
1	1	True	0	0	0	0

where $i = 1 \ldots M \times N$, and $\Delta\omega$ is the rotation angle. Rotation angle $\Delta\omega$ depends on the corresponding Q-bit value of the binary vector solution S, the binary vector solution with lowest energy consumption S_b, α and β. The rotation angle $\Delta\omega$ is updated as:

$$\Delta\omega = 0.05\pi \times r(\alpha\beta) \tag{7.23}$$

where $r(\alpha\beta)$ is the sign of the rotation angle. We use a similar lookup table as in [117] shown in Table 7.1, where $s_{m,n}^{l,k}$ is the n_{th} binary bit of the l_{th} quantum individual of the k_{th} sub-population in the m_{th} generation generated by the observing operation and $s_{m,n}^{b,k}$ is the n_{th} binary bit of S_b^k in the m_{th} generation generated by the clonal and mutation operations.

8) Q-bits Mutation

In order to prevent premature convergence, we design a Q-bits mutation operator which randomly chooses some Q-bits with a very small fixed rate P_{mu} and replaces α on on these bits with some random decimal between 0 and 1. The value of β on corresponding Q-bits can be calculated by the formula $\beta = \sqrt{1-\alpha^2}$.

9) H_ε Gate

When $|\alpha|^2$=1, $|\beta|^2$= 0 or $|\alpha|^2$=0, $|\beta|^2$= 1, the ordinary QEA is easy to fall into local optima which would cause evolutionary stagnation. We adopt the H_ε gate which was first proposed in [151] to prevent evolutionary stagnation. We use a small positive number $\varepsilon = 0.01$ to adjust the quantum bits in all the quantum individuals. If $|\alpha|^2 < \varepsilon$, in other words $|\beta|^2 > 1-\varepsilon$, we set $\alpha = \sqrt{\varepsilon}$ and $\beta = \sqrt{1-\varepsilon}$. If $|\beta|^2 < \varepsilon$, in other words $|\alpha|^2 > 1-\varepsilon$, we set $\beta = \sqrt{\varepsilon}$ and $\alpha = \sqrt{1-\varepsilon}$. α and β do not change in other cases.

10) Elite List Update

Record the solution with the lowest energy consumption S_{best} in the current generation. Compare the fitness value of S_{best} found in the current generation and the elite solution S_e in the elite list. The elite binary solution S_e will be replaced by S_{best} if S_{best} has a lower fitness value. If S_e has a lower fitness value than S_{best}, the elite binary solution S_e will be kept in the elite list.

11) Termination Condition

Check whether the algorithm has reached the maximum number of generations. If not, set $m = m + 1$ and go to Step 3); otherwise, convert the elite binary solution S_e to route $p(v_1, v_n)_e = \{v_1 \to \cdots \to v_i \to \cdots \to v_n\}$, and then output the route and its energy cost.

The convergence analysis of the algorithm is presented in Appendix A.

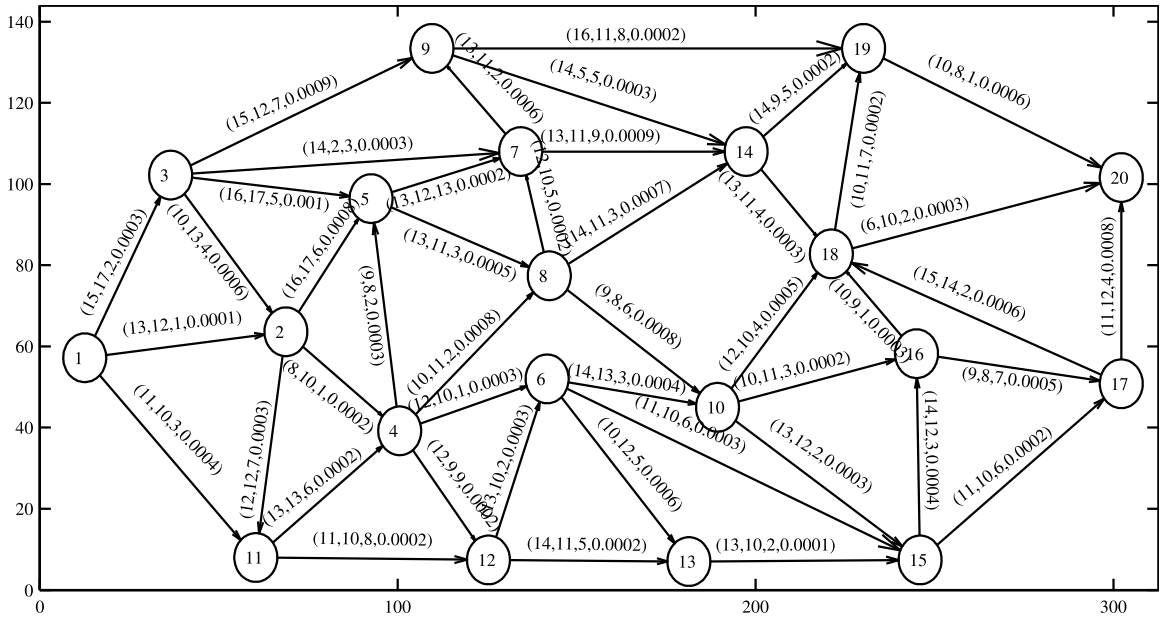


Figure 7.1: Network topology of a wireless sensor network with 20 nodes. Distances are in meters, the parameters on each link are the delay, bandwidth, delay jitter and packet loss.

7.4 Simulation Results and Disscussion

We simulate the proposed PECQEA with 20 nodes for multimedia service delivery in a wireless sensor network. For comparison purposes, in this work, we consider a similar network topology as the directed graph in [150]. The topology of the network is shown in Fig.7.1 and the parameters on each link are the delay, bandwidth, delay jitter and packet loss, respectively. The energy consumption is calculated by (7.1), (7.2) and (7.3) with the distance in meters shown as in Fig.7.1. Similar to [128], we set $E_{elec} = 50\text{nJ/bit}$, $k = 1\text{Mbits}$, $\varepsilon_{amp1} = 10\text{pJ/bit/m}^2$, $\varepsilon_{amp2} = 0.0015\text{pJ/bit/m}^4$, and $d_0 = 1\text{m}$. To facilitate the representation, delay, bandwidth and delay jitter are normalized to positive integers. According to the parameters requirement of some practical multimedia wireless sensor networks, we set the maximum delay to 105ms, minimum bandwidth to 8Mbps, maximum

delay jitter to 36ms and maximum packet loss rate to 0.01.

Simulation experiments were conducted to demonstrate PECQEA capabilities. In the simulation experiments, the algorithm performance is evaluated by the fitness function proposed in (7.19). Comparisons are also made with GA and ACO in the same simulation environment. All numerical results are the average of 10000 simulation runs. In all three algorithms, the maximum number of iterations is 50 and we only compare the best solutions among the populations. In the proposed PECQEA, there are four populations and the number of individuals in each population is 3. In GA, the population consists of 12 chromosomes and the crossover and mutation probabilities are 0.9 and 0.1, respectively. In ACO, there are 12 ants in the population, the evaporation coefficient is set to 0.1 and the cognitive and social parameters are set to 2 and 2 respectively.

Fig. 7.2 shows that the proposed PECQEA has a lower cost compared to GA and ACO after 50 iterations. After 50 iterations, the average energy cost found by PECQEA is 0.6629J. In contrast, GA cost is 0.6863J, and ACO cost is 0.7201J. These results show that PECQEA achieved more than 3.4% and 7.9% energy cost reductions over GA and ACO, respectively. Moreover, PECQEA also enjoys a faster convergence rate than GA and ACO. The energy cost of PECQEA is reduced to below 0.7 after only 9 generations, while GA takes 34 generations to reach that cost level and ACO takes more than 50 generations.

Fig. 7.3 shows that the proposed PECQEA has a slightly higher average delay time compared to GA and ACO after 50 iterations. After 50 iterations, the average delay time of PECQEA is 73.02ms. In contrast, the average delay time of GA is 71.83ms and the average delay time of ACO is 69.01ms. All three algorithms have a delay time lower than maximum delay 105ms.

Fig. 7.4 shows that the proposed PECQEA has a higher bandwidth compared to GA and ACO during 50 iterations. After 50 iterations, the average bandwidth found by

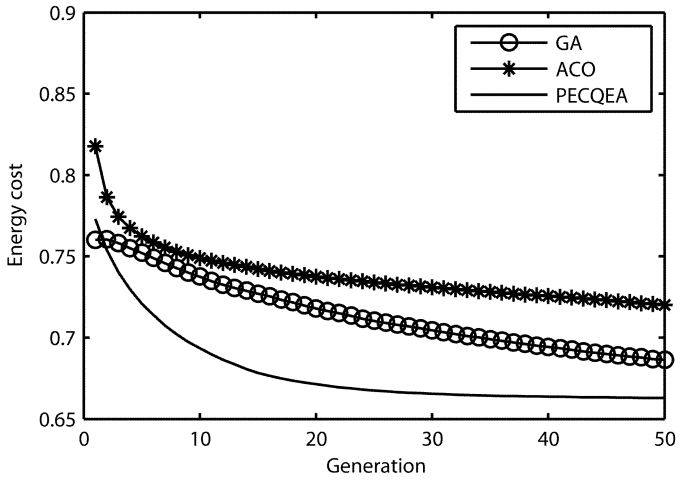


Figure 7.2: Average energy cost for GA, ACO and PECQEA at each generation.

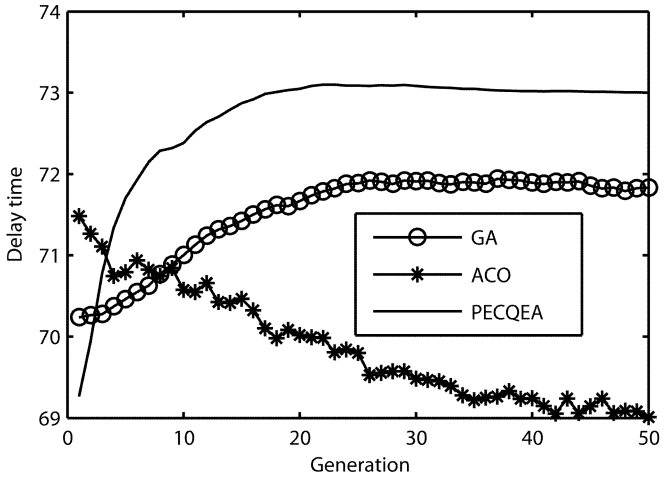


Figure 7.3: Average route delay time for GA, ACO and PECQEA at each generation.

PECQEA is 9.160Mbps. In contrast, GA bandwidth is 8.853Mbps, and ACO bandwidth is 8.937Mbps. These results show that PECQEA enjoys a higher bandwidth than GA and ACO. All three algorithms have bandwidth values higher than minimum bandwidth 8Mbps.

Fig. 7.5 shows that the proposed PECQEA has a lower delay jitter compared to GA and ACO after 50 iterations. After 50 iterations, the average delay jitter found by

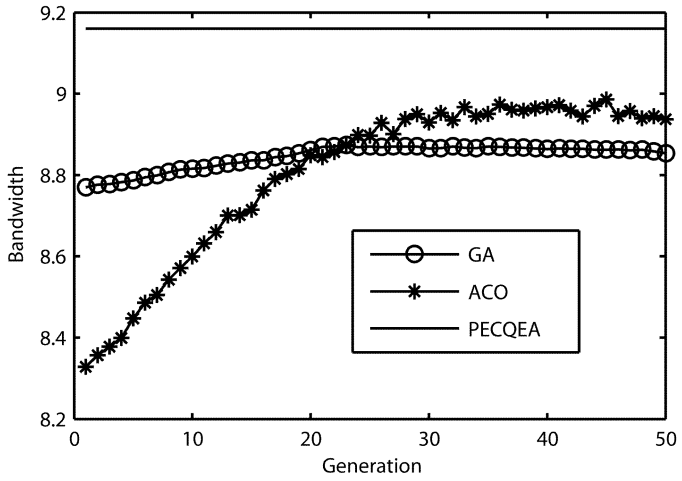


Figure 7.4: Average route bandwidth for GA, ACO and PECQEA at each generation.

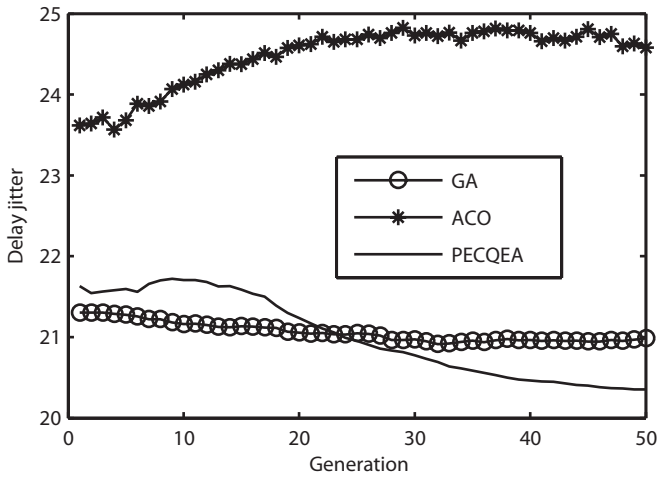


Figure 7.5: Average route delay jitter for GA, ACO and PECQEA at each generation.

PECQEA is 20.35ms. In contrast, the value for GA is 20.98ms and the value for ACO is 24.58ms. These results show that PECQEA enjoys more stable packet arrivals.

Fig. 7.6 shows that the proposed PECQEA has a slightly higher packet loss rate compared to GA and ACO after 50 iterations. At 50 iterations, the packet loss rate of PECQEA is 0.003049. In contrast, GA packet loss rate is 0.003036, and ACO packet loss rate is 0.002391. All three algorithms have packet loss rates lower than the maximum

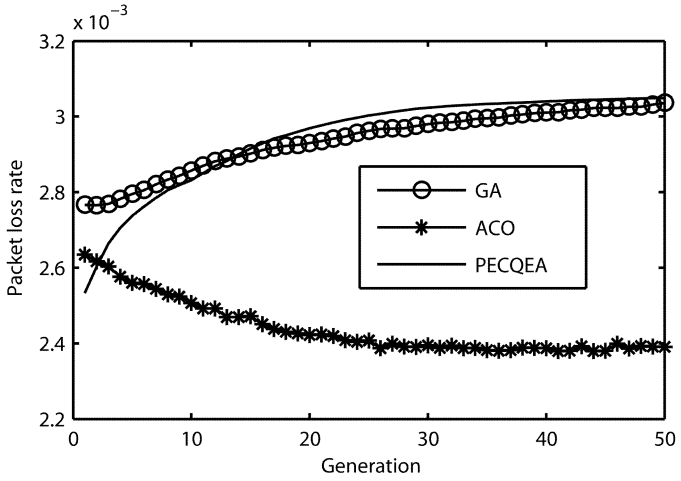


Figure 7.6: Average route packet loss rate for GA, ACO and PECQEA at each generation.

packet loss rate 0.01.

7.5 Summary

We proposed a novel parallel elite clonal quantum evolutionary algorithm (PECQEA) to solve the multi-constrainted QoS unicast routing problem for multimedia service delivery in WSNs. PECQEA aims to reduce energy cost while meeting the delay, bandwidth, delay jitter and packet loss constraints. By using the Markov chain theory, the global convergence property of PECQEA is proved. Simulation experiments were conducted to demonstrate the capabilities of the proposed PECQEA and performance comparisons are made with GA and ACO. The results show that PECQEA can effectively reduce the energy cost under multiple constraints.

Chapter 8

Conclusions and Future Work

With the fast growth of wireless communications technologies, an increasing number of scientists focus on several bottleneck issues in the optimization technique of wireless communications, especially NP-hard problems. Our studies have been inspired from the need to solve hard discrete optimization problems in wireless communications systems. We investigated discrete optimization problems in five different cases, built system models and utility functions and designed novel evolutionary algorithms for the relevant problems.

8.1 Innovations in the Thesis

In this thesis, we studied five key technologies in wireless communication based on evolutionary algorithms. The main innovations and contributions of this thesis are as follows: We propose a new modified chaos clonal shuffled frog leaping algorithm (MCCSFLA) for PAPR reduction in orthogonal frequency division multiplexing (OFDM) systems. We also present a novel approach based on quantum ant colony evolutionary algorithm (QACEA) for target coverage problem in large-scale self-organizing wireless sensor networks. Then we propose a novel quantum immune clonal evolutionary algorithm (QICEA) for the duty cycle sequence design with full coverage constraint and present a novel fuzzy simulated

evolutionary computation clustering (FSEC) means for the clustering in large-scale wireless sensor networks. Finally, we propose a new parallel elite clonal quantum evolutionary algorithm (PECQEA) to solve the multi-constraints QoS routing problem.

8.2 Future Work

The following areas can be considered for future works based on the work conducted in this thesis.

(1) WSN coverage based on evolutionary algorithms

This thesis studied the point coverage issue based on evolutionary algorithms. Other issues such as area coverage and barrier coverage are directions for future work. A further exciting study direction is how to deal with the heterogeneous node coverage issue where several types of sensors with various capabilities could be implemented for the coverage problem. Improving WSNs coverage will ultimately produce better functioning of the system. We intend to make use of evolutionary algorithms strategies in such optimizations.

(2)WSN duty cycle design based on evolutionary algorithms

In our proposed system model, we simply let all the nodes have same remaining energy. Hence, more accurate models need to be built to support WSNs. Future additions of this study involve considering mobile nodes and heterogeneous nodes, as well as other more practical system models.

(3) WSN routing based on evolutionary algorithms

In this thesis, we have studied unicast routing in WSNs. Much of the work in this thesis is designed with stationary nodes. A future direction is to extend our work to include mobile nodes. It will be also beneficial to research if our outcomes make generalizations in multicast routing. We also believe that more problems could be considered if cluster movement is made possible in WSNs.

(4) Clustering based on evolutionary algorithms

Our study concentrated only on single-hop clustering in WSNs. It is essential to expand this work to multihop clustering for WSNs. Some assumptions made in this study, such as unlimited cluster size, can be relaxed by building more realistic models. By doing this kind of research, we hope to have the ability to offer detailed methods to improve the entire efficiency of WSNs.

(5) Cognitive radio optimization based on evolutionary algorithms

Cognitive radio spectrum allocation is an excellent candidate for optimization employing evolutionary algorithms strategy. An optimization approach can often assign spectrum from the preset collection to every user based on requirements. This might include formation of new fitness functions to evaluate this kind of allocation. Another interesting direction is on the mobility of users while dealing with the channel allocation problem. This presents new challenges with regards to users' distribution and density.

Appendix A

A.1 Convergence Analysis

Convergence is an important property of PECQEA for the multi-constrained QoS unicast routing problem for multimedia WSNs. We discuss this property with the Markov chain theory based on the work in [110]. Similar work for genetic algorithms can be found in [115], [113], [114] and [116]. We first show that the evolution process of PECQEA constitute an ergodic Markov chain. The Markov chain is then extended and its properties are investigated. Finally, PECQEA is proved to converge to the global optimum.

A.1.1 Basic Definitions and Theory in Markov Chains

A discrete-time Markov chain is a sequence of random values that transitions from one state to another with memoryless character.

Definition 1 *[99]. Let random process* $\{Z(m)\}_{j=1}^{\infty}$ *in discrete state space* Φ*, if*

$$\begin{aligned} &P\{Z(m+1)\} = j_{m+1} | Z(0) = j_0, \cdots, Z(m) = j_m\} \\ &= P\{Z(m+1) = j_{m+1} | Z(m) = j_m\} \end{aligned} \tag{A.1}$$

, then $\{Z(m)\}_{j=1}^{\infty}$ *is called Markov chain.*

Next, we will give the definition of time homogeneous Markov chains.

Definition 2 *[112]. If the transition probability $h_{ij}(m)$ for Markov chain $\{Z(m)\}_{j=1}^{\infty}$ is independent from m, which means $H_{ij}(m, m+1) = H\{Z(m+1) = j | Z(m) = i\} = H_{ij}(i, j \in \Phi)$, then $\{Z(m)\}_{j=1}^{\infty}$ is said to be homogeneous.*

We use a 2-D matrix H to represent the transition probability between states, which can be represented as:

$$H = [h_{i,j}] = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,j} & \cdots \\ h_{1,0} & h_{1,1} & \cdots & h_{1,j} & \cdots \\ \vdots & \ldots & \ddots & \cdots & \cdots \\ h_{i,0} & h_{i,1} & \cdots & h_{i,j} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{A.2}$$

where $h_{ij}(m)$ is the transition probability from state i to state j. Matrix H generally satisfies the constraints $h_{ij} \geq 0$ and $\sum h_{ij} = 1$ for any $i, j \in \Phi$. In this way, a homogeneous Markov chain $\{Z(m)\}_{j=1}^{\infty}$ in any time m can be characterized by the couple (h^0, H), where h_0 is the initial distribution of the Markov chain.

Definition 3 *[112]. A Markov chain is called Ergodic if it can reach to any state from any state (unnecessary in one move).*

Theorem 1 *[99]. Let H be the transition probability matrix of an ergodic Markov chain with size $k \times k$, the powers H^m converge to a $k \times k$ matrix H^{∞} with the same rows h^{∞}. If $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1 \times k}$, then $H^{\infty} = g' h^{\infty}$.*

From Theorem 1 we could know that the limit probability distribution of an ergodic Markov chain has no relationship with the initial distribution h^0.

A.1.2 The base Markov chain for PECQEA

In the searching process of PECQEA, each quantum individual can be represented as a vector of Q-bits and each Q-bit can be represented as a decimal $|\alpha_i|^2$ between 0 and 1.In this way, the value of $|\beta_i|^2$ equal to $1-|\alpha_i|^2$. As in computer simulations the decimal place of each Q-bit is represented as a finite binary decimal between 0 and 1, each pair of Q-bits can be represented as a finite binary number with a limited number of bits. The length of the binary number depends on the accuracy of the computer simulation. In this way, each Q-bit can be represented as a binary sequence $R=\{r_1, r_2, \cdots, r_W\}, r_i \in \{0,1\}$, where W is the length of the binary sequence. Similarly, each quantum individual q can be represented as $q = \{\ R_1,\ \cdots,\ R_O\ \} = \{\underbrace{r_1 \cdots r_W}_{R_1} \cdots \underbrace{r_{W(O-1)+1} \cdots r_{WO}}_{R_O}\}$. As there are K sub-populations in the whole quantum population Q and L quantum individuals in each sub-population, the whole quantum population Q can be represented as $Q = \{\ Q_1,\ \cdots,\ Q_K\ \} = \{\underbrace{r_1 \cdots r_W}_{R_1} \cdots \underbrace{r_{W(KLO-1)+1} \cdots r_{KLWO}}_{R_{KLO}}\}$. In this formula, K is the number of the sub-populations in the whole population, L is the size of each sub-population and O is the number of Q-bits in each quantum individual.

Lemma 1 *In PECQEA, the evolutionary process of the whole quantum population Q constitutes a Markov chain.*

Proof: The iteration of the whole quantum population Q can be viewed as a stochastic process.

According to the evolutionary process of PECQEA, we may know that the quantum population $Q(m)$ in the current generation is obtained from the quantum population in the former generation $Q(m-1)$ and it is nothing to do with $Q(m-2)$, $Q(m-3)$, $\cdots$, $Q(0)$. Moreover, all the elements in the rotation Table 7.1 are generated in the current or the generation. So we have $H\{Q(m)=i_m|Q(0)=i_0, Q(1)=i_1, \cdots, Q(m-1)=i_{m-1}\} = H\{Q(m)=i_m|Q(m-1)=i_{m-1}\}$, where H is the transition matrix in generation m.

Thus the iteration of the whole quantum population Q can be modelled with a Markov chain and its character could be analyzed by the relevant theory.

Theorem 2 *In PECQEA, the population sequence $Q(m)_{m=1}^{\infty}$ constitutes a finite Markov chain.*

Proof: In this work, the population sequence is defined as a binary string $Q(m)$ with length $KLWO$ over the set $C = \{0, 1\}$. So there are 2^{KLWO} possible strings in the whole search space, forming the population space $\Omega = IC^{KLWO}$. The number of the possible states in the population space is $|\Omega| = 2^{KLWO}$. So the state space of $Q(m)$ is finite, and the population sequence $Q(m)_{m=1}^{\infty}$ constitutes a finite Markov chain.

Theorem 3 *The Markov chain $\{Q(m)\}$ for PECQEA is time homogeneous.*

Proof: Let $H(m)$ denote the transition matrix in generation m. Because the transition probability of $H(m)$ in PECQEA is only relevant to the elements in the rotation Table 7.1 and all the elements in the rotation Table 7.1 are generated in the current or former generation and have nothing to do with the iteration number m. In this way, the evolution process of $\{Q(m)\}$ does not change with a generation. Thus, the evolution process of $\{Q(m)\}$ constitutes a homogeneous Markov chain.

In order to facilitate analysis, we divide the evolution process in the base Markov chain into two stages. The first stage includes the operations before the Q-bits mutation and the second stage is the Q-bits mutation. In other words, the first stage includes the steps from 3 to 7 in Section 7.3.6 and the second stage is the step 8 in Section 7.3.6. In order to facilitate the analysis, in the first stage we use the transition matrix R to represent the stochastic process. In other words, matrix R represents the quantum rotation operation for a quantum individual. In the second stage we use the transition matrix C to represent the Q-bits mutation operation for a quantum individual. In this way, the base Markov

chain can be represented as $H{=}RC$, where H is the transition matrix of the base Markov chain.

Lemma 2 *The transition matrix of the quantum rotation operation R in PECQEA is stochastic.*

Proof: The transition matrix of the quantum rotation operation R can be shown as:

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1,2^{KLWO}} \\ \vdots & \ddots & \vdots \\ r_{2^{KLWO},1} & \cdots & r_{2^{KLWO},2^{KLWO}} \end{pmatrix}. \tag{A.3}$$

According to Table 7.1, in the quantum rotation operation, each bit r of the Q-bits binary sequence $Q = \{r_1, r_2, \cdots, r_{KLWO}\}$ is generated from the set $\{0, 1\}$ with a certain probability. As the length of the Q-bits binary sequence is constant $KLWO$, the transition operator maps in the space Ω. The number of the states is $|\Omega| = 2^{KLWO}$. This operation produces no new state and no state disappears. So the total probability of transitions from any state to other states is 1, which can be represented as $\sum_{j=1}^{2^{KLWO}} r_{ij} = 1$ for any $i, j \in \{1, \ldots, 2^{KLWO}\}$. So matrix R is stochastic.

Lemma 3 *The transition matrix of Q-bits mutation C is both stochastic and positive.*

Proof: We use C to denote the $2^{KLWO} \times 2^{KLWO}$ mutation matrix, which can be shown as

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1,2^{KLWO}} \\ \vdots & \ddots & \vdots \\ c_{2^{KLWO},1} & \cdots & c_{2^{KLWO},2^{KLWO}} \end{pmatrix}. \tag{A.4}$$

Similar to the quantum rotation operation, in the Q-bits mutation operation, each bit r of the Q-bits binary sequence $Q = \{r_1, r_2, \cdots, r_{KLWO}\}$ is selected from the set $\{0, 1\}$.

As the length of the Q-bits binary sequence is constant $KLWO$, we have $\sum_{j=1}^{2^{KLWO}} c_{ij} = 1$ for any $i, j \in \{1, \ldots, 2^{KLWO}\}$. So the matrix C is stochastic.

The Q-bits mutation operator randomly chooses some Q-bits with a very small fixed rate P_{mu} and replaces these Q-bits with some random decimal numbers between 0 and 1. For each bit on the Q-bits binary sequence $Q = \{r_1, r_2, \cdots, r_{KLWO}\}$, the probability of keeping the original value is $1 - P_{mu}/2$, and the probability of changing to complement from 1 to 0 or from 0 to 1 is $P_{mu}/2$. From the probability theory point of view, the Q-bits mutation operation randomly maps from the current state to all the other states in the space with different nonzero probabilities. So we have $c_{ij} > 0$ for all $i, j \in \{1, \ldots, 2^{KLWO}\}$. Thus the transition matrix C is positive.

Lemma 4 *The transition matrix of the base Markov chain H is a positive stochastic matrix.*

Proof: According to the above discussion, the base Markov chain can be represented as H=RC. In this way, for any element h_{ik} in transition matrix H, we have $h_{ik} = \sum_{j=1}^{2^{KLWO}} r_{ij}c_{jk}$. For matrix R, we have $\sum_{j=1}^{2^{KLWO}} r_{ij} = 1$ for any $i, j \in \{1, \ldots, 2^{KLWO}\}$. Thus, for any row i in matrix R, we could assume that the maximum positive element in the row i is in column max. In other words, $r_{i,\max}$ is the largest element in row i. In this way, we have

$$h_{ik} = \sum_{j=1}^{2^{KLWO}} r_{ij}c_{jk} \geq r_{i,\max}c_{\max,k} > 0. \tag{A.5}$$

Thus the transition matrix of the base Markov chain H is a positive matrix.

As both R and C are right stochastic matrices, for any row j we have $\sum_{j=1}^{2^{KLWO}} r_{ij} = 1$ and $\sum_{j=1}^{2^{KLWO}} c_{ij} = 1$. Thus, for any row i in matrix C, we have

$$\begin{aligned}\sum_{k=1}^{2^{KLWO}} c_{ik} &= \sum_{k=1}^{2^{KLWO}} \sum_{j=1}^{2^{KLWO}} r_{ij}c_{jk} = \sum_{j=1}^{2^{KLWO}} r_{ij} \sum_{k=1}^{2^{KLWO}} c_{jk} \\ &= \sum_{j=1}^{2^{KLWO}} r_{ij} = 1\end{aligned} \tag{A.6}$$

Thus the transition matrix of the base Markov chain H is a stochastic matrix.

Lemma 5 *The transition matrix of the base Markov chain H is a primitive matrix.*

Proof: As H is a positive matrix, $H^1 > 0$, thus H is a primitive matrix.

Definition 4 *Definition 1: [99] A Markov chain is called an ergodic chain if it is possible to go from every state to every state (not necessarily in one move).*

Theorem 4 *In PECQEA, the evolutionary process of the whole quantum population Q constitutes an ergodic Markov chain.*

Proof: H is the transition matrix for the evolutionary process of the whole quantum population Q and it is a positive matrix. So we have $h_{ij} > 0$ for all i and j. So in space Ω, the possibility of going from every state to every state is positive in just one move. Thus the Markov chain is an ergodic chain.

According to Theorem 4, we can easily get the following result:

Theorem 5 *[99]. Let H be the transition probability matrix of the evolutionary process of the whole quantum population Q with size $2^{KLWO} \times 2^{KLWO}$, the powers $\lim_{m\to\infty} H^m$ converge to a $2^{KLWO} \times 2^{KLWO}$ matrix H^∞ with the same rows h^∞.*

If $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times 2^{KLWO}}$, *then* $H^\infty = g'x^\infty$.

A.1.3 The Extended Elite Markov Chain

In order to keep the best solution in each generation, the best binary solution generated in the current iteration will replace the solution in the elite list if it has a lower

fitness value. The elite binary solution reserved in the elite list can be represented as $S_e = [\ s_1\ \ \cdots\ \ s_i\ \ \cdots\ \ s_O\ ]$, where $s_i \in \{0,1\}$,$O = n \times \lceil \log_2(MAX(outdegree)) \rceil$. In this way, we can extend the base Markov chain to the elite Markov chain by adding the elite binary vector solution to the state space. Thus, the population space $\Omega = IC^{KLWO}$ extends to $\Omega^+ = IC^{(KLW+1)O}$ and the number of the possible states in the space grows to $|\Omega^+| = 2^{(KLW+1)O}$. So the extended whole quantum population Q^+ can be represented as $Q^+ = \{S_e,\ Q_1,\ \cdots,\ Q_K\ \} = \{\underbrace{r_1 \cdots r_O}_{S_e} \underbrace{r_{O+1} \cdots r_{O+W}}_{R_1} \cdots \underbrace{r_{(KLW+1)O-W+1} \cdots r_{(KLW+1)O}}_{R_{KLO}}\}$. In this way, for Q^+, the size of the transition matrix $F(m)$ is $2^{(KLW+1)O} \times 2^{(KLW+1)O}$.

Lemma 6 *In PECQEA, the evolutionary process of Q^+ constitutes a Markov chain.*

Proof: The iteration of of the whole quantum population Q^+ can be viewed as a stochastic process. According to the evolutionary process of PECQEA, we may know that both the elite binary sequence S_e and the quantum population $Q(m)$ in the current generation are obtained from the elite binary sequence and the quantum population in the former generation and it has nothing to do with any element in generation $m-2$, $m-3$,$\cdots$,1. So we have $F\{Q^+(m) = i_m | Q^+(0) = i_0, Q^+(1) = i_1, \cdots, Q^+(m-1) = i_{m-1}\} = F\{Q^+(m) = i_m | Q^+(m-1) = i_{m-1}\}$, where $F(m)$ is the transition matrix in generation m. Thus the evolutionary process of Q^+ can be modelled with a Markov chain and its character could be analyzed by the Markov chain theory.

Lemma 7 *In PECQEA, the extended population sequence $Q^+(m)_{m=1}^{\infty}$ constitutes a finite Markov chain.*

Proof: In Q^+, the length of the binary sequence is defined as a binary string $Q(m)$ with length $(KLW+1)O$ over the set $C = \{0,1\}$. So there are $2^{(KLW+1)O}$ possible strings in the whole search space, forming the population space Ω^+. The number of the possible states in Q^+ is $|\Omega| = 2^{(KLW+1)O}$. So the state space of Q^+ is finite and the population sequence $Q^+(m)_{m=1}^{\infty}$ constitutes a finite Markov chain.

We divide the iteration process of Q^+ into two stages. The first stage includes the operations from observing Q-bits mutation and the second stage is the elite list update. In other words, the first stage includes the steps from 3 to 8 in Section 7.3.6 and the second stage is the step 9 in Section 7.3.6. In order to facilitate the analysis, we use the transition matrix $A(m)$ to represent the transition process in the first stage and we use transition matrix $T(m)$ to represent the transition process in the second stage. In this way, the base Markov chain can be represented as $F(m) = A(m)T(m)$, where $F(m)$ is the transition matrix of the base Markov chain.

Lemma 8 *The Markov chain $Q^+(m)$ for PECQEA is time homogeneous.*

Proof: Transition probability of $F(m)$ in PECQEA is only relevant to the elements in the rotation Table 7.1 and all the elements in the rotation table 7.1 are generated in the current or former generation and have nothing to do with the iteration number m. In this way, the evolution process of $Q^+(m)$ does not change with a generation. Thus, the evolution process of $Q^+(m)$ constitutes a homogeneous Markov chain. So we can use F to represent $F(m)$.

Lemma 9 *The transition matrix of the transition matrix F for the extended population sequence $Q^+(m)$ in PECQEA is stochastic.*

Proof: The transition matrix F can be shown as:

$$F = \begin{pmatrix} f_{11} & \cdots & f_{1,2^{(KLW+1)O}} \\ \vdots & \ddots & \vdots \\ f_{2^{(KLW+1)O},1} & \cdots & f_{2^{(KLW+1)O},2^{(KLW+1)O}} \end{pmatrix}. \tag{A.7}$$

As all the Q-bits in the binary sequence $Q^+ = \{r_1 \cdots\cdots r_{(KLW+1)O}\}$ are generated from the set $\{0, 1\}$ with a certain probability, the length of the Q-bits binary sequence is constant $(KLW + 1)O$ and the transition operator maps in the space Ω^+. The number

of the states is $|\Omega^+| = 2^{(KLW+1)O}$. This operation produces no new state, and no state disappears. So the total probability of transitions from any state to other states is 1, which can be represented as $\sum_{j=1}^{2^{(KLW+1)O}} f_{ij} = 1$ for any $i, j \in \{1, \ldots, 2^{(KLW+1)O}\}$. So matrix F is stochastic.

We assume that there exists a unique globally optimal solution S_g with the lowest fitness value. In order to facilitate the representation, we arrange the solution space Ω^+ with a decreasing order according to the fitness value of S_e. In this way, states with better S_e are ranked higher in the solution space and the states with the worst S_e are ranked at the bottom in the solution space. For the states with the same S_e, we order the states according to the binary value. In other words, as there are $|\Omega^+| = 2^{(KLW+1)O}$ states in the whole space, the states that are numbered from 1 to 2^{KLWO} have the best S_e which is equal to S_g and the states that are numbered from $2^{KLWO} * (2^O - 1) + 1$ to $2^{KLW(O+1)}$ have the worst S_e. In the states with the same S_e, the state with an all-zero binary string following S_e is ranked in the top, and the state with an all-one binary string following S_e is ranked at the bottom. In other words, in the states with the same S_e, the state that with higher binary value will be ranked lower.

In the first stage the solution in the elite list is not affected by the quantum rotation operation, so the extended first stage transition matrix $A(m)$ can be represented as a block diagonal matrix

$$A = \begin{pmatrix} H & & & \\ & H & & \\ & & \ddots & \\ & & & H \end{pmatrix}. \tag{A.8}$$

where the size of each H is $2^{KLWO} \times 2^{KLWO}$ and the size of A is $2^{(KLW+1)O} \times 2^{(KLW+1)O}$. So the number of H in the diagonal of A is 2^O.

Lemma 10 *The transition matrix $T(m)$ in the second stage for the extended Markov chain $Q^+(m)$ is a lower triangular matrix.*

Proof: First, we divide the transition matrix $T(m)$ in to square matrix into sub-blocks and the size of each sub-block is $2^{KLWO} \times 2^{KLWO}$. In this way, the transition matrix $T(m)$ can be represented as

$$T = \begin{pmatrix} T_{11} & T_{12} & \cdots & T_{1,2^O} \\ T_{21} & T_{22} & \cdots & T_{2,2^O} \\ \vdots & \vdots & \ddots & \vdots \\ T_{2^O,1} & T_{2^O,2} & \cdots & T_{2^O,2^O} \end{pmatrix}. \tag{A.9}$$

To facilitate the representation, we define the current state as state i and the next state as state j. We can easily find that the states i in the sub-blocks with the same row number have the same elite solution S_e in the elite list. As the state space is a decreasing order according to the fitness value, the lower row number, the lower the fitness in state i. For example, the square matrix sub-blocks T_{11}, T_{12}, $\cdots$,$T_{1,2^O}$ have the same elite solution $S_e = S_g$ in state i and they have the lowest fitness. In contrast, the square matrix sub-blocks $T_{2^O,1}$, $T_{2^O,2}$, $\cdots$, $T_{2^O,2^O}$ have the same elite solution with the highest fitness value in state i, which means their energy consumption in state i is highest.

According to the elite list update procedure, we compare the fitness value of S_{best} found in the current generation and the elite solution S_e in the elite list. If S_{best} has a lower fitness value, the elite binary solution S_e will be replaced by S_{best}, which means that state j has a lower fitness value than state i and $j < i$. If S_e has a lower or equal fitness value than S_{best}, the elite binary solution S_e will be kept on the elite list. In this case, state j has the same fitness value as state i and $j = i$. In other words, the situation that $j > i$ can not occur in PECQEA, so $T(m)$ is a lower triangular matrix. In this way, $T(m)$ could be expressed as

$$T = \begin{pmatrix} T_{11} & & & \\ T_{21} & T_{22} & & \\ \vdots & \vdots & \ddots & \\ T_{2^O,1} & T_{2^O,2} & \cdots & T_{2^O,2^O} \end{pmatrix}. \tag{A.10}$$

Lemma 11 *Sub-block T_{11} in transition matrix $T(m)$ is a unit matrix.*

Proof: The square matrix sub-blocks T_{11} have the elite solution with the lowest fitness $S_e = S_g$ in state i. So, according to the elite list update procedure, S_g will not be replaced by any S_{best}. Thus we have $j = i$ and in this way sub-block T_{11} is a unit matrix.

Lemma 12 *Sub-blocks T_{21} and T_{22} are nonzero matrices.*

Proof: S_{best} in each iteration is generated by the quantum individuals with a certain probability. As we use the H_ε gate, the probability of the bit value being equal to 1 on each bit of S_{best} is between ε and $1-\varepsilon$, and the probability of the bit value being equal to 0 on each bit of S_{best} is also between ε and $1-\varepsilon$. Thus, for each quantum individual, the probability of $S_{best} = S_g$ is larger than ε^O. As there are $K \times L$ quantum individuals in the whole population, the probability of $S_{best} = S_g$ is larger than $1 - \left(1-\varepsilon^O\right)^{KL}$. So for element f_{ij} in transition matrix F within sub-block T_{21}, we have $f_{ij} \neq 0$ if $i-j = KLWO$, and $f_{ij} = 0$ if $i - j \neq KLWO$.

Similarly, for each quantum individual, the probability of $S_{best} \neq S_g$ is larger than $1-(1-\varepsilon)^O$. As there are $K \times L$ quantum individuals in the whole population, the probability of $S_{best} \neq S_g$ is larger than $\left(1-(1-\varepsilon)^O\right)^{KL}$. So for element f_{ij} in transition matrix F within sub-block T_{22}, we have $f_{ij} \neq 0$ if $i = j$ and $f_{ij} = 0$ if $i \neq j$. So sub-blocks T_{21} and T_{22} are nonzero matrices.

A.1.4 The Convergence for the PECQEA

In order to facilitate the convergence analysis, first we give a precise definition for the convergence of PECQEA:

Definition 5 *Let S_e be the elite solution of PECQEA at generation m and S_g be the unique globally optimal solution with the lowest fitness value. If $\lim_{m\to\infty} \Pr(S_e = S_g) = 1$, then the sequence $Q^+(m)_{m=1}^{\infty}$ converges to the globally optimal.*

Theorem 6 *[110]. Let A be a non-negative stochastic matrix with the form $A = \begin{pmatrix} X & 0 \\ U & V \end{pmatrix}_{k\times k}$, where $U, V \neq 0$ and X be a $z \times z$ primitive stochastic matrix, then the limit*

$$A^{\infty} = \lim_{m\to\infty} \begin{pmatrix} X^m & 0 \\ \sum_{k=0}^{m-1} V^k U X^{m-k-1} & V^m \end{pmatrix} = \begin{pmatrix} X^{\infty} & 0 \\ U^{\infty} & 0 \end{pmatrix} \tag{A.11}$$

is a stable stochastic matrix. A^{∞} has the form $A^{\infty} = g' \cdot a^{\infty}$, where $a^{\infty} = (x_1, x_2, \cdots, x_z, 0, 0, \cdots 0)$, x^{∞}=$(x_1, x_2, \cdots, x_z)$, $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times k}$, $h = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times z}$, $X^{\infty} = h' x^{\infty}$ and $\sum_{i=1}^{z} x_i = 1$.

The proof of Theorem 6 are omitted, please refer to reference [110].

Theorem 7 *PECQEA converges to the global optimum.*

Proof: According to the state order, there are $|\Omega^+| = 2^{(KLW+1)O}$ states in the whole space and the states that are numbered from 1 to 2^{KLWO} have the best S_e which is equal

to S_g. As T_{11} is a unit matrix, the transition matrix F can be represented as

$$\begin{aligned} F &= AT \\ &= \begin{pmatrix} H & & & \\ & H & & \\ & & \ddots & \\ & & & H \end{pmatrix} \begin{pmatrix} T_{11} & & & \\ T_{21} & T_{22} & & \\ \vdots & \vdots & \ddots & \\ T_{2^O,1} & T_{2^O,2} & \cdots & T_{2^O,2^O} \end{pmatrix} \\ &= \begin{pmatrix} H & & & \\ HT_{21} & HT_{22} & & \\ \vdots & \vdots & \ddots & \\ HT_{2^O,1} & HT_{2^O,2} & \cdots & HT_{2^O,2^O} \end{pmatrix} \end{aligned} \tag{A.12}$$

Thus, the transition matrix F can be divided into four square matrices, which can be represented as

$$F = \begin{pmatrix} H & 0 \\ U & V \end{pmatrix}_{(KLW+1)O\times(KLW+1)O} . \tag{A.13}$$

where H represents the transition matrix of the base Markov chain, $U = \begin{bmatrix} HT_{21} \\ \vdots \\ HT_{2^O,1} \end{bmatrix}$ and $V = \begin{pmatrix} HT_{22} & & \\ \vdots & \ddots & \\ HT_{2^O,2} & \cdots & HT_{2^O,2^O} \end{pmatrix}$. According to Lemma 4, H is a positive stochastic matrix. With Lemma 12 we know that sub-blocks T_{21} and T_{22} are nonzero matrices, so we have $U, V \neq 0$. Thus F^∞ has the form $F^\infty = g' \cdot f^\infty$, where $g = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times 2^{(KLW+1)O}}$, and $f^\infty = (h_1, h_2, \cdots, h_{2^{KLWO}}, 0, 0, \cdots 0)$.

When $l = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}_{1\times 2^{KLWO}}$, x^∞=$(x_1, x_2, \cdots, x_{2^{KLWO}})$, $X^\infty = l'x^\infty$ and $\sum_{i=1}^{KLWO} x_i = 1$.

Therefore, $Q^+(m)$ converges to state 1 to state 2^{KLWO} in $|\Omega^+| = 2^{(KLW+1)O}$ states after infinite iterations. As the solution space $|\Omega^+|$ has a decreasing order according to the fitness value of S_e and the states that are numbered from 1 to 2^{KLWO} have the best S_e which is equal to S_g, according to Definition 5, the sequence $Q^+(m)_{m=1}^{\infty}$ converges to the global optimal.

Bibliography

[1] G.Y. Li, Z. Xu, C. Xiong, C. Yang, S. Zhang, Y. Chen and S. Xu, "Energy-efficient wireless communications: tutorial, survey, and open issues," *IEEE Wireless Communications*, vol.18, pp. 28-35, 2011.

[2] D. Feng, C. Jiang, G. Lim, Jr. L. J. Cimini, G. Feng and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, vol.15, pp. 167-178, 2013.

[3] T. Hwang, C. Yang, G. Wu, S. Li and G. Y. Li, "OFDM and Its Wireless Applications: A Survey," *IEEE Transactions on Vehicular Technology*, vol.58, pp. 1673-1694, 2009.

[4] Y. Rahmatallah and S. Mohan, "Peak-To-Average Power Ratio Reduction in OFDM Systems: A Survey And Taxonomy," *IEEE Communications Surveys & Tutorials*, vol.15, pp. 1567-1592, 2013.

[5] S. A. A. Kumar, K. Ovsthus and L. M. Kristensen, "An Industrial Perspective on Wireless Sensor Networks - A Survey of Requirements, Protocols, and Challenges," *IEEE Communications Surveys & Tutorials*, vol.16, pp. 1391-1412, 2014.

[6] Z. Lu, W. W. Li and M. Pan, "Maximum Lifetime Scheduling for Target Coverage and Data Collection in Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol.64, pp. 714-727, 2015.

[7] C. Yang and K. Chin, "Novel Algorithms for Complete Targets Coverage in Energy Harvesting Wireless Sensor Networks," *IEEE Communications Letters*, vol.18, pp. 118-121, 2014.

[8] J. Hao, B. Zhang and H. T. Mouftah, "Routing protocols for duty cycled wireless sensor networks: A survey," *IEEE Communications Magazine*, vol.50, pp. 116-123, 2012.

[9] J. Esch, "A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues," *Proceedings of the IEEE*, vol.101, pp. 2534-2537, 2013.

[10] I. F. Akyildiz, T. Melodia and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol.14, pp. 32-39, 2007.

[11] N. A. Pantazis, S. A. Nikolidakis, D. D. and Vergados, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol.15, pp. 551-591, 2013.

[12] F. Li, K. Wu and A. Lippman, "Minimum energy cooperative path routing in all-wireless networks: NP-completeness and heuristic algorithms," *Journal of Communications and Networks*, vol.10, no. 2, pp. 204-212, 2008.

[13] W. C. Ke, B. H. Liu and M. J. Tsai, "Constructing a Wireless Sensor Network to Fully Cover Critical Grids by Deploying Minimum Sensors on Grid Points Is NP-Complete," *IEEE Transactions on Computers*, vol.56, no. 5, pp. 710-715, 2007.

[14] T. Back, U. Hammel and H. P. Schwefel, "Evolutionary computation: comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol.1, pp. 3-17, 1997.

[15] S. J. Ku, "Low-Complexity PTS-Based Schemes for PAPR Reduction in SFBC MIMO-OFDM Systems," *IEEE Transactions on Broadcasting*, vol.60, no. 4, pp. 650-658, 2014.

[16] Z. Liao, J. Wang, S. Zhang, J. Cao and G. Min, "Minimizing Movement for Target Coverage and Network Connectivity in Mobile Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol.26, no. 7, pp. 1971-1983, 2015.

[17] K. Han, J. Luo, L. Xiang, M. Xiao and L. Huang, "Achieving Energy Efficiency and Reliability for Data Dissemination in Duty-Cycled WSNs," *IEEE/ACM Transactions on Networking*, vol.23, no. 4, pp. 1041-1052, 2015.

[18] N. Gautam and J. Y. Pyun, "Distance aware intelligent clustering protocol for wireless sensor networks," *Journal of Communications and Networks*, vol.12, no. 2, pp. 122-129, 2010.

[19] D. Djenouri and I. Balasingham, "Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol.10, no. 6, pp. 797-809, 2011.

[20] D. Qu, S. Lu and T. Jiang, "Multi-Block Joint Optimization for the Peak-to-Average Power Ratio Reduction of FBMC-OQAM Signals," *IEEE Transactions on Signal Processing*, vol.61, no. 7, pp. 1605-1613, 2013.

[21] A. Ghassemi and T. A. Gulliver, "PAPR reduction of OFDM using PTS and error-correcting code subblocking - Transactions Papers," *IEEE Transactions on Wireless Communications*, vol.9, no. 3, pp. 980-989, 2010.

[22] L. Wang and J. Liu, "Cooperative PTS for PAPR reduction in MIMO-OFDM," *Electronics Letters*, vol.47, no. 5, pp. 351-352, 2011.

[23] L. Li and D. Qu, "Joint Decoding of LDPC Code and Phase Factors for OFDM Systems With PTS PAPR Reduction," *IEEE Transactions on Vehicular Technology*, vol.62, no.1, pp. 444-449, 2013.

[24] L. Yang, K. K. Soo, S. Q. Li and Y. M. Siu, "PAPR Reduction Using Low Complexity PTS to Construct of OFDM Signals Without Side Information," *IEEE Transactions on Broadcasting*, vol.57, no. 2, pp. 284-290, 2011.

[25] C. Ye, Z. Li, T. Jiang, C. Ni and Q. Qi, "PAPR Reduction of OQAM-OFDM Signals Using Segmental PTS Scheme With Low Complexity," *IEEE Transactions on Broadcasting*, vol.60, no. 1, pp. 141-147, March 2014.

[26] L. Li, D. Qu and T. Jiang, "Partition Optimization in LDPC-Coded OFDM Systems With PTS PAPR Reduction," *IEEE Transactions on Vehicular Technology*, vol.63, no. 8, pp. 4108-4113, 2014.

[27] S. J. Ku, C. L. Wang and C. H. Chen, "A Reduced-Complexity PTS-Based PAPR Reduction Scheme for OFDM Systems," *IEEE Transactions on Wireless Communications*, vol.9, no. 8, pp. 2455-2460, 2010.

[28] P. Varahram, W. F. Al-Azzo and B. M. Ali, "A low complexity partial transmit sequence scheme by use of dummy signals for PAPR reduction in OFDM systems," *IEEE Transactions on Consumer Electronics*, vol.56, no. 4, pp. 2416-2420, 2010.

[29] J. Chen, J. Li and T. H. Lai, "Trapping Mobile Targets in Wireless Sensor Networks: An Energy-Efficient Perspective," *IEEE Transactions on Vehicular Technology*, vol.62, no. 7, pp. 3287-3300, 2013.

[30] Y. Bejerano, "Coverage Verification without Location Information," *IEEE Transactions on Mobile Computing*, vol.11, no. 4, pp. 631-643, 2012.

[31] T. Y. Lin, H. A. Santoso and K. R. Wu, "Global Sensor Deployment and Local Coverage-Aware Recovery Schemes for Smart Environments," *IEEE Transactions on Mobile Computing*, vol.14, no. 7, pp. 1382-1396, 2015.

[32] C. Shang, G. Chen, C. Ji and C. Y. Chang, "An Efficient Target Tracking Mechanism for Guaranteeing User-Defined Tracking Quality in WSNs," *IEEE Sensors Journal*, vol.15, no. 9, pp. 5258-5271, 2015.

[33] L. Kong et al., "Surface Coverage in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol.25, no. 1, pp. 234-243, 2014.

[34] R. Carrano, D. Passos, L. Magalhaes and C. Albuquerque, "An Exact Model of the Neighbor Discovery Time for Schedule-Based Asynchronous Duty Cycling," *IEEE Wireless Communications Letters*, vol. 2, no. 6, pp. 635-638, 2013.

[35] M. Xiao, J. Wu and L. Huang, "Time-Sensitive Utility-Based Single-Copy Routing in Low-Duty-Cycle Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1452-1465, 2015.

[36] C. Zhu, L. T. Yang, L. Shu, V. C. M. Leung, J. J. P. C. Rodrigues and L. Wang, "Sleep Scheduling for Geographic Routing in Duty-Cycled Mobile Sensor Networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6346-6355, 2014.

[37] X. Gao, Y. Vanq and D. Zhou, "Coverage of communication-based sensor nodes deployed location and energy efficient clustering algorithm in WSN," *Journal of Systems Engineering and Electronics*, vol. 21, no. 4, pp. 698-704, 2010.

[38] Y. Liu, N. Xiong, Y. Zhao, A. V. Vasilakos, J. Gao and Y. Jia, "Multi-layer clustering routing algorithm for wireless vehicular sensor networks," *IET Communications*, vol. 4, no. 7, pp. 810-816, 2010.

[39] S. Ganesh and R. Amutha, "Efficient and secure routing protocol for wireless sensor networks through SNR based dynamic clustering mechanisms," *Journal of Communications and Networks*, vol. 15, no. 4, pp. 422-429, 2013.

[40] A. Thakkar and K. Kotecha, "Cluster Head Election for Energy and Delay Constraint Applications of Wireless Sensor Network," *IEEE Sensors Journal*, vol. 14, no. 8, pp. 2658-2664, 2014.

[41] B. C. Cheng, H. H. Yeh and P. H. Hsu, "Schedulability Analysis for Hard Network Lifetime Wireless Sensor Networks With High Energy First Clustering," *IEEE Transactions on Reliability*, vol. 60, no. 3, pp. 675-688, 2011.

[42] P. Nayak and A. Devulapalli, "A Fuzzy Logic-Based Clustering Algorithm for WSN to Extend the Network Lifetime," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 137-144, 2016.

[43] L. Mottola and G. P. Picco, "MUSTER: Adaptive Energy-Aware Multisink Routing in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1694-1709, Dec. 2011.

[44] G. A. Shah, V. C. Gungor and O. B. Akan, "A Cross-Layer QoS-Aware Communication Framework in Cognitive Radio Sensor Networks for Smart Grid Applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1477-1485, Aug. 2013.

[45] D. Zhang, G. Li, K. Zheng, X. Ming and Z. H. Pan, "An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 766-773, Feb. 2014.

[46] J. Zhang, F. Ren, S. Gao, H. Yang and C. Lin, "Dynamic Routing for Data Integrity and Delay Differentiated Services in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 328-343, Feb. 1 2015.

[47] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley, 1966.

[48] I. Rechenberg, "Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Ph.D. dissertation, reprinted by Fromman-Holzboog, 1973.

[49] J. Jagerskupper, "How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms," *Theoretical Computer Science*, vol.361, pp. 38-56, 2006.

[50] N. Hansen and S. Kern, "Evaluating the CMA Evolution Strategy on Multimodal Test Functions," *Parallel Problem Solving from Nature, PPSN VIII*, pp. 282-291, 2004.

[51] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*, Published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk, 2008.

[52] S. Steven, *The Algorithm Design Manual (2nd ed.)*, Springer Science+Business Media, New York, USA, 2010.

[53] Y. Wu and W. Liu, "Routing protocol based on genetic algorithm for energy harvesting-wireless sensor networks ," *IET Wireless Sensor Systems*, vol. 3, pp. 112-118, 2013.

[54] X. M. Hu, J. Zhang, Y. Yu, , H. S. H. Chung, Y. L. Li, Y. H. Shi and X. N. Luo, "Hybrid Genetic Algorithm Using a Forward Encoding Scheme for Lifetime Maximization of Wireless Sensor Networks," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 766-781, 2010.

[55] W. Cheng, H. Shi, X. Yin and D. Li, "An Elitism Strategy Based Genetic Algorithm for Streaming Pattern Discovery in Wireless Sensor Networks," *IEEE Communications Letters*, vol. 15, pp. 419-421, 2011.

[56] C. Ting, C. Lee, H. Chang and J. Wu, "Wireless Heterogeneous Transmitter Placement Using Multiobjective Variable-Length Genetic Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 945-958, 2009.

[57] Y. Yoon, Y. Kim, "An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks," *IEEE Transactions on Cybernetics*, vol. 43, pp.1473-1483, 2013.

[58] N. Sharma and A. S. Madhukumar, "Genetic Algorithm Aided Proportional Fair Resource Allocation in Multicast OFDM Systems," *IEEE Transactions on Broadcasting*, vol. 61, pp. 16-29, 2015.

[59] T. Lin, K. Hsieh and H. Huang, "Applying Genetic Algorithms for Multiradio Wireless Mesh Network Planning," *IEEE Transactions on Vehicular Technology*, vol.61, pp. 2256-2270, 2012.

[60] B. Lorenzo and S. Glisic, "Optimal Routing and Traffic Scheduling for Multihop Cellular Networks Using Genetic Algorithm," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 2274-2288, 2013.

[61] S. Yang, H. Cheng and F. Wang, "Genetic Algorithms With Immigrants and Memory Schemes for Dynamic Shortest Path Routing Problems in Mobile Ad Hoc Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, pp. 52-63, 2010.

[62] T. Lu and J. Zhu, "Genetic Algorithm for Energy-Efficient QoS Multicast Routing," *IEEE Communications Letters*, vol. 17, pp. 31-34, 2013.

[63] L. M. O. Khanbary and D. P. Vidyarthi, "Reliability-Based Channel Allocation Using Genetic Algorithm in Mobile Computing," *IEEE Transactions on Vehicular Technology*, vol. 58, pp.4248-4256, 2009.

[64] Beni G. and Wang J. "Swarm Intelligence in Cellular Robotic Systems", in Proc. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26-30, 1989.

[65] A. Colorni, M. Dorigo and V. Maniezzo, *Distributed Optimization by Ant Colonies*, Elsevier Publishing, Paris, France, 1991.

[66] S. Scott-Hayward and E. Garcia-Palacios, "Channel Time Allocation PSO for Gigabit Multimedia Wireless Networks," *IEEE Transactions on Multimedia*, vol. 16, pp. 828-836, 2014.

[67] T. Wimalajeewa and S. K. Jayaweera, "Optimal Power Scheduling for Correlated Data Fusion in Wireless Sensor Networks via Constrained PSO," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3608-3618, 2008.

[68] C. Chen, X. Gao, Q. Pei and X. Li, "BNE-based concurrent transmission considering channel quality and its PSO searching strategy in Ad Hoc networks," *Journal of Systems Engineering and Electronics*, vol. 23, pp. 649-660, 2012.

[69] E. R. Dosciatti, W. Godoy and A. Foronda, "TQ/PSO - A New Scheduler to Optimize the Time Frame with PSO in WiMAX Networks," *IEEE Latin America Transactions*, vol. 13, pp. 365-376, 2015.

[70] N. Taspnar, D. Karaboga, M. Yildirim and B. Akay, "Partial transmit sequences based on artificial bee colony algorithm for peak-to-average power ratio reduction in multicarrier code division multiple access systems," *IET Communications*, vol. 5, pp. 1155-1162, 2011.

[71] Y. Wang, W. Chen and C. Tellambura, "A PAPR Reduction Method Based on Artificial Bee Colony Algorithm for OFDM Signals," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2994-2999, 2010.

[72] Z. Zong, Z. Yin and Z. Wu, "The population declining AFSA based identification of active users for cooperative cognitive radio networks with DS-UWB based control channel," in Proc. 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Shanghai, China, September 21-23, 2012.

[73] J.D. Farmer, N. Packard and A. Perelson, "The immune system, adaptation and machine learning", *Physica D*, vol. 2, pp. 187-204, 1986.

[74] S. Forrest, A. S. Perelson, L. Allen and R. Cherukuri, "Self-Nonself Discrimination in a Computer", in *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, 1994.

[75] N. K. Jerne, "Towards a network theory of the immune system", *Ann. Immunol*, 125C, 373-389, 1974.

[76] F. M. Burnet, "A modification of Jerne's theory of antibody production using the concept of clonal selection", *CA: A Cancer Journal for Clinicians* vol. 26, pp. 119-21, 1976.

[77] G. J. V. Nossal, L. Joshua, "Antibody Production by Single Cells", *Nature* vol. 181, pp. 1419-1420, 1958.

[78] S. Sarafijanovic and J. L. Boudec, "An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks," *IEEE Transactions on Neural Networks*, vol. 16, pp. 1076-1087, 2005.

[79] G. V. P. Kumar and D. K. Reddy, "An agent based intrusion detection system for wireless network with artificial immune system (AIS) and negative clone selection," in Proc. 2014 International Conference on Signal Processing and Computing Technologies in Electronic Systems (ICESC), Guilin, China, August 5-8, 2014.

[80] M. A. Lebbe, J. I. Agbinya, Z. Chaczko and R. Braun, "Artificial immune system inspired danger modelling in wireless mesh networks," in Proc. International Conference on Computer and Communication Engineering (ICCCE) 2008, May 13-15, Kuala Lumpur, Malaysia, 2008.

[81] X. Lu, Y. Ding and K. Hao, "Adaptive design optimization of wireless sensor networks using artificial immune algorithms," in Proc. IEEE World Congress on Computational Intelligence 2008, June 1-6, Hong Kong, China, 2008.

[82] S. I. Suliman, G. Kendall and I. Musirin, "Artificial immune algorithm in solving the channel assignment task," in Proc. 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, November 28-30, 2014.

[83] F. Barani, "A hybrid approach for dynamic intrusion detection in ad hoc networks using genetic algorithm and artificial immune system," in Proc. 2014 Iranian Conference on Intelligent Systems (ICIS), Bam, Iran, February 4-6, 2014.

[84] A. Visconti and H. Tahayori, "Detecting misbehaving nodes in MANET with an artificial immune system based on type-2 fuzzy sets," in Proc. International Conference for Internet Technology and Secured Transactions(ICITST), London, UK, November 9-13, 2009.

[85] H. Jeon, K. Kim, J. No and D. Shin, "Bit-Based SLM Schemes for PAPR Reduction in QAM Modulated OFDM Signals," *IEEE Trans. Broadcast.*, vol. 55, pp. 679-685, 2009.

[86] R. P. Feynman, "Simulating physics with computers". *International Journal of Theoretical Physics*, vol.21, pp. 467-488, 1982.

[87] D. Alanis, P. Botsinis, X. N. Soon and L. Hanzo, "Quantum-Assisted Routing Optimization for Self-Organizing Networks," *IEEE Access*, vol. 2, pp. 614-632, 2014.

[88] Z. Zhao, Z. Peng, S. Zheng and J. Shang, "Cognitive radio spectrum allocation using evolutionary algorithms," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 4421-4425, 2009.

[89] H. M. Ali, S. Ashrafinia, J. Liu and D. Lee, "Broadband wireless network planning using evolutionary algorithms," in Proc. 2013 IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, June 20-23, 2013.

[90] H. M. Ali, S. Ashrafinia, J. Liu and D. C. Lee, "Wireless mesh network planning using quantum inspired evolutionary algorithm," in Proc. 2011 IEEE Vehicular Technology Conference (VTC Fall), San Francisco, USA, September 5-8, 2011.

[91] L. Ni and S. Du, "Quantum evolutionary complex sphere decoder algorithm for MIMO systems," in Proc. 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Shenzhen, China, September 23-25, 2010.

[92] O. Muta, "Effect of phase control-based peak-to-average power ratio reduction on multi-input multi-output adaptive modulated vector coding systems," *IET Commun.*, vol. 6, pp. 2769-2774, 2012.

[93] G. Lu, P. Wu and D. Aronsson, "Peak-to-average power ratio reduction in OFDM using cyclically shifted phase sequences," *IET Commun.*, vol. 1, pp. 1146-1151, 2007.

[94] E. Hong, Y. Park, S. Lim and D. Har, "Adaptive phase rotation of OFDM signals for PAPR reduction," *IEEE Trans. Consum. Electron.*, vol. 57, pp. 1491-1495, 2011.

[95] X. Zhu, W. Pan, H. Li and Y. Tang, "Simplified Approach to Optimized Iterative Clipping and Filtering for PAPR Reduction of OFDM Signals," *IEEE Trans. Commun.*, vol. 61, pp. 1891-1901, 2013.

[96] Y. Wang, W. Chen and C. Tellambura, "Genetic Algorithm Based Nearly Optimal Peak Reduction Tone Set Selection for Adaptive Amplitude Clipping PAPR Reduction," *IEEE Trans. Broadcast.*, vol. 58, pp. 462-471, 2012.

[97] J. Wang, Y. Guo and X. Zhou, "PTS-clipping method to reduce the PAPR in ROF-OFDM system," *IEEE Trans. Consum. Electron.*, vol. 55, pp. 356-359, 2009.

[98] J. Chen, "Partial transmit sequences for PAPR reduction of OFDM signals with stochastic optimization techniques," *IEEE Trans. Consum. Electron.*, vol. 56, pp. 1229-1234, 2010.

[99] C. M. Grinstead, J. L. Snell, *Grinstead and Snell's Introduction to Probability*, pp.433, 2009.

[100] J. Chen and C. Wen, "A Low-Complexity Scheme to Reduce the PAPR of an OFDM Signal Using Sign-Selection Algorithms," *IEEE Signal Process. Lett.*, vol. 17, pp. 189-192, 2010.

[101] J. Chen, "Application of Quantum-Inspired Evolutionary Algorithm to Reduce PAPR of an OFDM Signal Using Partial Transmit Sequences Technique," *IEEE Trans. Broadcast.*, vol. 56, pp. 110-113, 2010.

[102] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, pp. 29-41, 1996.

[103] Y. Wang, W. Chen, and C. Tellambura, "PAPR Reduction Method Based on Parametric Minimum Cross Entropy for OFDM Signals," *IEEE Communications Letters*, vol. 14, no. 6, pp.563-565, 2010.

[104] J. K. Lain, S. Y. Wu, and P. H. Yang, "PAPR reduction of OFDM signals using PTS:a real-valued genetic approach," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, pp.1-8, 2011.

[105] F. S. Abu-Mouti and M. E. El-Hawary, "Optimal Distributed Generation Allocation and Sizing in Distribution Systems via Artificial Bee Colony Algorithm," *IEEE Trans. Power Del.*, vol. 26, pp. 2090-2101, 2011.

[106] M. M. Eusuff, K. E. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no.2, pp. 129-154, 2006.

[107] Y. S. Cho, J. Kim, W. Y. Yang and C. G. Kang, "MIMO-OFDM Wireless Communications with MATLAB", *John Wiley and Sons (Asia) and IEEE Press*, Singapore, 2010.

[108] A. Khorsandi, A. Alimardani, B. Vahidi and S. H. Hosseinian, "Hybrid shuffled frog leaping algorithm and Nelder-Mead simplex search for optimal reactive power dispatch," *IET Gener. Transm. Distrib.*, vol. 5, pp. 249-256, 2011.

[109] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol.261, pp.459-467, 1976.

[110] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Transactions on Neural Networks*, vol. 5, pp. 96-101, 1994.

[111] H. Li, T. Jiang and Y. Zhou, "A Novel Subblock Linear Combination Scheme for Peak-to-Average Power Ratio Reduction in OFDM Systems," *IEEE Transactions on Broadcasting*, vol. 58, pp. 360-369, 2012.

[112] M. Iosifescu, *Finite Markov Processes and Their Applications. Chichester: Wiley*, pp.126, 1980.

[113] D. Simon, M. Ergezer, D. W. Du and R. Rarick, "Markov Models for Biogeography-Based Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, pp. 299-306, 2011.

[114] S. Y. Yuen and K. S. Cheung, "Bounds for probability of success of classical genetic algorithm based on hamming distance," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 1-18, 2006.

[115] O. Francois, "An evolutionary strategy for global minimization and its Markov chain analysis," *IEEE Transactions on Evolutionary Computation*, vol. 2, pp. 77-90, 1998.

[116] J. Suzuki, "A Markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, pp. 655-659, 1995.

[117] R. Nowotniak and J. Kucharski, "GPU-based tuning of quantum-inspired genetic algorithm for a combinatorial optimization problem," *Bull. Pol. Ac.: Tech.*, vol. 60, pp. 323-330, 2012.

[118] J. Zhou, E. Dutkiewicz, R. P. Liu, G. F. Fang, Y. A. Liu and X. J. Huang, "A Modified Shuffled Frog Leaping Algorithm for PAPR Reduction in OFDM Systems," in Proc. IEEE TENCON 2014, Bangkok, Thailand, October 22-25, 2014.

[119] H. Pei, X. Li, S. Soltani, M.W. Mutka, and X. Ning, "The Evolution of MAC Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 101-120, 2013.

[120] D. Dong, X. Liao, K. Liu, Y. Liu, and W. Xu, "Distributed Coverage in Wireless Ad Hoc and Sensor Networks by Topological Graph Approaches," *IEEE Transactions on Computers*, vol. 61, pp. 1417-1428, 2012.

[121] V. Akbarzadeh, C. Gagne, M. Parizeau, M. Argany, and M.A. Mostafavi, "Probabilistic Sensing Model for Sensor Placement Optimization Based on Line-of-Sight Coverage," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, pp. 293-303, 2013.

[122] X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, pp. 354-370, 2007.

[123] M. Cardei, M. T. Thai , and Y. Li, "Energy-Efficient Target Coverage in Wireless Sensor Networks," in Proc. IEEE INFOCOM 2005, Miami, USA, March 13-17, 2005.

[124] L. Aslanyan, H. Aslanyan, and H. Khosravi, "Optimal node scheduling for integrated connected-coverage in wireless sensor networks," in Proc. the Computer Science and Information Technologies (CSIT) 2013, Yerevan, September 23-27, 2013.

[125] A. H. Navin and M. K. Mirnia , "Solving coverage problem in wireless camera-based sensor networks by using a distributed evolutionary algorithm," in Proc. the 2014 5th IEEE International Conference on Software Engineering and Service Science, Beijing, China, June 27-29, 2014.

[126] L. Mo, L. Zhenjiang, A. V. Vasilakos, "A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues," *Proceedings of the IEEE*, vol. 101, pp. 2538-2557, 2013.

[127] O. Demigha, W. K. Hidouci and T. Ahmed, "On Energy Efficiency in Collaborative Target Tracking in Wireless Sensor Network: A Review," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 1210-1222, 2013.

[128] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660-670, 2002.

[129] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing,* vol. 3, pp. 366-379, 2004.

[130] X. Li, "Research on clustered-network in wireless sensor networks," Ph.D. dissertation, Tianjin University, Tianjin, China, 2006.

[131] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proceedings of the Aerospace Conference*, 2002, pp. 3-1125-3-1130.

[132] M. Srinivas, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 24, pp. 656-667, 1994.

[133] K. M. Passino and S. Yurkovich, "Fuzzy Control", *Addison Wesley Longman*, Menlo Park, California, 1998.

[134] H. Ying, "Fuzzy Control and Modeling:Analytical Foundations and Applications", *Wiley-IEEE Press*, New York, 2000.

[135] Q. Li, "A new kind of fuzzy genetic algorithm," *Journal of University of Science and Technology Beijing,* vol. 23, pp. 85-89, 2001.

[136] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for Wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaill International Conference on System Sciences*, 2000, pp. 1-10.

[137] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems,* vol. 4, pp. 103, 1996.

[138] S. Choi, S. Gong and J. Lee, "An average velocity-based routing protocol with low end-to-end delay for wireless sensor networks," *IEEE Commun. Lett.*, vol. 13, pp. 621-623, 2009.

[139] S. Lai and B. Ravindran, "Least-Latency Routing over Time-Dependent Wireless Sensor Networks," *IEEE Trans Comput.*, vol. 62, pp. 969-983, 2013.

[140] J. Heo, J. Hong and Y. Cho, "EARQ: Energy Aware Routing for Real-Time and Reliable Communication in Wireless Industrial Sensor Networks," *IEEE Trans Ind. Informat.*, vol. 5, pp. 3-11, 2009.

[141] H. Zhang and H. Shen, "Energy-Efficient Beaconless Geographic Routing in Wireless Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, pp. 881-896, 2010.

[142] S. Ehsan and B. Hamdaoui, "A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks," *IEEE Commun. Surveys Tuts.*, vol. 14, pp. 265-278, 2012.

[143] D. S. Reeves and H. F. Salama, "A distributed algorithm for delay-constrained unicast routing," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 239-250, 2000.

[144] S. S. Iyengar, H. Wu, N. Balakrishnan and S. Chang, "Biologically Inspired Cooperative Routing for Wireless Mobile Sensor Networks," *IEEE Syst. J., vol. 1*, pp. 29-37, 2007.

[145] F. Ren, J. Zhang, Y. Wu, T. He, C. Chen and C. Lin, "Attribute-Aware Data Aggregation Using Potential-Based Dynamic Routing in Wireless Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, pp. 881-892, 2013.

[146] G. Froc, I. Mabrouki and X. Lagrange, "Design and Performance of Wireless Data Gathering Networks Based on Unicast Random Walk Routing," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 1214-1227, 2009.

[147] C. Wei, C. Zhi, P. Fan and K. B. Letaief, "AsOR: an energy efficient multi-hop opportunistic routing protocol for wireless sensor networks over Rayleigh fading channels," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 2452-2463, 2009.

[148] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 580-593, 2002.

[149] C. H. Bennett and P. W. Shor, "Quantum information theory," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2724-2742, 1998.

[150] J. Wang, "Ant Colony Algorithm and Application in QoS Routing," M.S. thesis, Xi'an University of Electronic Science and Technology, Xi'an, China, 2009.

[151] K. Han and J. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, H_ε gate, and two-phase scheme," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 156-169, 2004.