SINGLE-OBJECTIVE AND MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS BASED ON SPERM FERTILIZATION PROCEDURE

HISHAM AHMAD THEEB SHEHADEH

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

SINGLE-OBJECTIVE AND MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS BASED ON SPERM FERTILIZATION PROCEDURE

HISHAM AHMAD THEEB SHEHADEH

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Hisham Ahmad Theeb Shehadeh

Matric No: WHA160032

Name of Degree: Ph.D in Computer Science

Title of Thesis: Single-Objective and Multi-Objective Optimization Algorithms Based on Sperm Fertilization Procedure.

Field of Study: wireless sensor networks.

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

SINGLE-OBJECTIVE AND MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS BASED ON SPERM FERTILIZATION PROCEDURE

ABSTRACT

In this work, Single Objective Optimization Algorithm (SOOA) is proposed. The SOOA version is extended to Multi Objective Optimization Algorithm (MOOA). To demonstrate the applicability of the proposed MOOA, a set of Wireless Sensor Network (WSN) problems is optimized. In SOOA, a novel metaheuristic approach based on a metaphor of a natural fertilization procedure, called “Sperm Swarm Optimization (SSO)” is proposed. In this approach, an optimization model of a sperm fertilization procedure is devised. The model follows the characteristics of sperm swarm, which moves forward from a low-temperature zone called Cervix. During this direction, sperm searches for a high-temperature zone called Fallopian Tubes where the egg is waiting for the swarm to fertilize at this zone, which this area is considered as the optimal solution. The SSO is tested with several benchmark functions used in the area of optimization. The obtained results are compared with the results of four algorithms. These algorithms are Genetic Algorithms (GA), Parallel Genetic Algorithm (PGA), Particle Swarm Optimization (PSO) and Accelerated Particle Swarm Optimization (APSO). The results show that the proposed SOOA outperformed other SOOAs algorithms in term of convergence and quality of the result. Then, the SSO has been extended to MOOA, called “Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP)” depends on Pareto dominance, mutation operations and a crowding factor, that crowd and filter out the list of the best sperms (global best values). The proposed MOSFP is compared against three well-known MOOAs in the field of optimization. These algorithms are SPEA2, NSGA-II, and OMOPSO. The experimental results show that the efficiency and performance of the proposed MOSFP are highly competitive, which outperformed both of SPEA2 and NSGA-II algorithms in solving all the problems. In addition, the proposed MOSFP outperformed OMOPSO in solving problems such as WFG5, WFG8, and ZDT3. At the end, the proposed MOSFP has been used to solve a real-life problem such as optimizing a set of Quality of Services (QoS) objective functions (network models) in WSN. These objective functions are end-to-end latency, end-to-end delay, energy efficiency and network throughput. The optimal value of packet payload size that able to maximize the energy efficiency and network throughput as well as to minimize the end-to-end latency and end-to-end delay is sought. The result of the proposed MOSFP is compared against SPEA2, NSGA-II, and OMOPSO. Different packet payload sizes are supplied to the algorithms and their optimal value is derived. From the experiments, the intersection point and the knee point of all the obtained Pareto fronts for all the algorithms show that the optimal packet payload size that balances and manages the trade-offs between the four network models is equal to 45 bytes. The results also show that the performance of our proposed MOSFP is highly competitive and have the best average value compared to the other three algorithms. Furthermore, the overall performance of MOSFP from four models outperformed SPEA2, NSGA-II, and OMOPSO by 51%, 6% and 3% respectively.

Keywords: Single-Objective Optimization (SOO), Multi-Objective Optimization (MOO), Sperm Swarm Optimization (SSO), Optimization, Optimality.

ALGORITMA OPTIMISASI TANAH-OBJEKTIF DAN MULTI-OBJEKTIF BERDASARKAN PROSEDUR PERUBAHAN SPERM

ABSTRAK

Dalam karya ini, Algoritma Pengoptimuman Objektif Tunggal (SOOA) dicadangkan. Versi SOOA diperluaskan ke Algoritma Pengoptimuman Multi Objektif (MOOA). Untuk menunjukkan kebolehgunaan algoritma yang dicadangkan, ia digunakan untuk mengoptimumkan satu set masalah Rangkaian Sensor Tanpa Wayar (WSN). Dalam SOOA, pendekatan metaheuristik yang novel berdasarkan metafora prosedur persenyawaan semulajadi, yang dipanggil "Optimasi Kawanan Sperma (SSO)" telah dicadangkan. Dalam pendekatan ini, satu model pengoptimuman prosedur persenyawaan sperma telah dirangka. Model ini mengikuti ciri-ciri sperma, yang bergerak ke hadapan dari zon suhu rendah iaitu serviks. Semasa pergerakan ini, sperma mencari zon suhu tinggi iaitu tiub fallopio di mana telur sedang menunggu kawanan untuk menyuburkan zon ini, yang mana kawasan ini dianggap serbagai penyelesaian optimum. SSO diuji dengan beberapa fungsi penanda aras yang digunakan dalam bidang pengoptimuman. Hasil yang diperoleh dibandingkan dengan hasil daripada empat algoritma. Algoritma ini adalah Algoritma Genetik (GA), Algoritma Genetik Selari (PGA), Pengoptimuman Kawanan Partikel (PSO) dan Pengoptimuman Kawanan Partikel Dipercepat (APSO). Keputusan menunjukkan bahawa SOOA yang dicadangkan mengatasi algoritma lain dari segi penumpuan dan kualiti keputusan. Selepas itu, SSO telah diperluaskan kepada MOOA, yang dipanggil "Algoritma Pengoptimuman Multi-Objektif Berdasarkan Prosedur Persenyawaan Sperma (MOSFP)" yang bergantung kepada dominasi Pareto, operasi mutasi dan faktor pemanjangan, dimana mengumpul dan menapis senarai sperma yang terbaik (nilai terbaik global). MOSFP yang dicadangkan dibandingkan dengan tiga algoritma yang terkenal dalam bidang pengoptimuman. Algoritma ini adalah SPEA2, NSGA-II, dan

OMOPSO. Keputusan eksperimen menunjukkan bahawa kecekapan dan prestasi MOSFP yang dicadangkan sangat kompetitif dan mengatasi kedua-dua algoritma SPEA2 dan NSGA-II dalam menyelesaikan semua masalah. Di samping itu, MOSFP yang dicadangkan mengatasi OMOPSO dalam menyelesaikan masalah seperti WFG5, WFG8, dan ZDT3. Pada akhirnya, MOSFP yang dicadangkan telah digunakan untuk menyelesaikan masalah kehidupan sebenar seperti mengoptimumkan satu set model Kualiti Perkhidmatan (QoS) di WSN. Model-model ini adalah latensi akhir-ke-akhir, kelewatan akhir-ke-akhir, kecekapan tenaga dan penghantaran rangkaian. Nilai optimum bagi saiz muatan paket yang dapat memaksimumkan kecekapan tenaga dan penghantaran rangkaian dan juga untuk meminimumkan latensi akhir-ke-akhir dan kelewatan akhir-ke-akhir dicari. Hasil daripada MOSFP yang dicadangkan dibandingkan dengan tiga algoritma yang terkenal dalam bidang pengoptimuman. Algoritma ini adalah SPEA2, NSGA-II, dan OMOPSO. Saiz muatan paket yang berbeza dibekalkan kepada algoritma-algoritma untuk memperoleh nilai optimum mereka. Dari eksperimen ini, titik persimpangan dan titik lutut bagi semua bahagian Pareto yang diperolehi untuk semua algoritma menunjukkan bahawa saiz muatan paket yang optimum bagi mengimbangi dan menguruskan pertukaran antara empat fungsi objektif rangkaian adalah sama dengan 45 bait. Hasilnya juga menunjukkan bahawa prestasi kaedah MOSFP yang dicadangkan sangat berdaya saing dan didapati mempunyai nilai purata yang terbaik berbanding tiga algoritma yang lain. Selain itu, prestasi keseluruhan MOSFP dari empat fungsi objektif melebihi SPEA2, NSGA-II, dan OMOPSO sebanyak 51%, 6% dan 3%.

ACKNOWLEDGEMENTS

بسم الله الرَّحْمَنِ الرَّحِيمِ ﴿ قُلْ إِنَّ صَلَاتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلَّهِ رَبِّ الْعَالَمِينَ ﴾ صدق اللَّهُ العظيم

First of all, I am thankful to Allah Almighty for endowing me the strength, wisdom, and endless blessings to study.

I would like to sincerely express my deepest gratitude to my supervisors, Assoc. Prof. Dr. Mohd Yamani Idris, and Dr. Ismail Ahmedy for their invaluable guidance, supervision, and encouragement to me throughout this research. Their continuous guidance and support assisted me conducting a valuable piece of research reported in this thesis. They provided me the opportunity to broaden my experience and prepare me for future challenges. In addition, I am deeply indebted and grateful to Assoc. Prof. Dr. Mohd Yamani Idris for his extensive technical guidance and his wonderful support.

I would also like to express my sincerest gratitude and special appreciation to my parents and siblings for their endless love and support during my PhD journey. Without their moral support, this dissertation would never have been complete. Words cannot express how grateful I am to my parents and siblings for all of the sacrifices that they have made on my behalf so I dedicate highest achievement of my student life to them.

Finally, I would like to thank Faculty of Computer Science and Information Technology, University of Malaya, for the research grants and financial supports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ACS	:	Ant Colony Search Algorithm
APSO	:	Accelerated PSO
AMO	:	Animal Migration Optimization
ABC	:	Artificial Bee Colony
APHE	:	Axis Parallel Hyper-Ellipsoid
AMI	:	Advanced Metering Infrastructure
BPSO	:	Binary Particle Swarm Optimization
BA	:	Bat Algorithm
BBCA	:	Big Bang-big Crunch Algorithm
CLOP	:	Coverage and Lifetime Optimization
COA	:	Convex Optimization Algorithms
CS	:	Cuckoo Search
CSS	:	Charged System Search
CSMA	:	Carrier Sense Multiple Access
DE	:	Differential Evolution
DAU	:	Data Aggregator Unit
EHF	:	Extremely High Frequency
ES	:	Evolutionary Strategy
EMO	:	Evolutionary Multi-objective Optimization
EA	:	Evolutionary algorithms
GA	:	Genetic algorithm
GP	:	Genetic Programming
HS	:	Harmony Search
HSA	:	Hurricane Search algorithm

HAN : Home Area Network

ITS : Intelligent Transportation Systems

ISM : Industrial, Scientific and Medical band

ICA : Imperialist Competitive Algorithm

IGD : Inverted Generational Distance

Log : Logarithm

MOPSO : Multi-Objective Particle Swarm Optimization

MOEA/D : Multi-objective Evolutionary Algorithm Based on Decomposition

MOSFP : Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure

Mod : Modulus

MPE : Manual of Political Economy

MOO : Multi-Objective Optimization

MOOP : Multi-Objective Optimization Problems

MDMS : Meter Data Management System

NSGA : Non-dominated Sorting Genetic Algorithm

NAN : Neighborhood Area Network

OGDC : Optimal Geography Density Control

PSO : Particle Swarm Optimization

PGA : Parallel Genetic Algorithm

CSP : PSO-Cyclic Shift Population

PBIL : Probability-Based Incremental Learning

P : Pareto front

PER : Packet Error Rate

QoS : Quality of Service

SOO : Single-Objective Optimization

SSO	:	Sperm Swarm Optimization
SOEA	:	Single Objective Evolutionary Algorithms
SOMA	:	Self-Organizing Migrating Algorithm
SIMO	:	Swarm Intilligance Multi-objective Optimization
SP	:	Spread
SIA	:	Swarm Intelligence Algorithms
SONET	:	Synchronous Optical Network
WSN	:	Wireless Sensor Network
WFG	:	Walking-Fish-Group
WAN	:	Wide Area Network
WAMS	:	Wide-Area Measurement System
ZDT	:	Zitzler-Deb-Thiele

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the last decades, natural inspired algorithms and heuristic approaches have experienced a rapid growth in different fields such as the socio-economic systems, engineering and industrial fields. These algorithms are dynamic and effective in solving hard and real-world optimization problems (Alami & Imrani, 2008) such as combinatorial optimization problems (Beheshti et al. 2013), data mining, image processing, neural network training, pattern recognition (Benameur et al. 2009), objective function optimization, etc. (El Imrani et al. 2000; Alami et al. 2007).

The algorithms to solve the optimization problems can be categorized as Single Objective Optimization Algorithm (SOOA) and Multi Objective Optimization Algorithm (MOOA). SOOAs search optimum solution for single objective function of maximization or minimization function. Examples of SOOAs are Genetic Algorithm (GA) (Holland et al. 1992), Parallel Genetic Algorithm (PGA) (Muhlenbein, 1992), Ant Colony Search Algorithm (Dorigo et al. 1996), Particle Swarm Optimization (PSO) (Kennedy et al. 1995), Accelerated PSO (APSO) (Gandomi, 2013), and Simulated Annealing (Kirkpatrick et al. 1984).

Most of these algorithms have been extended to MOOAs to balance and manage a set of conflicting objective functions at the same time. For example, Non-dominated Sorting Genetic Algorithm (NSGA) and Non-dominated Sorting Genetic Algorithm (NSGA-II) ((Coello, 2007) ; (Srinivas, 1994)) are the extended versions of GA to optimize Multi-Objective Optimization Problems (MOOPs). Multi-Objective Particle Swarm Optimization (MOPSO) and Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) are the extended versions of PSO.

These algorithms are widely accepted to solve problems in different fields. However, these algorithms suffers from low solving precision, slow convergence, and bad local searching ability (Zingg et al. 2008; De et al. 2015; Bai et al. 2010). These limitations motivated us to propose a single objective optimization algorithm (meta-heuristic method) based on sperm fertilization procedure named as "Sperm Swarm Optimization (SSO)". SSO is presented in Chapter 4 of this thesis. Then, the proposed SSO is extended to multi-objective optimization version to optimize conflicting objective functions. In SSO, a mathematical model simulating sperm swarm motility during the fertilization process is developed. The model reflects the sperm movement from Cervix zone to the location of the egg in Fallopian tubes. Each sperm has a velocity, called sperm personal solution. This velocity will be changed in the memory just if the current solution is better than the old solution. The global velocity of the swarm is represented by the closest sperm to the target (closest solution to the optimal solution). To ensure the reliability and efficiency of the proposed SSO method, they are evaluated through several stages. The evaluation starts with commonly used benchmark functions such as Sphere, Rosenbrock and Rastrigin ((Shi, & Eberhart, 1999) ;(Rbouh & Imrani, 2014)). These functions contain the element of single local minima and several local minima that able to test the performance of SOOA through the quality of result metric and convergence metric. The proposed SSO algorithm with other well-established SOOA such as PSO, APSO, PGA, and GA are tested with these benchmark functions. Their performances are compared and analyzed to understand their advantages and limitations.

SOOA are effective in optimizing single objective problems such as minimization or maximization. However, real-life problems often contain conflicting problems of both minimization and maximization. For this reason, the proposed SSO algorithm is extended to MOOA called MOSFP as presented in Chapter 5 of this thesis. The MOSFP used the concept of Pareto optimality. Non-dominate solutions are used to balance a set of conflicting objective functions. Similar to its SOOA version, the MOSFP is evaluated by utilizing benchmark functions in MOOA. These benchmark functions include Zitzler-Deb-Thiele (ZDT) test suite, and Walking-Fish-Group (WFG) test suite (Huband, 2006). The proposed MOSFP together with well-known MOOA such as SPEA2, NSGA-II, and OMOPSO are tested with these benchmark functions. Three notions are chosen to compare the performance of the proposed MOSFP with prior MOOA methods. These notions are minimizing the distance between the global Pareto front and the true Pareto front of any problem, maximizing the spread of the solutions, maximizing the convergence of the algorithm. To answer these notions, three metrics are chosen: Inverted Generational Distance (IGD), Spread (SP), and Epsilon ($\in$) ((Patnaik & Mandavilli, 1996); (Monroy, 2004); (Riquelme, 2015)). Qualitative and quantitative techniques of comparing the results are used. In the qualitative technique, a comparison between the proposed algorithm and other algorithms in term of quality of Pareto front for each benchmark function is conducted. In the quantitative technique, a comparison between the proposed algorithm and other algorithms in terms of median, average, best, and worst of each metric for each benchmark function is conducted.

To demonstrate the capability of the proposed MOOA, MOSFP is tested with Wireless Sensor Network (WSN) application as presented in Chapter 6 of this thesis. The prior works were focused on two conflicting problems, in which, minimizing energy consumption and maximizing network coverage, but some situations may need different optimization. For example, packet payload size can affect some of the important network model features such as end-to-end delay, end-to-end latency, energy efficiency, and network throughput. If the packet payload size is large, this will consume more energy and time through the process of packetizing and transmission. It is a challenging task to realize the optimal packet payload size that can increase the network QoS. Two ways can be used to realize the optimal packet payload size that increases the QoS of the network. First, is through empirical studies, which sensor nodes in WSN are configured on different packet payload sizes to determine the optimal packet payload size that manages the network as in ((Al-Anbagi, 2015); (Brown, 2012); (Sankarasubramaniam, 2003); (Liang, 2007)). However, the empirical study consumes more time and efforts to determine the optimal packet payload size that increases the QoS of the network. Second, is to use meta-heuristic methods to realize the optimal packet payload size as we will demonstrate in Chapter 6. We consider a smart grid application as a case study to find the optimal packet payload size. In a smart grid application, real-time data are important to monitor and generate power in a real-time, control power outage, monitor power quality, and control power load. This helps the power companies to use the data to develop a real-time pricing. Consumers can use the information of real-time power pricing to reduce their power consumption at peak times (the period when the price of power is high) ((Fadel, 2015); (Xiong, 2011)). Mainly, transmitting real-time data is compromised by energy consumption. Therefore, this thesis considers the balance between minimizing both end-to-end delay and end-to-end latency and maximizing both energy efficiency and packet throughput. To achieve this, the mostly used MOOAs are chosen to compare their results with the proposed algorithm (MOSFP) using the same environment, hardware, and platform. These algorithms are SPEA2, NSGA-II, and OMOPSO. SPEA2, NSGA-II, OMOPSO and our algorithm (MOSFP) are evaluated to find the most efficient algorithm. This is followed by Pareto-optimal set analysis to find the optimal value of packet payload size that balances between the aforementioned network models. Taking into account critical data transmission, this thesis focuses on the delay issue rather than network coverage as in prior research. Coverage is very important for applications that need deployment in the vast area of monitoring. However, in a smart grid, the scenario is different. WSN part of the smart grid is normally deployed in small areas of homes, buildings, etc. In most cases, smart ZigBee sensors can cover all the WSN part of the smart grid in which their range can reach up to 100 meters (Gungor, 2011). This is suitable for the smart grid because the average dimensions of the room in Asians smart buildings and homes will not exceed 10×10 meters (Yassein, 2016).

1.2 Problem Statement

1. Many Single Objective Optimization Algorithms (SOOAs) or (meta-heuristic methods) were proposed to solve nonlinear, complex and large-scale optimization problems that need solution in low cost with short time. Examples of these algorithms are GA and PSO along with their enhanced versions such as PGA and APSO respectively. However, optimization algorithms such as GA and PSO suffers from low solving precision, slow convergence, and bad local searching ability. In GA for example, mutation is used to increase the algorithm convergence. However, different mutation percentages lead to different solutions, thus, misunderstandings of optimality (Hassanat et al. 2016). PSO algorithm suffers from high-dimensional search space, which is easy to fall into local optimum while solving this problem (li et al. 2014). Other than that, various factors influencing the quality of final results of PSO such as learning factor C_1 and C_2 and inertia factor. Various values of these factors lead to various solutions. APSO and PGA are simplified and yet enhanced versions of both PSO and GA. These algorithms quickly explore and search the domain of any problem. However, if the number of execution iterations is too large, PGA and APSO may not always converge or reach toward the best, optimal or high-quality solutions. This is because PGA treats each individual in the population independently while APSO operates via random walks.

2. The multi-objective versions of the aforementioned algorithms suffers from slow convergence, and weak spread of solutions related to a Pareto front set. This is because they are working on the same technique of their single objective optimization versions in searching the search space domain.

3. Various WSN applications transmit critical data that should be received with a minimum delay. Examples of these applications are smart grid network, disaster monitoring network, and heart pulse and electrocardiogram monitoring network. Sensor nodes are the backbone of these applications. Sensor nodes detect the physical phenomena such as the load on the electricity power, heart pulse, or earthquake waves. Then, the network will send these data to control center. There are several challenges limiting the operational capabilities of these sensors such as limited communication range, limited memory size and storage size, and limited power in a battery (Singh, 2016). The misuse and mismanagement of these devices will reduce the Quality of Services (QoS) and network lifetime. As an instance, packet payload size plays a significant role in determining the network QoS. If packet payload size increases, the probability of dropping the packets will be increased. The retransmission of dropped packets requires reallocation in the memory and consumes more power of the battery. Consequently, the network delay will be increased.

1.3 Assumptions

(a) The proposed meta-heuristic technique called Sperm Swarm Optimization (SSO) will solve many benchmarks functions with good convergence and quality of results.

(b) Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP) will solve many benchmarks functions with good convergence and quality of results.

(c) MOSFP will help solving problems in wireless networks to achieve optimum network QoS in implementation phase in real life.

1.4 Objectives of Study

Objectives required to overcome the limitations in the previous works are stated as follows:

i. To review the most use single objective optimization algorithms in IEEE Xplore and ISI Web of Science databases. In addition, to review their extended version of multi objective optimization algorithms.

ii. To propose a Single-Objective Optimization Algorithm (SOOA) based on a metaphor of natural fertilization procedure.

iii. To appraise the SOOA with different benchmarks functions. Then, compare its results against results of four algorithms in the area of Single-Objective Optimization (SOO). These algorithms are PSO, APSO, GA, and PGA.

iv. To propose a Multi-Objective Optimization Algorithm (MOOA) as an extended version of the proposed SOOA based on the concept of non-dominated solution.

v. To evaluate the proposed MOOA with different benchmarks functions. Then, compare its results against results of three algorithms in the area of Multi-Objective Optimization (MOO). These algorithms are OMOPSO, NSGA-II, and SPEA2.

vi. To optimize a set of multi-objective problems related to wireless networks using the proposed MOOA.

1.5 Research Questions

To achieve the objectives of this work, the following questions require answers:

i. What are the advantages of using the meta-heuristic technique to solve different kinds of optimization problems?

ii. What are the disadvantages of previous types of meta-heuristic methods and their MOOAs?

iii. How to evaluate the proposed SOOA?

iv. Are there any benchmark functions allow us to evaluate the meta-heuristic methods?

v. What are the advantages of the proposed algorithm as compared to the other?

vi. What set of WSN attributes can be selected to perform multi objective optimization?

vii. How to extend the proposed algorithm to multi-objective version?

viii. How the efficiency for multi-objective version of the algorithm can be tested?

ix. What are the challenges of wireless networks?

x. How to simulate these challenges as the form of objective optimization models?

xi. How to optimize these models using the proposed multi-objective version of the algorithm?

1.6 Mapping the Objectives and Research Questions

The mapping between the objectives of the research and research questions are provided in Table 1.1 to show how the research questions are connected with the objectives.

Table 1.1: Objective Vs. research questions

Objectives	**Research questions**
1. To review the most use single objective optimization algorithms in IEEE Xplore and ISI Web of Science databases. In addition, to review their extended version of multi objective optimization algorithms.	i. What are the advantages of using the meta-heuristic technique to solve different kinds of optimization problems? **ii.** What are the main disadvantages of previous types of meta-heuristic methods and their MOOAs?
2. To propose a Single-Objective Optimization Algorithm (SOOA) based on a metaphor of natural fertilization procedure. 3. To appraise the SOOA with different benchmarks functions. Then, compare its results against results of four algorithms in the area of Single-Objective Optimization (SOO). These algorithms are PSO, APSO, GA, and PGA.	iii. How to evaluate the proposed SOOA? iv. Are there any benchmark functions allow us to evaluate the meta-heuristic methods? v. What are the advantages of the proposed algorithm as compared to the other?
4. To propose a Multi-Objective Optimization Algorithm (MOOA) as an extended version of the proposed SOOA based on the concept of non-dominated solution. 5. To evaluate the proposed MOOA with different benchmarks functions. Then, compare its results against results of three algorithms in the area of Multi-Objective	vi. How to extend the proposed algorithm to multi-objective version? vii. How the efficiency for multi-objective version of the algorithm can be tested?

Optimization (MOO). These algorithms are OMOPSO, NSGA-II, and SPEA2.	
6. To optimize a set of multi-objective problems related to wireless networks using the proposed MOOA.	viii. What are the challenges of wireless networks? ix. What set of WSN attributes can be selected to perform multi objective optimization? x. How to simulate these challenges as the form of objective optimization models? xi. How to optimize these models using the proposed multi-objective version of the algorithm?

Table 1.2 presents a list of terms that are mainly used in this thesis.

Table 1.2: A set of terms that are used in this thesis and their definition.

Term	**Definition**
Objective functions or optimization problems	Mathematical models to simulate real life problems in which, some of these models are maximization and the others are minimizations models (Andréasson, 2005).
Meta-heuristic method or Single-Objective Optimization Algorithm (SOOA)	A high-level procedure to find, generate, or select a solution to an optimization problem at each time run. Optimization problem can be classified to minimization or maximization objective function (Bianchi, 2009).
Multi-Objective Optimization Algorithm (MOOA)	An algorithm that is used to find a solution to a set of conflicting optimization problems. The solution is used to balance between minimization and maximization objective functions at each time run ((Coello, 2007) ; (Srinivas, 1994)).
Single-objective optimization	Is the process of optimizing a single minimization or maximization objective function at each time run (Rbouh, 2014).
Multi-objective optimization	Process of optimizing a set of conflicting objective functions at each time run. Some of these functions can be minimization and the others are maximizing functions ((Coello, 2007) ; (Srinivas, 1994)).
Multi-Objective Optimization Problem (MOOP)	A class of problems with solutions that can be evaluated along two or more incomparable or conflicting objectives in which some of these objectives are minimization objective functions and the others are maximization objective functions (Patil et al. 2014).
Benchmark functions	Are nonlinear mathematical functions in which some of them minimization and the others are maximizations. These functions are used to evaluate the optimization algorithms (Surjanovic, 2013).

1.7 Thesis Organization

This thesis includes seven chapters. Chapter 2 reviews the literature related to the optimization algorithms. Chapter 3 describes the overall methodology of this study.

Chapter 4 presents the proposed single-objective optimization algorithm based on sperm fertilization procedure. Chapter 5 presents the Multi-objective Optimization Algorithm based on Sperm Fertilization Procedure (MOSFP). Chapter 6 demonstrates the MOSFP algorithm in optimizing wireless sensor networks problems in smart grid applications. Chapter 7 concludes this work and highlights the future works.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter reviews Single-Objective Optimization Algorithms (SOOAs) and highlights their features and drawbacks. Figure 2.1 summarizes the percentage of SOOAs found in IEEE Xplore and ISI Web of Science databases (El-Hamrawy, 2016). From this figure, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithms are mostly used to solve complex and complicated nonlinear optimization problems (El-Hamrawy, 2016). For this reason, GA and PSO algorithms as well as their improved versions such as Accelerated PSO (APSO) algorithm (Wang, 2014), and Parallel Genetic Algorithm (PGA) (Muhlenbein, 1991) are chosen to be compared with our proposed SOOA. Additionally, this chapter reviews Multi-Objective optimization Algorithms (MOOAs) that are considered as the extended versions of the SOOAs. Example of these algorithms are Non-dominated Sorting Genetic Algorithm (NSGA-II) (Srinivas, 1994), and Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Gharari, 2016), which are extended from GA while Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) (Sierra, 2005) is extended from PSO. Finally, this chapter highlights relevant studies of MOOAs to optimize Multi-Objective Optimization Problems (MOOPs) for Wireless Sensor Network (WSN).

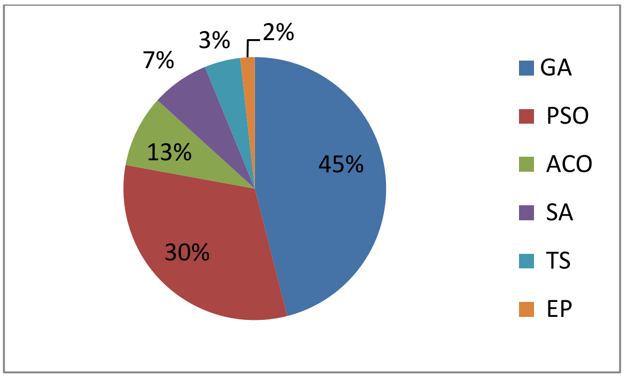


Figure 2.1: Percentage of meta-heuristic algorithms reported in IEEE Xplore and ISI Web of Science databases (El-Hamrawy, 2016).

2.2 Single Objective Optimization (SOO)

SOO comprises a single objective function in which, finding an optimal value of maximum or minimum (Savic, 2002). Single-Objective Optimization Algorithm (SOOA) or meta-heuristic algorithm is a high level procedure to generate, select or find a solution for maximization or minimization optimization problem. The quality of SOOA results can be determined by comparing the obtained results with the well-known optimal result of an optimization problem (Bianchi, 2009). An example of the well-known optimal result of an optimization problem is zero for Sphere optimization function (Surjanovic, 2013). This section reviews the most used SOOAs, then, highlights and compares their advantages and drawbacks.

2.2.1 Genetic Algorithm (GA)

Genetic algorithm is the pioneer algorithm in the area of optimization and proposed by (Holland, 1993). GA consists of adaptive techniques and procedures to find an optimal solution for different types of optimization problems by searching their search space domain (Mantas & Andrius, 2007). GA simulates a natural operation of choosing the most convenient chromosome in a wide set of population. This is to find a solution that near to the optimal solution or the exact optimal solution for a wide variety of problems (Paulinas & Ušinskas, 2007). GA can also be categorized as Biology Inspired Search (BIS) approach, which performs the procedure in a way closer in nature to the evolution theory that proposed by Charles Darwin (Goldberg & Holland, 1988). Based on a study of (Goldberg; 1992), GAs can find a solution for complex and complicated problems that many classical algorithms failed to provide a suitable solution for these problems in less effort with a short time (Goldberg, 1993).

Depending on the GA procedure steps, a primary population of individuals is arbitrarily created and every single individual goes through evaluation step. After the evaluation phase, individuals with high fitness value are reserved for the next step while the others with extremely low fitness value are discarded from the population set. The GA uses different natural operators and factors to search for the optimal solution such as crossover, natural selection, and mutation operations (Forrest & Mitchell, 1993; Meetei, 2014). The aim of the selection is to select good chromosomes in hope to produce an excellent offspring (new solutions). Mutation is an operation of any change in any gene in the chromosome that can affect the result (Langdon et al, 2013). The GA procedure is summarized in Algorithm (2.1) (Langdon, 2013).

Algorithm 2.1 Genetic Algorithm (GA)
1: Begin
2: ***Step 1:*** *define a genetic representation of the problem.*
3: ***Step 2:*** *create an initial population* $p(0) = x_1^0, \ldots . x_N^0$.
4: ***Step 3:*** *compute the average fitness* $\bar{F} = \sum_i^N F(x_i)/N$ *.Assign each individual the normalized fitness value* $F(x_i^t)/N$.
5: ***Step 4:*** *assign each* x_i *a probability* $p(x_j, t)$ *proportional to its normalized fitness. Using this distribution, select N vectors from P(t). This gives the set S(t).*
6: ***Step 5:*** *pair all of the vectors in S(t) at random forming N/2 pairs. Apply crossover with probability* p_{cross} *to each pair and other genetic operators such as mutation, forming a new population P(t+1).*
7: ***Step 6:*** *set t = t +1, return to Step 3.*
8: End Procedure.

2.2.2 Parallel Genetic Algorithm (PGA)

Parallel Genetic Algorithm (PGA) performs two enhancements to the standard genetic algorithm. First, PGA performs the mating selection based on distributed technique, which each individual in the population lives in two dimension search space and the selection of a mate is performed independently for each individual based on its neighborhood. Second, each individual can improve its fitness along with its lifetime using some procedures such as local hill-climbing, in which, each individual has the ability to apply various local hill-climbing methods. Hill-climbing is a mathematical optimization procedure that related to the family of local search. Hill-climbing starts with a random solution to a problem and then improves the solution by creating an incremental change to the previous solution (Russell, 2003). Algorithm 2.2 shows the pseudocode of Parallel Genetic Algorithm (PGA) (Muhlenbein, 1992). PGA links each individual with its neighborhood to apply the parallel search. This linkage can be done probabilistically, which constrained by the neighborhood. PGA is a distributed algorithm wherein each individual made its own decision without central control.

Algorithm 2.2 Parallel Genetic Algorithm (PGA)
1: Begin **2:** ***Step 1:*** *define a genetic representation of the problem.* **3:** ***Step 2:*** *create an initial population and its population structure.* **4:** ***Step 3:*** *each individual does local hill climbing.* **5:** ***Step 4:*** *each individual selects a partner for mating in its neighborhood.* **6:** ***Step 5:*** *an offspring is created with genetic crossover of the parents.* **7:** ***Step 6:*** *the offspring does local hill climbing. It replaces the parent, if it is better than some criterion (acceptance).* **8:** ***Step 7:*** *if not finished, return to Step 4.* **9: End Procedure.**

2.2.3 Particle Swarm Optimization (PSO)

PSO is inherently continuous algorithm proposed by Kennedy et al. (Kennedy, 1995; Kennedy, 2001). PSO stores a previous position of an individual to generate a new position and simulates the strategy of birds (particles) searching for foods. Each particle in PSO has velocity and location, which represents a possible solution. The particles explore the search space to search for the optimal solution or near to an optimal solution. The particles change their velocity and location based on their accumulated knowledge and data of exploring the search space. The location and velocity of a particle with index i are denoted by (X_i) and (V_i), respectively.

The particle index can be in the range of (i = 1,…,N) where N is the maximum number of solutions (particles) in the swarm. $X_i = (X_{i1}, X_{i2}, X_{i3}, \cdots X_{iN})$ is used to represent i^{th} solutions (particles). The current velocity of each particle is a degree obtained based on the previous velocity of each particle. $V_i = (V_{i1}, V_{i2}, V_{i3}, \cdots V_{iN})$ is used to represent the velocity of particle i. Initially, position and velocity for each particle in the dimensional search space will be assigned randomly. Then, the location and velocity of the particles are updated according to Equations 2.1 and 2.2 until the maximum number of search iteration is reached (Kennedy, 1995; Kennedy, 2001).

$$\begin{aligned} V_{i,m}^{(t+1)} &= w * v_{i,m}^{(t)} + c_1 * rand_1() * (pbest_{i,m} - x_{i,m}^{(t)}) \\ &+ c_2 * rand_2() * (gbest_m - x_{i,m}^{(t)}), \end{aligned} \tag{2.1}$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + V_{i,m}^{(t+1)}, \tag{2.2}$$

Performance of an algorithm is often varied under different conditions. (Kennedy et al., 2001) tested PSO and found that the performance and the quality of the results are varied with different population sizes. C_1 and C_2 are two learning factors, which should be equal and in the range of (0, 4) (Panigrahi, 2013). There are two random numbers called stochastic variables. Stochastic variables increase the convergence of the algorithm by multiplying the velocity of the particle with random numbers in the range of 0 and 1. These variables are stand-alone functions and used to raise the particles' velocity.

The swarm has two ideal positions, including, the ideal position that located by the particle itself and the best position that located by the neighboring particles. There are two kinds of position that located by neighboring particles. The first position is the global neighborhood, which is the position that related only to the best particle in the swarm. The second position is the local neighborhood that considered based on the location of particle neighbors itself.

Particles update their positions as described in Equation 2.2. The initial position of any particle is incremented together with the new velocity. After the position and velocity are updated, a new iteration of search is begun from the former positions of the swarm. The optimal position is finalized when the stopping criterion or a maximum number of iterations of the procedure is satisfied or reached (Ab Aziz, 2012). The inertia weight (inertia parameter) is used in PSO algorithm and represented by w (Ab Aziz, 2012). (Shi and Eberhart, 1998) discussed the issues of inertia weight and its effect in finding the global and local best values. They found that it is appropriate to begin the search via a larger inertia value to help getting more global search and local search. However, using a small value of inertia parameter will help in increasing the speed of convergence rather than exploring the whole global search space (Shi and Eberhart, 1998). There are two ways to choose appropriate values for inertia factor. The first way is a linear method as shown in Equation 2.3 (Kennedy, 2001). The linear method decreases the values for inertia factor linearly until the maximum available number of inertia is reached (Kennedy, 2001).

$$w_{i+1} = w_{\max} - \frac{w_{\max} - w_{\min}}{i_{\max}} i, \tag{2.3}$$

The second way is dynamic method in which the values for inertia factor reduces from the initial value to the determined final value fractionally by using Equation 2.4.

$$w_{i+1} = \Delta_w w_i \ , \tag{2.4}$$

Where the value of Δ_w varies between 1 and 0. Shi and Eberhart (Kennedy, 2001; Kennedy, 2004), found that dynamic method outperforms linear method in term of convergence. Algorithm 2.3 shows the pseudo-code for PSO (AbuNaser, 2015).

Algorithm 2.3 Particle Swarm Optimization (PSO)

Begin
Step 1: *initialize particles.*
Step 2: for *i=1: number of particles in P* ***do***
    Calculate fitness value fp=f(p).
    If *f_p is better than f(pbest).*
    Set p best=p.
    End if
    End for
Step 3: set gbest = choose the best value in population (P).
Step 4: for *i=1: number of particles in P* ***do***
    Calculate velocity for each particle.
    End for

13: ***Step 5:*** *Return to step 2 and repeat until reaching the maximum iteration***.**
14: End Procedure.

2.2.4 Accelerated PSO (APSO)

As mentioned earlier, PSO updates the location of each particle based on the individual best position and global best position of the whole swarm. In contrast, accelerated PSO (APSO) eliminates the individual best position and use only the global best position of the swarm to update the location of each particle. This has speed up the algorithm convergence compared to the standard PSO. The APSO updates the velocity of each particle (Gandomi, 2013) as follows:

$$v_i^{t+1} = v_i^t + \alpha r(t) + \beta r(g^* - x_i^t), \tag{2.5}$$

Where r is a random number in the range of (0, 1). The velocity of each individual in APSO is not essential and thus can be eliminated. The rule to update the location for individuals can be written in a single step to increase the algorithm convergence.

$$x_i^{t+1} = (1\text{-}\beta)x_i^t + \beta g^* + \alpha r, \tag{2.6}$$

From Equation 2.6, APSO does not contain velocity. Therefore, the initial velocity is not required to update the swarm positions on search space domain. This makes APSO simple and easy to implement. The basic term in the Equation 2.7 is the random term (r), which is used to apply the swarm mobility. The determination of r value depends on the value of α , which $\alpha = 0.1 - 0.5L$. L is the scale of each variable. β has the range of 2.0-0.7 and appropriate for most real-world problems (Gandomi, 2013).

APSO can reduce the randomness gradually by using the following formula:

$$\alpha = \delta^t, \tag{2.7}$$

Where

- δ has a value in the range of $0 < \delta < 1$ in most of the cases,

- $t \in [0, t_{\max}]$,
- t_{max} is the maximum value of generations (Gandomi, 2013).

Algorithm 2.4 summarizes the full procedure of APSO (Wang, 2014):

Algorithm 2.4 Accelerated Particle Swarm Optimization (APSO)
Begin
Step 1: initialization. *Set the position of all particles randomly.*
Step 2: *evaluate the fitness for each particle in the initial population.*
Step 3: ***while*** *(MaxGeneration)* ***do***
For *i=1:NP (all the NP particles)* ***do***
The positions of all particles are updated by Eq.(2.6).
Check x_i and limit to the allowed limits if appropriate.
End for
Increase the counter: t=t+1
Step 4: ***end while***
Step 5: *return the values of g*and f(g*) found so far.*
End Procedure.

2.2.5 GA versus PSO

The PSO algorithm performs an inherently continuous procedure in three steps to update the population until the maximum number of iterations is reached. In the first step, PSO initializes the velocity and position of the population. Second, PSO updates the velocity of each individual. Finally, the position of each individual is updated based on their velocity.

In contrast, GA performs an inherently discrete procedure wherein the past position of the individual has no effect on selecting the new position of the new individuals. GA encodes population into 1's and 0's; therefore, it easily performs discrete design variables. GA uses a set of operations to update the population namely, crossover, natural selection, and mutation operations (Sörensen, 2013). GA deals with each individual independently. However, PSO determines the new position of each particle based on the past position using the global best position and the neighborhood position to guide the search on the search space domain (Hassan, 2005). Table 2.1

summarizes the comparisons between GA and PSO (Kachitvichyanukul, 2012) while their advantages are described in Table 2.2 ((Riko & Andreja, 2013); (Bai et al., 2010)).

Table 2.1: Comparisons between GA and PSO (Kachitvichyanukul, 2012)

Comparison criteria	**GA**	**PSO**
Type of procedure	Inherently-discrete procedures	Inherently-continuous procedures
Type of a metaphor	GA is a metaphor of the Darwinian theory of evolution, which simulates the construction of chromosome and its evolution	PSO is a metaphor of a social interaction, which simulates the movement of birds flock while searching food
Solutions need ranking and selection	Solutions will be ranked through the evaluations. In addition, the selection operator will be applied to filter out the population. *Roulette wheel selection* is an example of selection operator in GA	Solutions will not be ranked through the evaluations. In addition, there is no selection operation
Influence of population size or swarm size on solution time	Exponential	Linear
Population affected by best solution	The procedure deals with each individual independently	The procedure uses the solution of swarm leader (best solution) to add it for other individual solutions
Average fitness value cannot get worse	Average fitness will not be worse because the individual will be ranked from the best to the worse. The best individuals will be reserved for next step and worst-individuals will be eliminated	Average fitness will not be worse because the velocity of the leader of the swarm (best solution) will be added to all other velocities in the swarm
Convergence	Less than PSO	More than GA

Table 2.2: Advantages of GA and PSO

Type of feature	**Advantages of GA**	**Advantages of PSO**
Ability of understanding and implementation.	GA is easy to implement and understand.	PSO is easy to implement and understand.
Area of applying	GA has the ability to solve different types of optimization problems, which can be represented by chromosome encoding.	The main idea of PSO is inspired by the intelligence, which can be applied to both engineering use and scientific research.
Type of calculations	GA does not affect by the error surface, which can solve non-parametrical, non-differential, non-dimensional, multi-dimensional and even non-continuous problems.	PSO has no mutation, crossover or overlapping calculation.
Complexity of applying	Operations of GA such as crossover, selection and mutation can be easily applied.	The calculation, rules, and complexity in PSO are very simple, which can be easily applied.
Dimension value	The dimension value is always equal to the solution, which is a	The calculation of mathematical rules in PSO based on real

	constant value.	numbers, which is decided directly by the solution. The dimension value is always equal to the solution, which is a constant value.

Drawbacks of GA and PSO can be summarized in following table:

Table 2.3: Drawbacks of GA and PSO

Drawbacks of GA	Drawbacks of PSO
In GA, mutation is used to increase the convergence. However, it influences the quality of the solution. Different mutation percentages lead to various solutions (Hassanat, 2016). This variety leads to misunderstand of the optimality.	PSO algorithm has the problem with a high and wide dimensional search space (Li, 2014), which the exploration ability of the algorithm is reduced to locate at zones that contain good solutions.
-	C_1 and C_2 are two learning factors, which should be in the range of (0, 4). This diversity of the learning factors influences the quality of the solution. Different values of these factors can lead to various solutions. Particle velocity in PSO depends mainly on values of C_1 and C_2, which if they increase, the velocity will be increased.
-	The inertia factor values play a significant role in determining the convergence of the PSO algorithm (Kennedy, 2001; Rane, 2013), which if the inertia factor is small the particle velocity will be slow.

2.3 Multi-Objective Optimization (MOO)

Multi-Objective Optimization Algorithm (MOOA) is a method to solve and balance solutions between multi-objective optimization functions (multi-mathematical models). The multi-objective optimization functions can include minimizations and maximizations. Various Single-Objective Optimization Algorithms (SOOAs) have been extended to solve Multi-Objective Optimization Problems (MOOPs) (Capitanescu, 2017). MOO is an optimization of multi objective functions (conflicting mathematical models) that often conflict, inconsistent and non-commensurable with each other. The multi objective functions can be a combination of minimization and maximization functions rather than a single function of minimization or maximization as in SOO. To solve and find the optimal solution for MOO problems, Pareto optimality has been introduced instead of using the optimality concept of SOO. The final solutions of MOOPs are selected among a set of Pareto optimal solutions (Sakawa, 2013).

Pareto optimality has inspired researchers to extend single optimization algorithms to solve MOOPs. Some of these algorithms are inspired by the metaphor of manmade procedure or natural processes ((Saka, 2016); (Watanabe, 2004)). GA and PSO are considered as important examples of these algorithms (Holland, 1992; Kennedy, 1995). Non-dominated Sorting Genetic Algorithm (NSGA) (Coello, 2007) and Non-dominated Sorting Genetic Algorithm (NSGA-II) (Srinivas, 1994) are the extended versions of GA to optimize different types of MOOPs whereas Multi-Objective Particle Swarm Optimization (MOPSO) and Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) are the extended versions of PSO algorithms to solve MOOPs (Coello, 2002; Sierra, 2005). Details of these algorithms and other well-known algorithms for MOO will be discussed in the following subsections:

2.3.1 Non-dominated Sorting Genetic Algorithm (NSGA-II):

NSGA-II is a multi-objective version of single objective GA. NSGA-II uses a set of genetic operations such as natural selection, crossover and mutation operators (Srinivas, 1994). Algorithm 2.5 summarizes the pseudo-code of NSGA-II (Coello, 2013).

Algorithm 2.5 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

Begin
Step 1: *initialize Population*
    Generate random population – size M
Step 2: *evaluate objective values*
Step 3: *assign rank (level) based on Pareto dominance-“sort”*
Step 4: *generate child population*
Step 5: *binary tournament selection*
Step 6: *recombination and mutation*
Step 7: for *i=1 to the number of generations* ***do***
        With parent and child population
        Assign rank (level) based on Pareto – “sort”
        Generate sets on non-dominated fronts
        Loop (inside) by adding solutions to next generation
        Starting from the “first” front until M individuals found
        Determine crowding distances between points on each front

Select points (elitist) on the lower front (with lower rank) and are outside a crowding distance

Create next generation

Binary tournament Selection

Recombination and mutation

Increment generation index

End for

End Procedure.

2.3.2 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

SPEA2 is one of the most important algorithms in the field of MOO (Gharari, 2016). SPEA2 was proposed by Zitzler et al. as an improvement of SPEA algorithm (Laumanns, 2001). This algorithm applies the dominance procedure and guides the search based on the nearest neighbor technique. SPEA2 utilizes the truncation method to preserve solutions on the boundary (Theophila, 2008). The pseudo-code of this algorithm is summarized in Algorithm (2.6) (Bandyopadhyay, 2013).

Algorithm 2.6 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

Begin

Step 1: *initialize population P*

Step 2: *evaluate objective functions*

Step 3: *create external archive A*

Step 4: for *i=1 to the number of generations* **do**

Compute fitness of individual in P and A

Add non-dominated individuals from P and A

If *the capacity of A is exceeded* ***Then***

Remove individuals from A by truncation operator

End If

Perform binary tournament selection to create a mating pool

Perform crossover

Perform mutation

End for

End Procedure.

2.3.3 Optimized Multi-Objective Particle Swarm Optimization (OMOPSO)

OMOPSO is a Multi-Objective Optimization Algorithm (MOOA) proposed by Sierra et al. (Sierra, 2005) as an extended approach of the PSO algorithm. This algorithm performs a set of operators to solve MOOPs, such as aggregates the global best solutions that known as leaders by using crowding operation and mutation operation. In addition, it archives the global best solutions (leaders). The OMOPSO procedure is summarized in Algorithm (2.7) (Sierra, 2005).

Algorithm 2.7 Optimized multi-Objective Particle Swarm Optimization (OMOPSO)

Begin
Step 1**:** *initialize swarm and leaders. Send leaders to* $\in$*- archive*
Step 2**:** *crowding*(leaders), $g = 0$
Step 3**:** ***while*** *g < max number of iterations (gmax)*
    For *<each particle>* ***do***
        Select leader. Flight. Mutation. Evaluation. Update pbest
        End for
        Update leaders, Send leaders to $\in$*-archive*
    Crowding(leaders), *g*++
End while
Step 4**:** *report results in* $\in$*-archive*
End Procedure.

In this chapter, all algorithms are kept as their resources without changes including the names and symbols convention ((Coello, 2013); (Bandyopadhyay, 2013); (Sierra, 2005)). Table (2.4) shows the abbreviations of previous mentioned algorithms. In Chapter 6, flow charts 6.4, 6.5, 6.6 give a detailed description about the flow of these algorithms in solving Multi-Objective Optimization Problems (MOOPs) of Wireless Sensor Networks (WSN).

Table 2.4: The abbreviations of previous mentioned algorithms

Abbreviation	Means
OMOPSO (Sierra, 2005)	
g	Iteration number
gmax	Maximum number of iterations
$\in$	Is the value of the bounding size of the $\in$−archive
NSGA-II (Coello, 2013)	
M	Is the size of random population
SPEA2 (Bandyopadhyay, 2013)	
P	Population
A	External archive

2.4 The Mutation and Crossover of Algorithms

The aforementioned algorithms use a set of operations to converge faster toward an optimal solution namely, mutation and crossover operations. These operations are used to produce a new solution that is better than the previous solution. Both of mutation and crossover operations are used in GA and its MOO versions, including, SPEA2 and NSGA-II while OMOPSO uses only mutation operation. In this section, we are going to review these operators based on the JMetal tool (JMetal, 2018). JMetal tool is one of the most popular tools in the area of optimizations and contains different types of multi-objective and single-objective optimization algorithms.

Crossover operator is a genetic operator to produce new chromosome (new variable or new solution) from the old one in hope reaching the optimal result by the end of the procedure. SPEA2 and NSGA-II use Simulated Binary Crossover (SBX) (Kumar, 1995). The SBX of the variable (*X*) or chromosome (*X*) can be calculated by:

$$\begin{cases} Y_1 = 0.5[(1-\beta)X_1 + (1+\beta)X_2, \\ Y_2 = 0.5[(1+\beta)X_1 + (1-\beta)X_2, \end{cases} \tag{2.8}$$

Where X is the value of variable or chromosome, Y is the value of variable or chromosome after the crossover (new solution) while β is a random variable in a range of 0 and 1. The probability distribution of variable β can be calculated by:

$$\begin{cases} P(\beta) = 0.5(\eta_c + 1)\beta^{\eta_c}, 0 \le \beta \le 1, \\ P(\beta) = 0.5(\eta_c + 1)\dfrac{1}{\beta^{\eta_c+2}}, \beta > 1, \end{cases} \tag{2.9}$$

Where η_c is the variable of distribution index.

Mutation operator is any changes occurred in gene of chromosome or variable to produce a better value. SPEA2 and NSGA-II use a polynomial mutation, while OMOPSO uses different kinds of mutation such as uniform or non-uniform mutation. For the proposed algorithm, uniform and non-uniform mutation are used to improve the algorithm convergence.

a) Polynomial mutation was proposed by Deb et al. (Deb, 2016) and can be calculated as follows:

$$p' = \begin{cases} p + \delta_L (p - x_{i,j}{}^{(L)}), \ for: u \le 0.5, \\ p + \delta_R (x_{i,j}{}^{(U)} - p), \ for: u > 0.5, \end{cases} \tag{2.10}$$

Where p represents the parent solution $p \in [x^{(U)}, x^{(L)}]$, $x^{(U)}$ is the upper bound value, $x^{(L)}$ is the lower bound value of the variable while the (u) is a random variable in the range of 0 and 1. The two parameters δ_L and δ_R are calculated as follows (Deb, 2016):

$$\begin{cases} \delta_L = (2u)^{1/(1+n_m)} - 1, \ for: u \le 0.5, \\ \delta_R = 1 - (2(1-u))1/(1+n_m), \ for: u > 0.5, \end{cases} \tag{2.11}$$

The parent point p = 3.0 in the bounded range of 1 and 8 with n_m = 20.

b) Uniform mutation of value x can be calculated (Shi, 2012) as follows:

$$x_{i,j} = x_{i,j}^{(L)} + u(x_{i,j}^{(U)} - x_{i,j}^{(L)}), \tag{2.12}$$

Where $x_{i,j}$ is the position of the variable on search space domain, $x^{(U)}$ is the upper bound value, $x^{(L)}$ is the lower bound value of the variable, and (u) is a random variable in the range of 0 and 1.

c) Non-uniform mutation of value $x_{i,j}$ can be calculated as follows (Zhao, 2007):

$$x'_{i,j} = \begin{cases} x_{i,j} + \Delta(t, x^{(U)} - x_{i,j}), if: u = 0, \\ x_{i,j} + \Delta(t, x_{i,j} - x^{(L)}), if: u = 1, \end{cases} \tag{2.13}$$

Where $x^{(U)}$ is the upper bound value, $x^{(L)}$ is the lower bound value of the variable while (u) is a random variable in the range of 0 and 1. The function $\Delta(t, y)$ can be calculated by (Zhao, 2007):

$$\Delta(t, y) = y \cdot (1 - u^{(1-\frac{t}{T})^d}), \tag{2.14}$$

Where y is a variable with two cases; case 1 is the ($x_{i,j}$- $x^{(L)}$); case 2 is the ($x^{(U)} - x_{i,j}$). (u) is a random variable in the range of 0 and 1. T is the maximum number of generations while d is a system parameter determining the degree of dependency on the iteration number.

2.5 Overview on WSN Challenges and Their Multi-Objective Optimization Problems

In last decade, demands for comfortable and portable mobile connectivity has emerged to facilitate access of various applications from anywhere at any-time. Therefore, many wireless communication and monitoring technologies have been developed to meet these requirements. As a result, wireless technologies such as IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), and 802.15.4 (Zigbee) ((Lopez-Aguilera, 2017); (Chang, 2014); (Teng, 2016)) are used in greenhouses, homes and hospitals. These technologies operate on 2.4-GHz free license band called Industrial, Scientific and Medical band (ISM band) (AlQahtani, 2017) and can be attached at different scales that can reach up to kilometers. Other than that, there are wireless telecommunication networks that use a set of wide bands of electromagnetic frequencies to send data from source to destination and can reach far places to serve a huge number of customers. Examples of these frequencies are Extremely High Frequency (EHF), Very High Frequency (VHF), and Ultra High Frequency (UHF) (Held et al, 2007). Recently, wireless sensors are used in many fields like tracking, controlling, automation and monitoring. These sensors are linked together to form Wireless Sensor Network (WSN) (Hamdan, 2015). WSN is a promising tool but faces many challenges as summarized below:

(1) Network scalability: WSN operates based on a widespread deployment of sensor nodes to cover the largest possible area for monitoring. This feature affecting the whole system and can be sensitive to failure (Gutiérrez, 2014). Threfore, the network coverage should be evaluated to mitigate this challenge and ensure a high Quality of Service (QoS).

(2) Energy management: This issue is considered as the biggest limitation in any WSN as sensor operates on limited power resources such as batteries. WSN are subject to

failure because of depletion of power resources (batteries) (Yassein, 2017). Sensors are developed to operate autonomously for prolonged periods of time in years or months after deployment mission. It is not an easy task to replace or recharge the nodes batteries (Jawad, 2017). Therefore, issues affecting the energy management should be examined to minimize the energy consumption of the nodes. This can be achieved by evaluating and examining issues in protocol layer and physical layer of the network.

(3) Limited storage and memory: Sensor nodes have a limited storage with a capacity in the range of 32 KB to 2 GB and the memory with a capacity in the range of 2 KB to 256 KB (Pinto, 2015). This limitation affects the network throughput and performance of the network, which in case of transmitting a big size data, it will take a long time and consume more power. The available sensor nodes along with their memory and storage characteristics are summarized in Table 2.5 (Singh, 2016).

Table 2.5: Available sensor nodes along with their memory and storage characteristics (Singh, 2016)

Platform	MCU	RAM	Program and data memory	Radio chip
BTnode3	ATMega128	64KB	128-180 KB	CC1000/Bluth
Cricket	ATMega128	4KB	128-512 KB	CC1000
Imote2	Intel PXA271	256 KB	32-MB	CC2420
MICA12	ATMega128	4KB	128-512 KB	CC1000
MICAZ	ATMega128	4KB	128-512 KB	CC2420
Shimmer	TI MSP 430	10 KB	48KB-UP to 2 GB	CC2420/Bluth
TelosA	TI MSP 430	2KB	60-512 KB	CC2420
TelosB	TI MSP 430	10 KB	48 KB -1 MB	CC2420
XYZ	ARM 7	32 KB	256 KB	CC2420

(4) Delay of data aggregation: This challenge is critical in applications when the data must be received without delay or with minimum delay. Therefore, it is crucial to minimize delay in these applications to get better QoS of the network (Yan, 2016).

Examples of these applications are disaster monitoring network (Chen, 2013), electrocardiogram and heart pulse monitoring network (Rout, 2017), power supply and power bill requests in smart grid (Devidas, 2016).

(5) Fading and interference: WSN most communicates using license-free bands (ISM band) (Li, 2017). However, various devices are also used the same frequency band in their operation such as microwave oven and Wi-Fi routers. This increases the possibility of fading and interference ((Guo, 2012); (Iturri, 2012)). Therefore, the network planner should design the network in ways that diminish these intrusions.

To solve these challenges, they can be simulated as mathematical network models (objective functions). Consequently, a higher Quality of Services (QoS) of the network can be achieved during the implementation phase. These network models also include the effect of different network parameters such as frequency range, packet payload size, and the distance between transmitter and receiver. However, some of these objective functions are conflicting and complex nonlinear mathematical problems. These problems are difficult for human brain to analyze them in a short time. Therefore, different types of Multi-Objective Optimization Algorithms (MOOAs) are used to obtain the optimal results of a wide variety of Multi-Objective Optimization Problems (MOOPs) in WSN. Studies about solving MOOPs in WSN are summarized as follows:

Energy-efficient Coverage Control Algorithm (ECCA) proposed by Jia et al. (Jia, 2009) works based on the NSGA-II multi-objective algorithm. ECCA optimizes two conflicting network models such as minimizing the network energy consumption and maximizing the network coverage. The authors conducted two experiments to measure the performance of ECCA. In the first experiment, a total number of 100 sensor nodes was used to cover a topology area of 100×100 meters. In the second experiment, a total number of 200 sensor nodes was used to cover topology area of 200×200 meters. The previous tests have been conducted by changing the number of generations from 10 to

200 generations. The results shows that the algorithm is efficient in providing the optimal coverage with less energy consumption.

Details of the coverage problem have been discussed in (Kukunuru, 2010). Kukunuru et al. used PSO algorithm to maximize the network coverage based on the distance between nodes. They optimized the previous model using two scenarios. First, the number of nodes was changed from 0 to 80 in the topology area of 50×50 meters. Second, the number of nodes was changed from 0 to 40 in the same topology area of 50×50 meters. From these tests, 40 sensor nodes was found to be optimal to cover the area of 50×50 meters. However, the execution time of the optimization task are not measured and studied in these tests. For this reasons, WSN models that affect the Quality of Services (QoS) have been discussed by Yang et al. (Yang, 2014).

Yang et al. (Yang, 2014) proposed to minimize task execution time and maximize the network lifetime. They proposed a modified version of Binary Particle Swarm Optimization (MBPSO) and compared it with GA and Binary Particle Swarm Optimization (BPSO). The network models were tested by changing the number of nodes from 0 to 60 and the number of execution tasks for each node from 0 to 10 in topology area of 500×500 meters. Their results showed that MBPSO algorithm outperformed other algorithms in terms of minimizing the task execution time and maximizing the network lifetime. However, this study is lacked in terms of network end-to-end delay and energy consumption. The distance between nodes can increase the network delay and this can increase the dropped packets in the network. The retransmission of these packets will consume more power.

Sagar et al. (Sagar, 2014) discussed the coverage issue of sensor nodes deployment in a wide area. Coverage is used to determine the optimal number of nodes to cover the topology area. They used two algorithms to maximize the node coverage and minimize the energy consumption namely, NSGA-II and Optimal Geography Density Control (OGDC). Two tests were conducted to find the optimal coverage in a topology area of 100×100. The maximum iteration of the algorithms was set to 250. The population size and crossover rate were 100 and 0.9 respectively. The Pareto-front carves were illustrated by changing the number of nodes from 0 to 400. The results showed that NSGA-II outperformed OGDC and used only 210 nodes to cover the topology area while OGDC used 327 nodes to cover the same topology area.

Chaudhuri et al. (Chaudhuri, 2010) discussed a Coverage and Lifetime Optimization (CLOP) problem of WSN. This problem aims to optimize two models, including, minimizing the network energy consumption and maximizing the coverage of nodes. Two algorithms were chosen to optimize these models namely, SPEA2 and NSGA-II. The experiment was repeated 10 times to ensure the quality of the results. The population size and the numbers of evaluations were changed from 300 to 5000 and from 50000 to 500000 respectively. The numbers of sensor nodes were changed from 5 to 20 nodes. The results showed that NSGA-II outperformed SPEA2 in solving the CLOP problem.

In a later study, Sengupta et al., (Sengupta, 2012) proposed controlling the nodes density based on scheduling algorithm to achieve the maximum node coverage and network lifetime. This algorithm is used to schedule the active nodes. If any failure occurs, the optimization algorithms will rearrange the network unless all nodes have lost their connectivity or energy. Sengupta et al. used two algorithms to obtain the minimum energy consumption and maximum coverage. The first algorithm is Non-dominated Sorting Genetic Algorithm (NSGA-II). The second algorithm is Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D), which is a genetic algorithm framework that decomposes a multi-objective to a set of single objective problems. The results showed that MOEA/D outperformed NSGA-II in optimizing a node density control problem. However, node selection problem of WSN is not discussed in this study. For this reason, the node selection problem of WSN has been discussed by Naeem et al. (Naeem, 2012). The aim of node selection is to reduce the energy consumption of the network by selecting a set of nodes to work in the network rather than utilizing all the nodes in the network. Binary Particle Swarm Optimization (BPSO), Genetic Algorithm (GA), Convex Optimization Algorithms (COA), and PSO-Cyclic Shift Population (CSP) were chosen in their study to solve the problem. The results showed that BPSO outperformed other algorithms in finding the optimal number of selected sensors. However, this study is lacked in evaluating other models that affecting the energy consumption model such as network coverage, which if the coverage increased, the energy consumption will be decreased.

An enhanced Multi-Objective Particle Swarm Optimization (MOPSO) proposed by Liu et al. (Liu, 2016) used archive method and crowding factor. The proposed version of MOPSO was tested in optimizing two conflicting models of WSN such as minimizing the energy consumption and maximizing the node coverage. The modified MOPSO was compared with the original version of MOPSO. The number of sensor nodes is set to 40 nodes to cover a topology area of 20×20 meters. The number of iterations and the number of particles are set to 300 and 30 respectively. The experimental results showed that the improved version of MOPSO outperformed MOPSO in terms of minimizing the network energy consumption and maximizing the network coverage.

Bara'a et al. (Bara'a, 2015) proposed a multi-objective optimization modeling of the WSN. Two algorithms were chosen such as MOEA/D and NSGA-II to minimize the energy consumption for each node by maximizing the network coverage. This to ensure an efficient routing to the sink node. The number of sensor nodes was set to 25 to cover the topology size of 100×100 meters. The parameters and settings of the algorithms were population size equal to 50, the crossover probability equal to 0.6, and mutation probability equal to 0.03. The results showed that NSGA-II outperformed MOEA/D in both maximizing the node coverage and minimizing the energy consumption. This study is lacking of examining the models that are affected by the interference resources.

(Hu et al. 2008) have proposed a multi-objective optimization of urban traffic sensor networks using PSO. This system consists of wireless cameras deployed in Intelligent Transportation Systems (ITS) such as ring roads and main arteries. These cameras monitor the road traffic in milliseconds. Hu et al. optimized the node coverage to reach the optimal transmission radius. They also focused on maximizing the network throughput and minimizing the energy consumption in each node based on maximizing nodes transmission radius. They conducted different tests by varying the numbers of nodes in each test from 100 to 800 nodes.

(Chaudhary et al. 2012) conducted a further discussion about the coverage problem in WSN. They have used MOPSO algorithm to minimize the energy consumption based on maximizing the network coverage. The simulation settings were the number of nodes equal to 10 and the topology size equal to 10×10 meters. The algorithm settings were the number of particles equal to 50, and the number of generations equal to 10. The results have been illustrated by the Pareto-front curve. The Pareto-front showed that the energy consumption decrease if the coverage ratio increased. This study is lacked in terms of the coverage problem in a cluster based WSN.

In a different study, (Jia et al. 2009) discussed a set of objective functions that affect the QoS of cluster-based networks such as the end-to-end delay, the network coverage, and the energy consumption. They used NSGA-II to maximize the node coverage and minimize the end-to-end delay by keeping a few numbers of active nodes in the cluster. The algorithm settings were the number of generations equal to 50, 200, and 500, mutation percentages equal to 0.9 and 1. The simulation settings were the topology area equal to 100×100 meters and the number of nodes equal to 1000 nodes. The experimental results showed that NSGA-II is able to solve this problem on the different numbers of generations. This study is lacking of examining the objective functions (models) that are affected by the interference resources.

For this reason, Hamdan et al. (Hamdan, 2017) discussed the interference challenges in 2.4 GHz ZigBee wireless sensor networks. There are many devices operate in the same frequency band, which make an interference of ZigBee network. Wi-Fi router and microwave oven are important examples of these devices that operate in 2.4 GHz. Hamdan et al. have chosen three models such as OMOPSO, NSGA-II, and SPEA2 to maximize the packet throughput and the energy efficiency and also minimize the interference. These models were evaluated by changing the distances between both receiver node and interference source, and also between receiver and transmitter. The results showed that the NSGA-II outperformed both OMOPSO and SPEA2 in optimizing the previously mentioned models.

Table 2.6: Comparison based on the previous works in optimizing multi-objectives problems of wireless sensor network

Authors	Algorithms	Models	Study/ Findings	Limitations
(Jia, 2009)	ECCA	Maximize network coverage & minimize network energy consumption	Tested by changing the number of sensor nodes and the topology sizes.	Model of end-to-end latency is not evaluated.
(Kukunuru, 2010)	PSO	Maximize network coverage	The best coverage for the area of 50×50 is when the number of nodes is equal to 40 nodes.	The end-to-end delay and energy consumption are not evaluated.
(Yang, 2014)	MBPSO, BPSO, GA	Maximize network lifetime & minimize task execution time	MBPSO outperformed the other algorithms in term of optimizing the proposed models.	The models of network coverage and throughput are not evaluated.
(Sagar, 2014)	OGDC, NSGA-II	Maximize network coverage & minimize network energy consumption	NSGA-II outperformed OGDC, which used 210 nodes to cover the topology while OGDC requires 327 nodes to cover the same topology area.	Other models such as network throughput are not discussed.

(Chaudhuri, 2010)	NSGA-II, SPEA2	Maximize network coverage & minimize network energy consumption	NSGA-II outperformed SPEA2.	End-to-end latency model and end-to-end delay model are not proposed.
(Sengupta, 2012)	MOEA/D, NSGA-II	Maximize network coverage & minimize network energy consumption	MOEA/D outperformed NSGA-II in finding the optimal results of the proposed objective functions.	The node selection problem of WSN is not discussed.
(Naeem, 2012)	GA, BPSO, CSP	Node selection problem to achieve minimum energy consumption	BPSO outperformed other algorithms in finding the optimal number of selected sensors.	Effect of node selection problem on network delay is not studied.
(Liu, 2016)	Improved version of MOPSO, MOPSO	Maximize network coverage & minimize network energy consumption	The improved MOPSO outperformed the original MOPSO in maximizing the network coverage and minimizing the network energy consumption.	Models that are affected by the interference resources are not examined.
(Bara'a, 2015)	NSGA-II, MOEA/D	Maximize network coverage & minimize energy consumption	NSGA-II outperformed MOEA/D in minimizing the energy consumption and maximizing the node coverage.	Optimizing end-to-end delay model is not discussed.
(Hu et al. 2008)	PSO	Maximize network throughput, nodes transmission radius (coverage) & minimize energy consumption	The results showed the models based on changing the number of nodes from 100 to 800 nodes.	End-to-end delay is not evaluated.
(Chaudhary et al. 2012)	MOPSO	Minimize the energy consumption & maximize network coverage	The Pareto-front showed that the energy consumption decreases if the coverage ratio increased.	Other algorithms are not used to test the models, which different algorithms help to confirm the final result.
(Jia et al. 2009)	NSGA-II	Maximize node coverage & maximize end-to-end delay	The experimental results showed the previous mentioned models by varying the number of active nodes in the topology area.	Network throughput is not evaluated.
(Hamdan, 2017)	NSGA-II, OMOPSO, SPEA2	Maximize packet throughput, energy efficiency & minimize interference	NSGA-II outperformed both SPEA2 and OMOPSO in optimizing the proposed models.	Network end-to-end delay is not evaluated.

The prior work studies different challenges of WSN. Some of them developed the enhanced versions of previous optimization algorithms and tested in optimizing problems related to WSN while the others used the existing optimization methods to solve a set of models that affect network QoS. Based on Table 2.6, we can summarize that the end-to-end latency, end-to-end delay and network throughput have not yet been evaluated in prior studies. These objective functions are very important in estimating the QoS of any WSN. If the network end-to-end delay increased, the dropped packets will be increased and the retransmission of these packets will consume more energy and time. Therefore, we will fill the gap of prior studies by optimizing the multi objective problem of minimizing both of network end-to-end delay and end to end latency and maximizing both of energy efficiency and network throughput using our propose algorithm and other well–known MOOAs. This will be further discussed in Chapter 6.

2.6 Chapter Summary

Chapter two can be summarized as follows;

1- Optimization algorithms (meta-heuristic algorithms) are used to solve different types of complex and nonlinear optimization problems.

2- There are many single objective optimization algorithms have been proposed to solve nonlinear optimization problems. Examples of these algorithms are GA, PGA, PSO, and APSO. Some of these algorithms are extended to multi-objective versions using dominance concept to manage the trade-offs between a set of objective functions. Examples of these algorithms are NSGA-II and SPEA2, which are the multi-objective versions of GA while OMOPSO is the multi-objective version of PSO.

3- However, these algorithms suffer from low solving precision, slow convergence, and bad local searching ability.

4- Most of the prior works for optimizing Multi-Objective Optimization Problems (MOOPs) related to Wireless Sensor Network (WSN) did not give much attention to network delay. Delay is crucial for many critical applications of WSN. These applications are used to send a critical data to control center such as, power load management requests, heart pulses, and earthquake alarm. These data should be received with minimum delay.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter describes the scheme of this work that based on sequential approaches to achieve the outlined objectives as illustrated in Figure 3.1. The first part of this scheme is to develop a Single Objective Optimization Algorithm (SOOA) to solve a single objective function either maximization or minimization function at a time. However, the real-life problems are often involved conflicting optimization problems (Multi-Objective Optimization Problems (MOOPs)) that consist of both minimization objective functions and maximization objective functions at the same time. Therefore, in the second part of this scheme, Multi-Objective Optimization Algorithm (MOOA) is developed as enhanced version of the proposed Single-Objective Optimization Algorithm (SOOA). The proposed MOOA manages the tradeoffs between a set of conflicting objective functions (minimization and maximization functions) at the same time. In the final part, the proposed MOOA is demonstrated to solve the conflicting optimization problems in WSN. Each part will further explain in Chapter 4, Chapter 5 and Chapter 6 respectively.

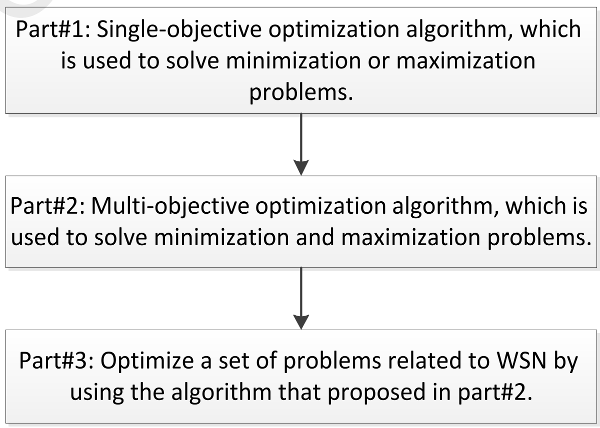


Figure 3.1: The scheme of sequential workflow

This chapter is organized as follows: Section 3.2 summarizes the first part of the scheme: single objective optimization algorithm to solve a single objective of maximization or minimization objective function at a time. Section 3.3 presents the second part of the scheme: multi-objective optimization algorithm to solve a multi objective functions at each time run. These objective functions are conflicting that involved both maximization and minimization objective functions. Section 3.4 presents the third and final part of the scheme: optimizing a set of multi-objective problems in WSN using the proposed multi-objective optimization algorithm. The multi-objective problems in WSN consist of conflicting problems of maximization and minimization objective functions at the same time. The mapping between the objectives of the research and research methodology is summarized in section 3.5 while Section 3.6 summarizes this chapter.

3.2 Part One: Single Objective Optimization Algorithm (SOOA)

The first part of this work contains the following activities and steps:

1. Study on the previous SOOA. Based on Figure 2.1, Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and their enhanced versions such as Parallel Genetic Algorithm (PGA) and Accelerated Particle Swarm Optimization (APSO) are chosen in this study.
2. Identify the limitations of the selected algorithms. Based on these limitations, comparisons between these algorithms with the proposed algorithm can be conducted. Examples of these limitations are quality of results and algorithm convergence. The quality of result is referred to the ability of the algorithm to find the optimal result of a problem. Quality of result can be determined by comparing the obtained result of SOOA with the well-known optimal result of an optimization problem (Bianchi, 2009). An example of the well-known optimal result of an optimization problem is the optimal result of Sphere optimization function, which is zero (Surjanovic, 2013). The algorithm

convergence is referred to the ability of the algorithm to converge toward optimal result with a determined number of generations ((Ping, 2013), (Pant, 2008)). For example, in finding the optimal value of Sphere function which is 0.00, algorithm *x* performed 100 generations while algorithm *y* performed 500 generations. Between these two algorithms, algorithm *x* has a better and faster convergence to reach the optimal value of Sphere function compared to algorithm *y*.

3. Propose a SOOA based on a metaphor of natural fertilization procedure. The proposed algorithm is inspired by sperm motility to fertilize the egg. The proposed algorithm called Sperm Swarm Algorithm (SSO) that simulates the forward moves of sperm swarm from a low-temperature zone called Cervix. During this direction, sperm searches for a high-temperature zone called Fallopian Tubes where the egg is waiting for the swarm for fertilization at this zone. This area is considered as the optimal solution.
4. The evaluation of the proposed algorithm is performed as follows:
    a. Identify a set of non-linear benchmark functions such as Sphere, Rosenbrock, Rastrigin, EggCrate, and Sum Square function (Surjanovic, 2013). Benchmark functions are non-linear mathematical functions in which some of them minimization and the others are maximizations. These functions are used to evaluate the meta-heuristic algorithms (optimization algorithms) (Surjanovic, 2013). Examples of the most used benchmark functions and their characteristics are given in Section 4.3.
    b. These benchmark functions are used to compare the proposed algorithm with the existing algorithms by varying the number of generations and population size.

c. Two metrics namely, quality of results and algorithm convergence are used to compare the proposed SOOA with other algorithms. The quality of result and convergence of SOOAs are described in Section 4.3, Figure from 4.11 to Figure 4.23.

The workflow of the single objective optimization algorithm is summarized as in Figure 3.2.

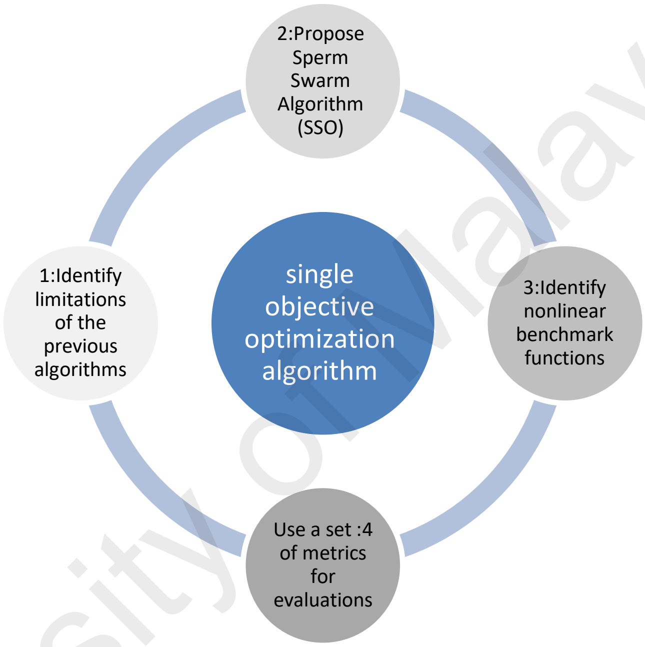


Figure 3.2: Workflow of the single objective optimization algorithm

3.3 Part Two: Multi-objective Optimization Algorithm

Multi-Objective Optimization (MOO) is an optimization of conflicting objectives involving minimization and maximization objective functions. The proposed SSO algorithm is extended to multi-objective version with the following activities and steps:

1. Propose a MOOA based on sperm fertilization procedure. Different concepts are defined such as Pareto dominance, crowding factor, archive operator, and mutation operators. Pareto dominance finds the non-dominated solution of a Multi-Objective Optimization Problem (MOOP). These non-dominated solutions are used to balance between the maximization and minimization

objective functions (conflicting objectives). The MOOA is used to find these non-dominated solutions and crowds them to archive them in the memory each time run. The full description of Pareto dominance, crowding and archiving operations are discussed in Section 5.3. Mutation is a mathematical operation that changes the values of these non-dominated solutions based on a set of rules to generate new and better values. Algorithm 5.1 describes the mutation procedure used in our proposed MOOA. Section 2.4 reviews the available mutation operations and their mathematical descriptions.

2. Identify a set of MOOAs to compare with our algorithm. As in Chapter 2, NSGA-II, SPEA2, and OMOPSO are chosen for this purpose.
3. The evaluation of MOOA is performed as follows:
   a. Identify a set of benchmark functions to use for comparing MOOAs. These functions are Zitzler-Deb-Thiele (ZDT) test suite, and Walking-Fish-Group (WFG) test suite (Huband, 2006). Benchmark functions (test suites) are conflicting non-linear mathematical objective functions to evaluate the MOOAs (Huband, 2006). Detailed mathematical formulation of Zitzler-Deb-Thiele (ZDT) and Walking-Fish-Group (WFG) test suites are provided in Part 1 and Part 2 of Appendix A. The features of these test suites are described in Table 5.1 and Table 5.2.
   b. Identify notions of comparisons to determine the types of comparison metrics. Examples of these notions are:
      i. Minimize the distance between the Pareto front that generated by the proposed algorithm and the global Pareto front of any problem. Global Pareto front is the well-known Pareto front of any problem. Examples of well-known Pareto fronts are the Pareto fronts of ZDT test suites.

ii. Maximize the spread of the solution that generated by the proposed algorithm so that the uniform distribution of vectors can be obtained.

iii. Maximize the convergence of the proposed algorithm to achieve a good quality of the Pareto optimal set.

c. Use a set of metrics that answer notions. Examples of these metrics are Inverted Generational Distance (IGD), Spread (SP), and Epsilon ($\in$) ((Patnaik & Mandavilli, 1996); (Monroy, 2004); (Riquelme, 2015)). These metrics are discussed in detail in Section 5.4.

d. Compare the proposed algorithm with the existing algorithms using qualitative and quantitative techniques:

(i) For qualitative technique: The quality of the Pareto front for each benchmark function is compared between the proposed and existing algorithms. The Pareto front of Zitzler-Deb-Thiele (ZDT) and Walking-Fish-Group (WFG) test suites are drawn for each algorithm. Comparisons between the proposed MOOA and the other MOOAs based on their obtained Pareto fronts of Zitzler-Deb-Thiele (ZDT) and Walking-Fish-Group (WFG) test suites are given in Figure 5.3 to Figure 5.13.

(ii) For quantitative technique: Median, average, best, and worst of each metric for each benchmark function are compared between the proposed and existing algorithms. These metrics are Inverted Generational Distance (IGD), Spread (SP), and Epsilon ($\in$) in which, calculated for Zitzler-Deb-Thiele (ZDT) and Walking-Fish-Group (WFG) test suites. Table 5.4 and Table 5.5 present these metrics for each test suite.

The workflow of the multi-objective optimization algorithm is summarized as in Figure 3.3.

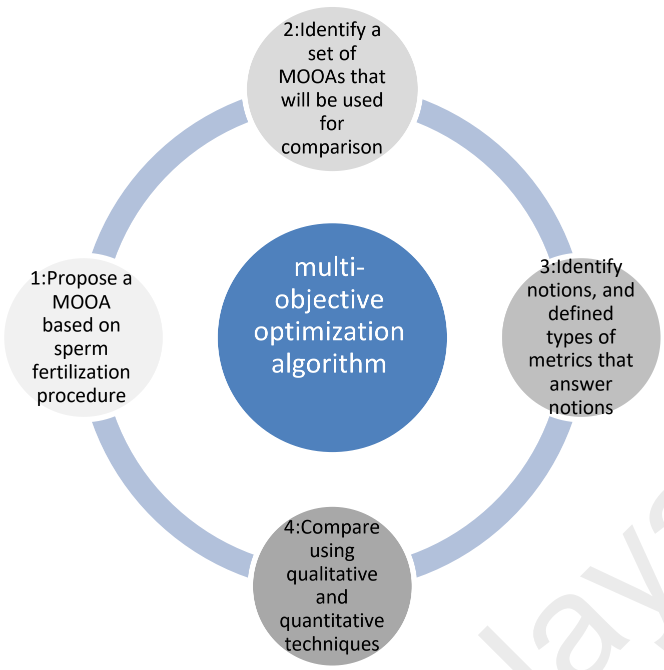


Figure 3.3: Workflow of the multi-objective optimization algorithm

3.4 Part Three: Optimize a Set of Multi-objective Problems Related to Wireless Networks Using the Proposed Multi-objective Optimization Algorithm

Challenges or problems in WSN require an accurate solution at low cost in short time. Examples of these challenges are energy consumption, delay of the network, and network latency. Therefore, SSO algorithm is extended to the Multi-Objective Optimization Algorithm (MOOA) to solve different types of conflicting non-nonlinear problems. A multi-objective version of SSO is applied to solve these problems with the following activities and steps:

1- Identify problems in WSN. These problems are represented as conflicting mathematical models (maximization and minimization objective functions).
2- Identify a set of multi-objective algorithms to confirm the results along with a multi-objective version of SSO algorithm.
3- Optimize the defined models using the proposed algorithms.

4- Use a set of criteria to evaluate the results of the algorithms. The evaluation is performed as follows:

   a. Compare the results of the proposed algorithm and other algorithms in terms of median, average, best, and worst for each objective function.

   b. Draw Pareto-front of each algorithm for each objective function. Analyze these fronts using the following points:

      i. Find a knee point of each Pareto front for each objective model. Knee point is a point on the Pareto front curve and the most preferred solution. Knee point is obtained by determining the greatest reflex angle that bends of the front from its right to its left or vice-versa (Deb & Gupta, 2011). Detail description about the knee point and its calculation can be found in Section 6.5.2 and Figure 6.9.

      ii. Defined Marginal concept of optimality to represent the optimal point of a set of conflicting objective functions (minimization models and maximization models). The optimal value based on this concept can be illustrated by the intersection points between these conflicting objective functions ((Bortolotti, 1999); (Massiani, 2013)). Section 6.5.2 and Figure 6.8 describe the detail of the Marginal concept of optimality.

      iii. Finally, the algorithms are compared based on these concepts. Figure 6.10 to Figure 6.14 present these comparisons based on the aforementioned concepts.

The workflow of optimizing a set of WSN problems is summarized as in Figure 3.4.

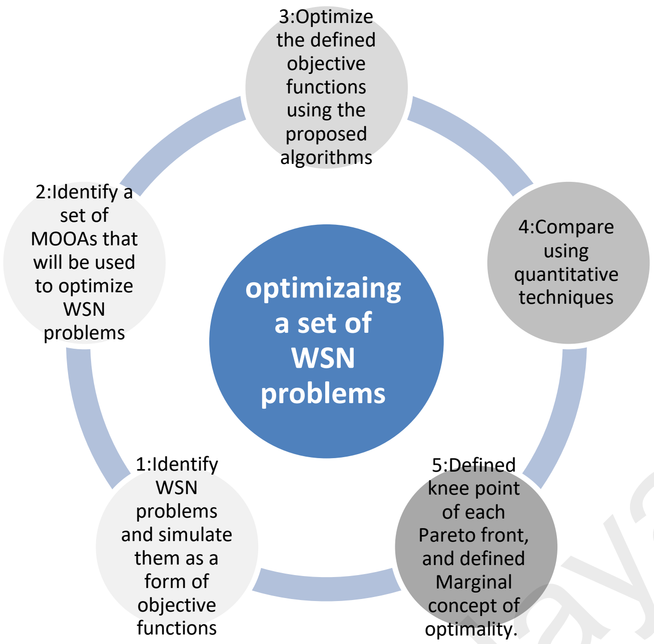


Figure 3.4: Workflow of optimizing a set of WSN problems

3.5 Mapping between the objectives of the research and research methodology

The mapping between the objectives of the research and research methodology are provided in Table 3.1.

Table 3.1: Mapping between the objectives of the research and research methodology

Objectives	Methodology	Technique and Material
i. To review the most use single objective optimization algorithms in IEEE Xplore and ISI Web of Science databases. In addition, to review their extended version of multi objective optimization algorithms.	A thematic taxonomy along with drawbacks and limitations of prior optimization algorithms have been devised. The description of their operation and procedures are reviewed. This objective is achieved in Chapter 2.	-
ii. To propose a Single-Objective Optimization Algorithm (SOOA) based on a metaphor of natural fertilization procedure. iii. To appraise the SOOA with different benchmarks functions. Then, compare its results against results of four algorithms in the area of Single-Objective Optimization (SOO). These algorithms are PSO, APSO, GA, and PGA.	A meta-heuristic optimization algorithm, called "Sperm Swarm Optimization (SSO)" is introduced. The algorithm is inspired by the sperm motility to fertilize the egg wherein the sperm swarm moves forward from a low-temperature zone called Cervix. During this direction, sperm searches for a high-temperature zone called Fallopian Tubes where the egg is waiting for the swarm for fertilization at this zone. This area is considered as the optimal solution. Objectives 1 and 2 are achieved in Chapter 3	MatLab version 7.0.4. is used to implement the proposed algorithm. SSO is tested with several benchmark functions used in the area of optimization. The results from GA, PGA, PSO, and APSO are used to compare with the results from our proposed algorithm.

iv. To propose a Multi-Objective Optimization Algorithm (MOOA) as an extended version of the proposed SOOA based on the concept of non-dominated solution. v. To evaluate the proposed MOOA with different benchmarks functions. Then, compare its results against results of three algorithms in the area of Multi-Objective Optimization (MOO). These algorithms are OMOPSO, NSGA-II, and SPEA2.	For this objective, SSO algorithm that proposed in objective 1 has been extended to a multi-objective optimization algorithm. The multi-objective version of the algorithm operates based on Pareto dominance and crowding factor that crowd and filter out the list of the best sperms (global best values). The swarm has been divided into three equal parts by taking modulus for index of each sperm to the number of three. The uniform mutation has been applied on the first part of swarm, non-uniform mutation has been applied on the second part, and the third part of the swarm has not any type of mutation. This helps to make the diversity of results, which the mutation helps to increase the convergence of algorithm to obtain good results. In case the previous types of mutations do not help the algorithm for converging to good results, the third part of the swarm (sperm without mutation) will reserve on good results. For evaluation, our algorithm is compared against three algorithms are used in the field of multi-objective optimization. These algorithms are NSGA-II, SPEA2, and OMOPSO. Objectives 3 and 4 are achieved in chapter 5.	JMetal tool is used to implement the algorithm by compiling the tool in NetBeans IDE. A comparison between the proposed algorithm and other algorithms is conducted by using benchmark test suites called Zitzler-Deb-Thiele (ZDT), and Walking-Fish-Group (WFG). In addition, three quality metrics are adopted to compare the convergence, accuracy, and diversity of results that obtained by these algorithms. These metrics are Inverted Generational Distance (*IGD*), Spread (*SP*) and Epsilon($\in$).
vi. To optimize a set of multi-objective problems related to wireless networks using the proposed Multi-Objective Optimization Algorithm (MOOA).	There are many problems related to WSN and wireless communication network. These problems need accurate solutions (optimal solution) at a short time with less effort. Smart grid network has been used as a case study to fulfill the objective number 5, which has many challenges affect the QoS of the whole network. Examples of these challenges are end-to-end delay and end-to-end latency. If the delay increases, the probability of dropping packets will be increased. The retransmission of these packets leads to consume more power and time. For these reasons, this work is intended to minimize both the end-to-end delay and end-to-end latency and also to maximize both energy efficiency and packet throughput of the smart grid network. Objective 5 is achieved in chapter 6.	JMetal tool is used to optimize different objective models related to the proposed problems.

3.6 Chapter Summary

This chapter can be summarized as follows:

1. Propose a Single-Objective Optimization Algorithm (SOOA) based on a metaphor of natural fertilization procedure.
2. Appraise the SOOA with different benchmarks functions. Then, its results are compared with the results from four algorithms in the area of Single-Objective Optimization (SOO). These algorithms are PSO, APSO, GA, and PGA.
3. Propose a Multi-Objective Optimization Algorithm (MOOA) as an extended version of the proposed SOOA based on the concept of non-dominated solution.
4. Evaluate the proposed MOOA with different benchmarks functions and compare its results against the results from three algorithms in the area of Multi-Objective Optimization (MOO). These algorithms are OMOPSO, NSGA-II, and SPEA2.
5. Optimize a set of multi-objective problems in WSN using the proposed Multi-Objective Optimization Algorithm (MOOA).

CHAPTER 4: SINGLE-OBJECTIVE OPTIMIZATION ALGORITHM BASED ON SPERM FERTILIZATION PROCEDURE

4.1 Introduction

Optimization is a procedure to find an optimal or near from the optimal solution. Researchers and scientists are inspired by this procedure to find optimization algorithms to solve challenging and non-linear problems (Andréasson, 2005). These problems can be represented as objective optimization functions. The objective function simulates different kinds of real-life problems in a form of mathematical models. This is done by simplifying, quantifying, and determining the limitations of the real problems. The objective functions can be divided into two quantities depending on the variables of the exact model. Examples of these objective functions are maximization and minimization objective functions. After determining the types of objective functions (e.g. minimum or maximum models), the modeling procedure can be devised to obtain a solution for these problems (objective functions). This can be performed by collaborating a wide variety of optimization algorithms. The modeling process is depicted in Figure 4.1 (Andréasson, 2005).

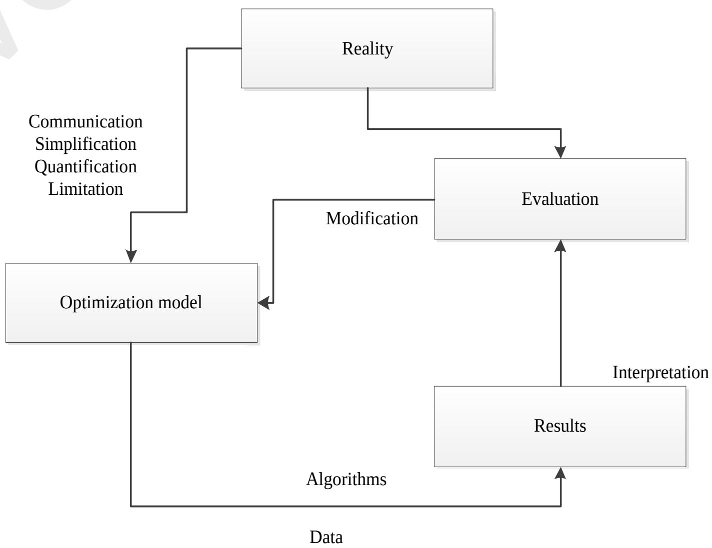


Figure 4.1: Modeling process (Andréasson, 2005)

The algorithms that are used in the modeling process can be classified into two groups; stochastic algorithms, and deterministic algorithms. Stochastic algorithms generate slightly various individuals regardless of their initial values, while deterministic algorithms have rigorous steps in their procedures, which can reach to some solutions if their procedures begin from initial solutions. Meta-heuristic algorithms have been used to find a solution for many non-linear complex problems. Examples of these problems are reliability-robust design optimization problems (Lagaros, 2010), permutation flow shop scheduling (Li, 2013), water and geotechnical engineering ((Gandomi, 2013a); (Yang, 2013)), engineering designs (Kaveh & Talatahari, 2011), education composition (Duan, 2012), and frequency assignment problem in mobile networks (da Silva Maximiano, 2012).

Optimization algorithms have been developed to find an optimal solution or a near optimal solution. Most of these algorithms are developed based on a metaphor of natural processes or manmade procedures. In practice, these algorithms can often achieve an optimal solution by begin the searching from a set of solutions (population) but it can be challenging to achieve the global optimality. Genetic Algorithm (GA) ((Goldberg, 1989); (Khoei, 2010)) proposed in 1960s is a powerful optimization method. Since then, various metaheuristic optimization methods (optimization algorithms) have been proposed, such as Cuckoo Search (CS) ((Gandomi, 2013b); (Gandomi, 2013c); (Wang, 2012a); (Yang, 2009)), Imperialist Competitive Algorithm (ICA) (Talatahari, 2012a), Animal Migration Optimization (AMO) (Li, 2014), Genetic Programming (GP) (Gandomi, 2011), Evolutionary Strategy (ES) ((Fogel, 1997); (Beyer, 2013)), Bat Algorithm (BA) ((Gandomi, 2011); (Gandomi, 2013d); (Dash, 2015); (Wang, 2012b); (Yang, 2010)), Ant Colony Optimization (ACO) (Dorigo, 2006), Differential Evolution (DE) ((Fan, 2011); (Gandomi, 2012); (Hachicha, 2011); (Li, 2012); (Storn & Price, 1997)), Probability-Based Incremental Learning (PBIL) (Baluja, 1994), Artificial Bee

Colony (ABC) (Karaboga & Basturk, 2007), Particle Swarm Optimization (PSO) (Kennedy, 1995), Harmony Search (HS) ((Geem, 2001); (Gholizadeh & Barzegar, 2013); (Wang, 2013)), Accelerated PSO (APSO) algorithm (Wang & Guo, 2014), Big Bang-big Crunch Algorithm (BBCA) ((Erol & Eksin, 2006); (Kaveh & Talatahari, 2009)), Parallel Genetic Algorithm (PGA) (Muhlenbein, 1991), and Charged System Search (CSS) ((Kaveh & Talatahari, 2010); (Talatahari, 2012b)).

PGA and APSO are the enhanced and simplified versions of GA and PSO respectively. PGA speeds up the convergence by independently selecting a mate for each individual depending on its neighborhood. Furthermore, PGA enhances the fitness of individuals during its lifetime using various search strategies such as local hill-climbing. Hill-climbing is a mathematical optimization procedure of a local search. Hill-climbing starts with a random solution to a problem. Then, attempts to get a better solution by creating an incremental change to the previous solution (Russell, 2003). APSO speeds up the convergence of PSO by using the global solution only. Previous studies have concluded that both PGA and APSO have the simplicity of structure, good optimization capability, and easy to implement.

PGA and APSO have a good convergence allowing the search space domain to be explored quickly. However, they may not always converge or reach toward a near optimal solution if the execution iteration is too large. This is because PGA deals with each solution (individual) independently and APSO operates based on a random walks of global best solution only. Therefore, in this chapter, an inherently continuous optimization algorithm is introduced. This algorithm uses the past location of each individual to determine its next location and uses different types of mutation operation to increase the algorithm convergence. This algorithm is inspired by sperm motility to fertilize the egg during human fertilization procedure. The objectives of this chapter are to test the proposed algorithm with different benchmark test functions and compared it with other well-known optimization algorithms such as GA, PGA, PSO, and APSO. This chapter consists of 5 sections. Section 4.2 summarizes the fertilization procedure and the sperm swarm optimization algorithm. Section 4.3 presents the experimental results the findings are discussed in Section 4.4. Finally, Section 4.5 summarizes this chapter.

4.2 Fertilization Procedure and Sperm Swarm Optimization Algorithm

Based on the limitations of prior optimization algorithms such as GA, PSO, PGA, and APSO, Single Objective Optimization Algorithm (SOOA) is proposed that operates based on the inherently continues technique to update the location of each individual in the search space domain. The proposed algorithm is inspired by the procedure of fertilization. This procedure is the epic story of a sperm struggling and facing incredible difficulties to unite with an Ovum (egg). Mostly, there are over than 130 million sperms competing to fertilize one Ovum, hence, in the normal case, there is only one sperm will fertilize the waited egg. In this subsection of Chapter 4, the procedure of the female reproductive system is summarized. This procedure is simulated as a SOOA. The fertilization procedure is described as follows:

The sperm swarms are triggered by the reproductive system of male locating inside the cervix, which the fertilization journey begins from this location. In the cervix, each sperm takes a random location to prepare itself in the fertilization journey, where each sperm in the swarm has two velocities on the X-axis and also on the Y-axis. These velocities of the sperm can be denoted as (Vsx[], Vsy[]).

Each sperm in the swarm swims forward from the cervix until reaching the Fallopian Tubes where the egg is waiting there. Researchers found that the sperms move in groups when they are swimming in viscoelastic fluids and their behavior of swimming exhibiting similar to “flocking”. The researchers also observed that the swarm has a certain degree of tail movements and beat of movement during grouping (Tung, 2015).

After insemination, the swarm will swim through the Fallopian Tubes. If the ovulation has occurred, the Ovum will release a chemical that attracts the swarm. This procedure of releasing the chemical called *Chemotactic*. Scientists think that sperm can find a waiting egg cell via a set of complex mechanisms. The sperm swarm swims toward the high-temperature location of the woman’s reproductive tract, where the *Ovums* (eggs) are located. This behavior of sperm swimming is called *Thermotaxis*. The researchers also found that the sperm prefers to move towards the positions of warmer temperatures (location of Fallopian Tubes). Sperm can sense and response to a temperature varied in the value less than 0.0006°C. Moreover, they found that the sperm swarm swims forward searching on the guidance (higher concentrations of molecules) that created and released by the Ovum, which this guidance called *Chemo-taxis* ((Bahat, 2012); (Bahat, 2006)). Based on these notions, we can realize that the sperm cell will not go backward to the cervix, but will swim forward towards warmer temperatures area (the Fallopian Tubes area where the egg is located). Each individual in the swarm changes its velocity and competes with other sperms as rally competition until they reach the *egg*. The first sperm reaches the egg location will release an enzyme created inside its acrosomes to make an opening (hole) in the membrane of the egg. This sperm is called as the winner because it enters the egg where only its head enters. Furthermore, the egg produces an order to the membrane to prevent other sperms from entering. This simulates the best solution that remarked by any sperm, which this sperm called the winner. The winner and other cells of the sperms reach the egg is depicted in Figure 4.2. The fertilization procedure (Carlson, 2012) inside the female reproductive system is summarized in Figure 4.3.

The sperm movement can be affected by many limitations as summarized below:

- pH value inside the female reproductive system:

1- A healthy female reproductive system has a normal pH value around 4.5–5.5 (Das Neves, 2006). However, a low pH of mucus acidic may deactivate and destroy the motility of sperm. For this reason, during the ovulation, the pH value of vaginal acidic or acid is in the range of 7 to 14, which is very suitable for sperm motility and is deemed very alkaline non-toxic to spermatozoa (Rodrigues, 2009).

2- There are many factors affecting the pH value of the female reproductive system, including, type of food consumed (Borges, 2011) and mood or emotional status of the female such as sadness or happiness, etc. (Edmunds, 2013). Based on this observation, we can estimate that the value of pH will be varied in the range of 7–14.

- Temperature inside the female reproductive system**:**

1- As we presented previously, the movement direction of sperms is affected by the temperature inside the female reproductive system. The scientists found that the sperm searches for a warmer area (the egg location), which acts like a temperature sensor (Bahat, 2012). The sperm head can sense and response to a temperature difference of <0.0006 degrees Celsius (Bahat, 2012).

2- Researchers found that the temperature inside the vagina can be varied based on female status. The temperature in the range of 35.1 °C to 37.4°C is considered as a normal temperature inside the female reproductive system (Christian, 2013). However, due to vaginal blood pressure circulation, the temperature may reach

38.5 °C (Health, 2011). Based on this observation, we can appreciate that the value of temperature will be varied in the range of 35.1–38.5.

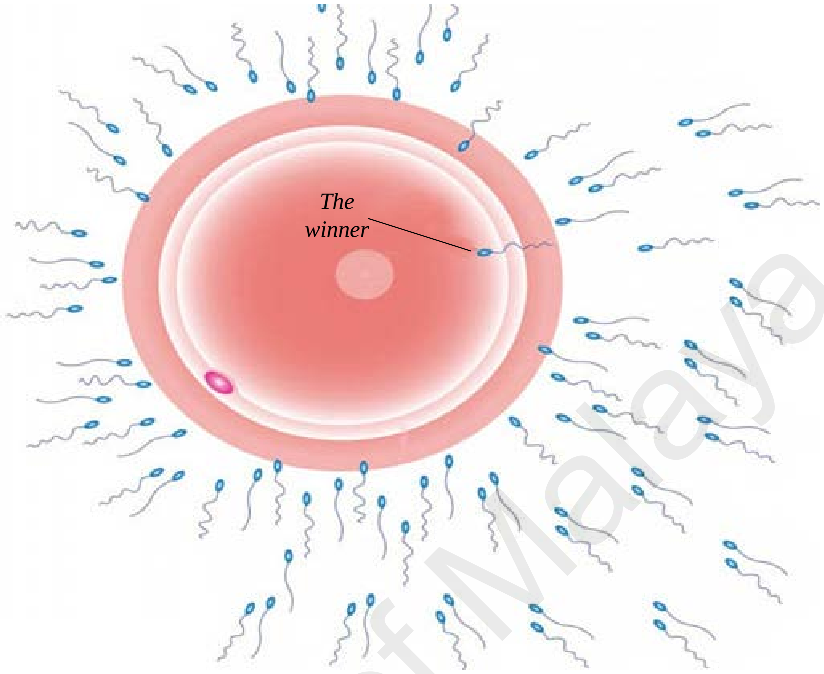


Figure 4.2: The winner and other sperms reach the egg (Facemama, 2017)

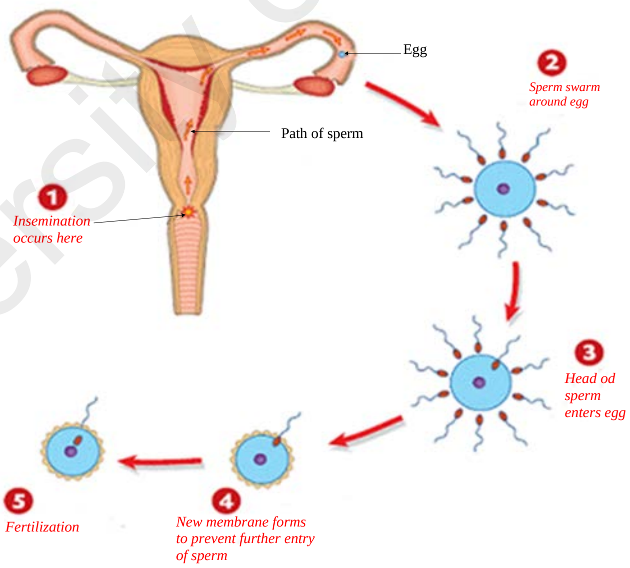


Figure 4.3: The fertilization procedure (Carlson, 2012)

Based on the prior information, we can observe that the sperm velocity is affected by the temperature and pH value inside the female reproductive system, which plays a significant role in sperm motility and its movement direction. The sperm velocity can be summarized in the following steps:

1) The initial velocity of spermatozoa: is the velocity that gained randomly by each sperm after the process of ejaculation in the cervical zone. Each sperm in the swarm takes a random position inside the Cervix and its velocity is affected by the pH value in that location. This velocity can be expressed as follows:

$$Initial\_Velocity = D \cdot V_i \cdot Log_{10}(pH\_Rand_1), \tag{4.1}$$

Where:

- D is a velocity damping factor, which takes a random number in the range of 0 to 1;
- V_i is the sperm cell velocity;
- pH_Rand_1 is a random number between (7, 14), which represents the pH value.

2) Personal sperm current best solution: refers to a best solution that obtained so far by the sperm itself. Based on aforementioned observation, sperm head behaves like a temperature sensor, which prefers to swim towards warmer temperatures (egg location) (Christian, 2013). Furthermore, researchers noted that sperm cell swims forward searching for the guidance knew as *Chemo-taxis* (higher concentrations of molecules) that produced and released from the egg (Bahat, 2006). Based on this acquaintance, we can realize that the swarm will not move backward to the Cervix, but will go forward towards warmer temperatures (the egg location inside the Fallopian tubes). This location can be reached by comparing the sperm current location on X-axis and Y-axis with a sperm past location that is stored in the memory. The past location can be replaced by the current location of the sperm, just if the current location of the sperm is better than its past location. The following equation is used to represent the personal sperm current best solution.

$$Current\_Best\_Solution = Log_{10}(pH\_Rand_2) \cdot Log_{10}(Temp\_Rand_1) \cdot (sb\_solution[] - current[]), \tag{4.2}$$

Where:

- Sperm Best (sb_ solution []): is the best solution achieved so far by a sperm;
- pH_Rand$_2$ is the pH value, which is a random number in the range of 7 to 14;
- Temp_Rand$_1$ is the area temperature. Its a random number in the range of 35.1 to 38.5.

3) Global best solution: refers to the sperm that currently closest to the egg (target), This sperm is labelled as the winner at the end. The sperm global best value can be represented by the following equation:

$$Global\_Best\_Solution = Log_{10}(pH\_Rand_3) \cdot Log_{10}(Temp\_Rand_2) \cdot (sgb\_solution[] - current[]), \tag{4.3}$$

Where:

- Sperm Global Best solution (sgb_solution[]) is the global best solution achieved so far by a sperm;
- pH_Rand$_3$ is the pH value, which is a random number in the range of 7 to 14;
- Temp_Rand$_2$ is the area temperature, which is random number in the range of 35.1 to 38.5;
- current[] is the current best solution, which is denoted by the following formula.

$$current[\,] = current[\,] + v[\,], \tag{4.4}$$

Where v[] is the sperm cell velocity. It can be achieved by merging the previous equations (velocities) in one equation as follows:

$$
\begin{aligned}
v[\,] = {} & Initial\_Velocity + Current\_Best\_Solution \\
& + Global\_Best\_Solution \,,
\end{aligned}
\tag{4.5}
$$

Based on Equation (4.5), there are three velocities to help the sperm reaching the location of the egg (the optimal solution). These velocities are the initial velocity of the sperm, personal sperm current best solution, and global sperm best solution. The initial velocity is affected by the pH value in the cervix zone while the personal sperm current best solution (personal velocity) and the global sperm best solution (global velocity) are affected by temperature and pH value of the visited area. However, the personal velocity is the best solution recorded by the sperm itself whereas the global velocity is the winner solution recorded by the whole swarm. Equation (4.5) can be clarified in the following equation:

$$
\begin{aligned}
V_i(t) = {} & D \cdot Log_{10}(pH\_Rand_1) \cdot V_i + Log_{10}(pH\_Rand_2) \cdot Log_{10}(Temp\_Rand_1) \\
& \cdot (x_{sbest_i} - x_i(t)) + Log_{10}(pH\_Rand_3) \cdot Log_{10}(Temp\_Rand_2) \cdot (x_{sgbest} - x_i(t))
\end{aligned}
\tag{4.6}
$$

Where D is a velocity damping factor, a random value in the range of (0, 1); pH_Rand_1, pH_Rand_2, and pH_Rand_3 are the pH values of the visited regions, a value in the range of (7, 14). $Temp\_Rand_1$ and $Temp\_Rand_2$ are the area temperature, a random number in the range of 35.1 to 38.5. X_{sbest} is the best solution achieved so far by a sperm. X_{sgbest} is the global best solution achieved so far by a sperm. The logarithm (log) is taken for both temperature and pH factors to normalize them to become small values. This will help achieving a slow acceptable velocity that simulates the normal motility of the real sperm. Based on the previous information and mathematical rules, a full procedure can be described as follows:

Algorithm 4.1 Sperm Swarm Optimization (SSO)

Begin
Step 1: initialize positions for all sperms.
Step 2: for *i=1: population size* ***do***
Step 3: *evaluate the fitness for each sperm*
        If *obtained fitness > sperm best solution* ***then***
        Set the current value as the sperm best solution
      End if

End for
Step 4: *choose the sperm global best solution based on the winner.*
Step 5: for *i=1: population size* ***Do***
Do the swim using velocity update rule
Update sperm location on the search space
End for
Step 6: *perform mutation*
Step 7: while *maximum iterations not achieved* ***return to step 2*** *and* ***repeat*** *until reaching the maximum number of iterations.*
End procedure.

Sperm Swarm Optimization (SSO) algorithm uses mutation operation to enhance the algorithm convergence and performance. Two different types of mutation can be applied in SSO algorithm. Examples of these mutation operations are non-uniform mutation and uniform mutation. These types of mutations are summarized in Section (2.4).

Based on the prior information, rules, and equations, it can be noticed that SSO algorithm is different than the existing algorithms such as Self-Organizing Migrating Algorithm (SOMA) and GA ((Deep, 2007); (Singh, 2016)). SOMA and GA are inherently discrete procedures. Classical GA and its adaptive methods (enhanced versions of GA using adaptive crossover, selection scheme, etc.), such as in (Nalepa, 2014) deal with each individual in the population independently. Therefore, they can easily use discrete design variables. For this reason, the static variables and parameters can be performed easily in their evaluation. In contrast, SSO algorithm is inherently continuous procedure that updates the location and velocity of the sperms based on the past position. SSO uses the random variables rather than static variables (i.e., pH and temperature) and play a significant role in updating the location of each individual until reaching the optimal location (optimal value). This randomness has been applied in many well-knows SOOAs such as PSO. Examples of these parameters are C_1 and C_2 in PSO algorithm, which have random values in the range of 0 to 4.

In SSO, the history of samples is not cached for each sperm such as temperature value and pH value of the visited area. Only, the location of each sperm in the swarm will be cached in the memory. This is because the position is the outcome of multiplying some numerical parameters with each other, such as the temperature value, pH value, sperm personal best solution, etc. Hence, the location of each sperm in the population is very important to be cached. This is due to SSO uses the past location (cached location) to compare it with the new location. The past location will be replaced if the new location is better than the past location as shown in Algorithm 4.1 and Equations 4.1 to 4.6.

4.3 Experiment and Results

The performance and efficiency of the proposed SSO are evaluated and validated by optimizing a set of benchmark test functions described as follows:

- Benchmark function is a non-linear model. It is used to evaluate different kinds of meta-heuristic methods (Krzeszowski, 2016).
- These benchmark functions are selected as they are considered as standard benchmark functions for evaluation SOOAs ((Marinakis, 2010); (Josiński, 2014); (Rbouh & Imrani, 2014); (Shirakawa; 2014); (Shi & Eberhart, 1999); (Valdez, 2009)).
- All the chosen benchmark test functions are minimization problems. Their results should be minimized through the evaluation process. Most of these benchmark functions have optimal values of zero. Examples of these benchmark functions are Sphere, Rosenbrock, Rastrigin, EggCrate, and Sum Squares (Surjanovic, 2013).

- The results of these benchmark test functions are used to compare between the proposed algorithm and other algorithms such as GA, PGA, PSO, and APSO algorithms.

The benchmark functions used in this chapter are discussed as follows.

- **The Sphere function:** This function is named from its shape that seems to be like a sphere as depicted in Figure 4.4 (Surjanovic, 2013). This benchmark model has a known global minimum value at (0, 0) with an optimal value equal to zero. The mathematical formula of this function is shown in Equation 4.7.

$$f_1 = \Sigma_{i=1}^{n} x_i^2, \tag{4.7}$$

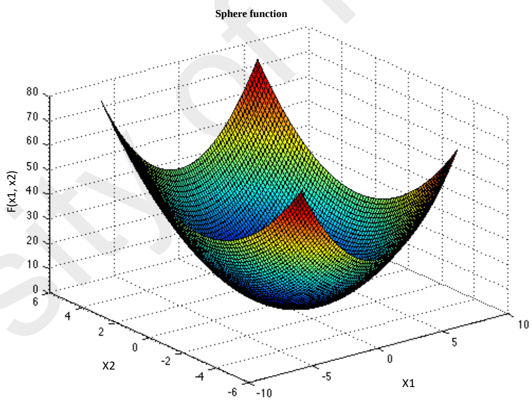


Figure 4.4: Sphere benchmark function (Surjanovic, 2013)

- **The banana (Rosenbrock) function:** This function has a shape of a banana. Equation 4.8 summarizes the mathematical description of this function. The Banana function has a well-known global minimum value at (1, 1) with an optimal value of zero. Rosenbrock function is depicted in Figure 4.5 (Surjanovic, 2013).

$$f_2(x) = \Sigma_{i=1}^{n-1}[100 \cdot (x_{i+1} - x_i^2) + (x_i - 1), \tag{4.8}$$

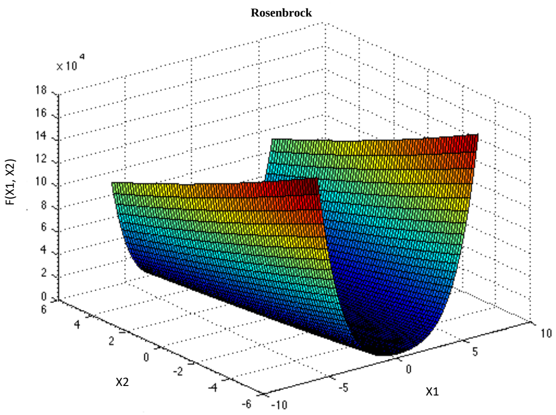


Figure 4.5: The Rosenbrock (banana) benchmark function (Surjanovic, 2013)

- **The Rastrigin function:** This function has several local minima and its global minimum is at the value of (0, 0) with an optimal value equal to zero. The mathematical description of Rastrigin function can be summarized in Equation 4.9. The Rastrigin function is depicted in Figure 4.6 (Surjanovic, 2013).

$$f_3(x) = \Sigma_{i=1}^{n}(x_i^2 - 10 \cdot \cos 2\pi x_i + 10), \tag{4.9}$$

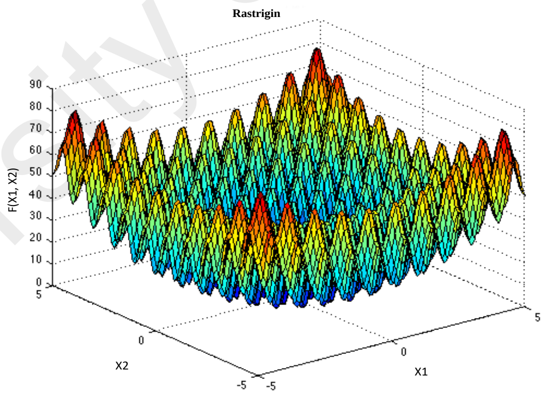


Figure 4.6: The Rastrigin benchmark function (Surjanovic, 2013)

- **The 2ⁿ Minima function:** The final optimal result and shape of this benchmark function are affected by the size of the search space domain. The mathematical formulation of this function can be summarized in Equation 4.10. 2^n Minima function can be depicted in Figure 4.7 (Surjanovic, 2013).

$$f_{4(x)} = \Sigma_{i=1}^{n}(x_i^4 - 16 \cdot x_i^{\,2} + 5 \cdot x_i), \tag{4.10}$$

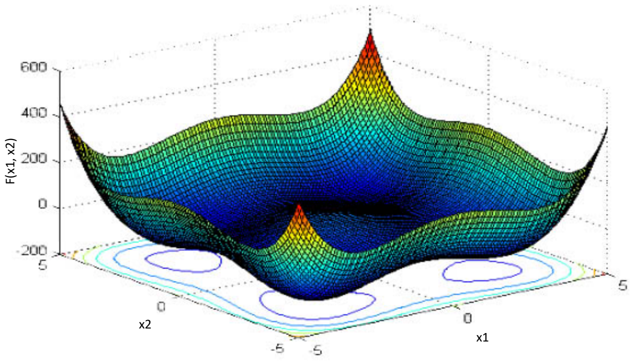


Figure 4.7: The 2^n Minima benchmark function (Surjanovic, 2013)

- **The EggCrate function:** This function has several local minima representing a multi-model minimization problem. EggCrate has a well-known global minimum value at (0, 0) with an optimal value of zero. Equation 4.11 presents the mathematical description of this benchmark function. Figure 4.8 shows the EggCreate benchmark function (Surjanovic, 2013).

$$f_{4}(x) = x_1^2 + x_2^2 + 25\left(\sin^2 x_1 + \sin^2 x_2\right), \tag{4.11}$$

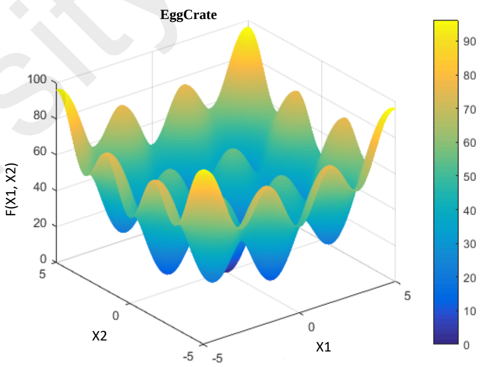


Figure 4.8: EggCrate benchmark function (Surjanovic, 2013)

- **The Sum Squares function:** This function is also known as Axis Parallel Hyper-Ellipsoid (APHE) function and has only a global minimum. Sum squares function has a well-known global minimum value at (0, 0) with an optimal value of zero. Equation 4.12 summarizes the mathematical description of this benchmark function and depicted in Figure 4.9 (Surjanovic, 2013).

$$f6(x) = \sum_{i=1}^{d} i x_i^2, \tag{4.12}$$

Where d is the dimensional size of the problem.

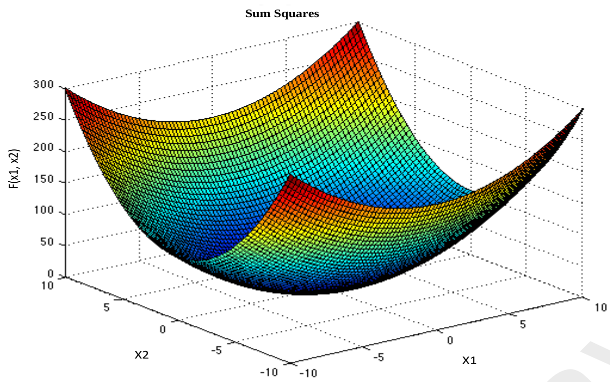


Figure 4.9: Sum Squares function (Surjanovic, 2013)

The dimensional sizes for each benchmark function are chosen carefully and set to 10, 20, and 30. These dimensional sizes are the standard sizes for evaluating the aforementioned benchmark functions ((Marinakis, 2010); (Josiński, 2014); (Shirakawa; 2014); (Shi & Eberhart, 1999)). Three different generations of the same population sizes of each algorithm are tested with those dimensions. Each benchmark function has the greater possible maximum value (denoted by X_{max}) in which equals to V_{max}. V_{max} and X_{max} are summarized in Table 4.1 The search domain for each function is represented in the third column of Table 4.1, where n is the dimension of each benchmark function.

Table 4.1: Functions X_{max} and V_{max} with search domain

Function	$X_{max} = V_{max}$	*Search domain*
Sphere *f1*	100	$[-100, 100]^n$
Rosenbrock *f2*	100	$[-15, 30]^n$
Rastrigin *f3*	10	$[-2.56, 5.12]^n$
2^n Minima *f4*	10	$[-5, 5]^n$
EGGCrate *f5*	10	$[-5, 5]^n$
Sum Squares Function *f6*	10	$[-10, 10]^n$

The development and experiments are conducted using Matlab version 7.0.4 running on Windows 7 (2 GB RAM, and Intel dual-core CPU). The performances of the proposed SSO algorithm are compared with four algorithms namely, PSO, APSO, GA, and PGA. Standard parameters as summarized in Table 4.2 are used to evaluate the algorithms. The standard parameters for the probability of mutation and the probability of crossover in GA and PSO algorithms are set to 0.05 and 0.8 as in ((Shi, 1999); (Saravanan, 2001)). Values of C_1 and C_2 in PSO are set to 2 as in (Zhang & Liu, 2004) while the parameters for both APSO and PGA are set as in ((Wang, 2014); (Muhlenbein, 1991)). Figure (4.10) presents the evaluation processes of the SSO algorithm on the search space domain of Sphere fitness function. The initial position of the sperms in the population on X-axis and Y-axis is shown in Figure (4.10. a). The positions of the search space domain simulate the sperms velocities denoted as (Vsx[], Vsy[]). After the initial velocity, all the sperm in a population searches for the optimal result (optimal position) of the fitness function. Their velocities are updated according to defined velocity update rule. The sperms change their position based on both personal current best solution and global best solution. Figure (4.10. b) and Figure (4.10. c) shows that the swarm is very near to the optimal solution and in Figure (4.10. d) the sperms stop at value 0, which is the optimal value of the Sphere benchmark function.

Table 4.2: Parameters of SSO, PSO, APSO, GA, and PGA

parameters	***Value***
SSO algorithm	
D: is a velocity damping factor	In the range of (0, 1)
pH	In the range of (7, 14)
Temperature	In the range of (35.5, 38.5)
Population sizes	20 and 40
Numbers of generations	100, 500, and 1000
PSO algorithm	
C_1 and C_2	2
Random range	In the range of (0, 1)
population sizes	20 and 40
Numbers of generations	100, 500, and 1000
APSO algorithm	
r	In the range of (0, 1)
β	In the range of (0.2, 0.7)
α	In the range of (0.1, 0.5)L
δ	In the range of (0.1.99)
population sizes	20 and 40
Numbers of generations	100, 500, 1000
GA	
Probability of mutation	0.05
Crossover probability	0.8
population sizes	20 and 40
Numbers of generations	100, 500, 1000
PGA	
Probability of mutation	0.05
Crossover probability	0.8
Population sizes	20 and 40
Numbers of generations	100, 500, 1000

Each test function is evaluated ten times. Table 4.3 summarizes the experimental results for SSO, PSO, APSO, GA, and PGA algorithms. In Table 4.3, the average of the final best value of the swarm is denoted by AVG while the best achievable fitness value for the benchmark functions is denoted by f(x). In this work, the number of iterations and generations are changed throughout the evaluation. This strategy of evaluating and comparing is used to evaluate many well-known algorithms such as Hurricane Search algorithm (HSA) and PSO, etc. ((Shi, 1991); (Rbouh & Imrani, 2014)). Two metrics to evaluate the performance and efficiency of the algorithms are explained below:

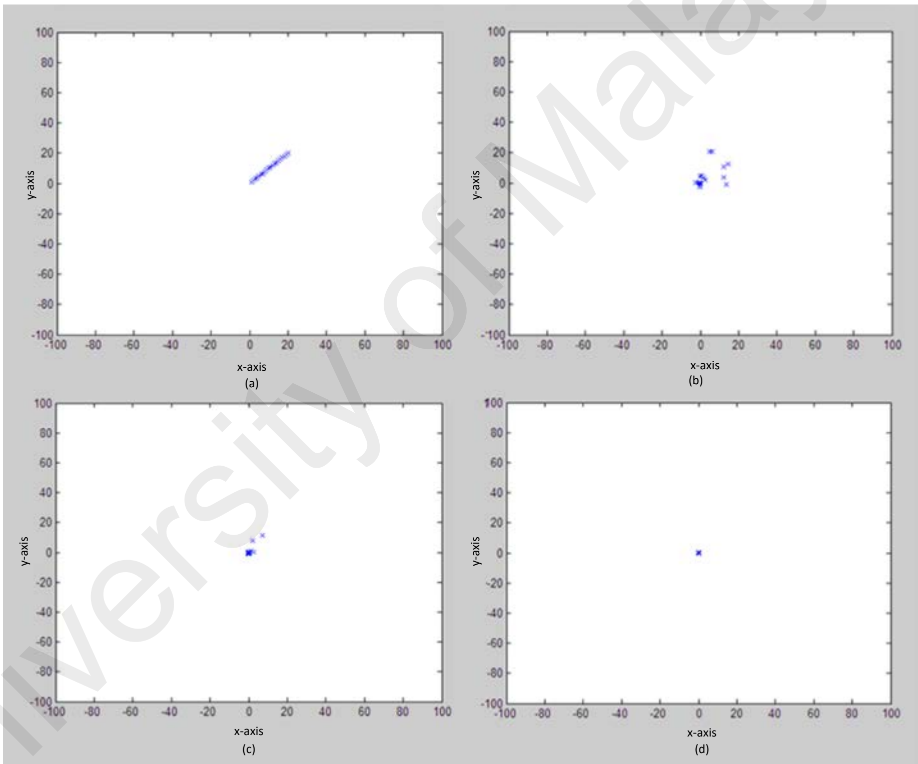


Figure 4.10: Evaluation processes of the SSO algorithm on sphere function. (a) The beginning of the search. (b) and (c) The sperms move toward the optimal position (optimal solution). (d) The sperms stop at 0 in which, the optimal solution for Sphere function

- **Convergence metric based on the variance of the population's fitness:**

Convergence is the ability of the meta-heuristic optimization algorithms to explore the whole search space domain. This metric is strongly related to the variance of the population's fitness. Therefore, if the variance of the population's fitness can be measured, the convergence can also be determined. The variance of the population's fitness can be measured as in Equation 4.13. This equation is also used to measure the convergence of an algorithm in solving a benchmark function at any generation.

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i - f_{avg}}{f}\right)^2, \tag{4.13}$$

Where:

- N is the number of chromosomes, sperms, or particles in the population;
- f_i *is* the fitness of the *i-th* chromosomes, particles, or sperms;
- f_{avg} *is* the current average fitness of the population;
- f is the normalized calibration factor to confine σ^2. The following equation is used to derive the value of *f:*

$$f = \max\{1, \max\{|f_i - f_{avg}|\}\}, i \in [1, N], \tag{4.14}$$

σ^2 represents the degree of convergence for all chromosomes, particles, or sperms in the population. A good convergence will yield a small value of σ^2 ((Ping, 2013), (Pant, 2008)).

- **Quality of Solution**

In some cases, the algorithm may converge away from the optimal result. Therefore, the ability of the algorithm to converge may not reflect a good result. For this reason, the effectiveness of the algorithm is measured. The effectiveness represents the ability of the algorithm to find the global best solution or a solution near to the global optimal solution when the metaheuristic algorithm begins the search from many random points on the search space domain. In other meaning, the effectiveness helps to determine how close the solution that obtained by the algorithm to the well-known global solution of the problem. The effectiveness (quality of a solution) can be measured as in Equation 4.15 (Saravanan, 2001):

$$Q_{sol} = \left| \frac{solution - knownSolution}{knownSolution} \right| \%, \tag{4.15}$$

It can be noted that Rosenbrock, Rastrigin, EggCrate and Sum Squares functions have optimal value of zero. An algorithm will have a good quality of result if its achievable fitness value is equal to the optimal value of these functions, which are zero. As shown in Table 4.3, the number of generations changes from 100 to 1000 generations to evaluate the convergence. An algorithm will have a good convergence if it can reach the optimal value with a less number of generations.

Based on the results from Figure (4.11) to Figure (4.16), and in Table 4.3, the algorithms can be ranked from the best achievable fitness value to the worse achievable fitness value for all benchmark test functions as listed in Table 4.4. From this table, the results of all benchmark functions can be summarized as follows:

Table 4.3: Experimental results where the optimal value for f1 to f6 is zero except for f4. The best value for f4 is -156.6647

Algorithm	*Function*	*Population. Size*	*Generation*	*The dimension of the function (n)*	**The best achievable fitness value *f(x)***	*Average final best value (Avg)*
SSO	Sphere *f1*	20	100	10	0.00000	0.00000
			500	20	0.00000	0.00000
			1000	30	0.00000	0.00000
	Rosenbrock *f2*	40	100	10	0.00000	3.15100
			500	20	0.00000	10.386E-08
			1000	30	0.00000	3.326E-07
	Rastrigin *f3*	40	100	10	0.00471	0.504261
			500	20	0.00406	0.495542
			1000	30	0.00226	0.495918
	2^n Minima *f4*	40	100	10	-156.6647	-156.6647
			500	20	-156.6647	-156.6647
			1000	30	-156.6647	-156.6647
	EGGCrate *f5*	40	100	10	0.00000	0.01562
			500	20	0.00000	0.01465
			1000	30	0.00000	0.013442
	Sum Squares *f6*	40	100	10	0.00000	0.00000
			500	20	0.00000	0.00000
			1000	30	0.00000	0.00000
	Sphere *f1*		100	10	0.00000	0.00000

PSO		20	500	20	0.00000	0.00000
			1000	30	0.00000	0.00000
	Rosenbrock *f2*	40	100	10	0.00000	4.53143
			500	20	0.00000	3.92E-09
			1000	30	0.00000	5.62E-08
	Rastrigin *f3*	40	100	10	0.3864	5.65491
			500	20	0.00000	4.59949
			1000	30	0.9950	4.89716
	2^n Minima *f4*	40	100	10	-156.6647	-145.15242
			500	20	-156.6647	-156.6647
			1000	30	-156.6647	-156.6647
	EGGCrate *f5*	40	100	10	0.00000	5.3337
			500	20	0.00000	3
			1000	30	0.00000	5.0001
	Sum Squares *f6*	40	100	10	0.00000	0.00000
			500	20	0.00000	0.00000
			1000	30	0.00000	0.00000
APSO	Sphere *f1*	20	100	10	3.4095e-032	2.55E-30
			500	20	3.7039e-156	3.04E-151
			1000	30	2.1229e-312	4.81E-303
	Rosenbrock *f2*	40	100	10	0.007695	3.583723
			500	20	0.006202	2.202E-07
			1000	30	7.6039E-010	7.6039E-010
	Rastrigin *f3*	40	100	10	0.99496	0.99496
			500	20	0.99496	0.99496
			1000	30	0.99496	0.9950
	2^n Minima *f4*	40	100	10	-156.6647	-156.6647
			500	20	-156.6647	-156.6647
			1000	30	-156.6647	-156.6647
	EGGCrate *f5*	40	100	10	8.0653e-030	1.91E-28
			500	20	8.2744e-155	1.79E-149
			1000	30	1.2512e-309	4.6299e-305
	Sum Squares *f6*	40	100	10	1.0648e-031	2.71E-29
			500	20	3.1837e-155	6.99E-151
			1000	30	1.7222e-310	8.06E-303
GA	Sphere *f1*	20	100	10	0.0000	0.14706
			500	20	0.0000	0.01623
			1000	30	0.0000	0.02758
	Rosenbrock *f2*	40	100	10	0.0000	0.85000
			500	20	0.0000	5.98763
			1000	30	0.0000	7.0303
	Rastrigin *f3*	40	100	10	0.0005	0.01746
			500	20	0.0000	0.021285
			1000	30	0.0000	0.023166
	2^n Minima *f4*	40	100	10	0.0000	0.035010
			500	20	0.0000	0.038100
			1000	30	0.0000	0.0319867
	EGGCrate *f5*	40	100	10	0.0000	0.006848
			500	20	0.0000	0.006184
			1000	30	0.0000	0.010755
	Sum Squares *f6*	40	100	10	0.0000	0.011384
			500	20	0.0000	0.00773
			1000	30	0.0000	0.009923
PGA	Sphere *f1*	20	100	10	0.0000	1.004675E-4
			500	20	0.0000	1.000004E-4
			1000	30	0.0000	9.928486E-5
	Rosenbrock *f2*	40	100	10	0.0000	0.007695
			500	20	0.0000	5.11542
			1000	30	0.0000	5.98623
	Rastrigin *f3*	40	100	10	0.0000	0.0041466
			500	20	0.0000	0.00129499
			1000	30	0.0000	2.8672288E-4
	2^n Minima *f4*	40	100	10	-0.0127686	-0.000898955
			500	20	-3.1567839	-0.159241
			1000	30	-13.716409	-8.67130934
	EGGCrate *f5*	40	100	10	0.0000	0.008854915
			500	20	0.0000	0.004965030

			1000	30	0.0000	0.007119022
	Sum Squares *f6*	40	100	10	0.0000	5.0914775E-5
			500	20	0.0000	5.0001215E-5
			1000	30	0.0000	5.0000061E-5

1. Sphere function: SSO algorithm outperforms GA, APSO, and PGA in finding the average final best value for this function when the number of generations are 1000, 500 and 100. SSO and PSO have a similar performance in solving the Sphere function. Both of them achieve the optimal solution of zero.
2. Rosenbrock function (banana function): SSO algorithm outperforms APSO in finding the average final best value for this function when the number of generations are 1000, 500 and 100. Also, SSO outperforms PSO in finding the average final best value of banana function when the number of generation is 100. SSO algorithm outperforms GA and PGA in finding the average final best value of this function when the number of generations are 1000 and 500.
3. Rastrigin function: GA and PGA are more capable to explore functions of several local minima compared to SSO and PSO. However, SSO outperforms PSO and APSO in solving this function on all generations as the mutation improves its convergence and performance.
4. 2^n Minima function: SSO algorithm outperforms GA and PGA in finding the average final best value of 2^n Minima function when the number of generations are 1000, 500, and 100. SSO algorithm outperforms PSO when the number of generation is 100. A similar performance is observed between SSO and APSO on all generations wherein the their final result is -156.6647.
5. EGGCrate function: APSO, GA, and PGA are more capable to explore functions of several local minima compared to SSO and PSO. However, SSO outperforms PSO in solving this function on all generations, which the mutation improves its convergence and performance.

6. Sum Squares function: SSO algorithm outperforms GA, APSO, and PGA in finding the average final best value for this function when the number of generations are 1000, 500 and 100. SSO and PSO have similar efficiency and performance in solving this function and obtained the final result of zero.

Table 4.4: Rank of the algorithms from the best achievable value to the worse achievable value for each benchmark function

Function	***The algorithms ranking from the best achievable fitness value to the worse achievable fitness value***
Sphere f1	*SSO and PSO have the same rank (first rank), followed by APSO, PGA, and GA respectively.*
Rosenbrock f2	*SSO, PSO, APSO, PGA, and GA respectively.*
Rastrigin f3	*PGA, GA, SSO, APSO, and PSO respectively.*
2n Minima f4	*SSO and APSO have the same rank (first rank), followed by PSO, PGA, and GA respectively.*
EGGCrate f5	*APSO, PGA, GA, SSO, and PSO respectively.*
Sum Squares f6	*SSO and PSO in the same rank (first rank), APSO, PGA, and GA respectively.*

Based on the results from Figure (4.17) to Figure (4.22), the algorithms can be ranked from minimum value of σ^2 (premature convergence) to maximum value of σ^2 (premature convergence), which means from higher value of convergence to lower value of convergence as in Table 4.5. The rank in Table 4.5 is obtained when the number of generation is 100. Therefore, it can be concluded that SSO has a good convergence especially at a small value of generations compared to other algorithms such as GA and APSO.

Table 4.5: The algorithms ranking from higher convergence to lower convergence

Function	***The algorithms ranking from higher convergence to lower convergence***
Sphere f1	*SSO and PSO in the same rank (first rank), PGA, APSO, and GA respectively.*
Rosenbrock f2	*SSO followed by PGA, APSO, PSO, and GA respectively.*
Rastrigin f3	*PGA followed by SSO, PSO, APSO, and GA respectively.*
2n Minima f4	*PGA, APSO, GA, SSO, and PSO respectively.*
EGGCrate f5	*PGA, SSO, APSO, GA, and PSO respectively.*
Sum Squares f6	*SSO and PSO in the same rank (first rank), PGA, APSO, and GA respectively.*

Figure 4.23 presents a comparison of solution quality between SSO, GA, PGA, PSO, and APSO for each benchmark function when the population size is 500 and 10 time runs. For Sum Squares, 2n minima, Rosenbrock, and Sphere functions, SSO and PSO algorithms obtained a better solution as their results are mostly very near to the optimal solutions. However, for EGGCrate and Rastrigin functions, GA, PGA, and APSO obtained a better solution as their results are very near or equal to the optimal results. These conclude that SSO and PSO are more superior in exploring the search space domain of functions with single local minima while GA, PGA, and APSO are more superior in exploring the search space domain of functions with several local minima.

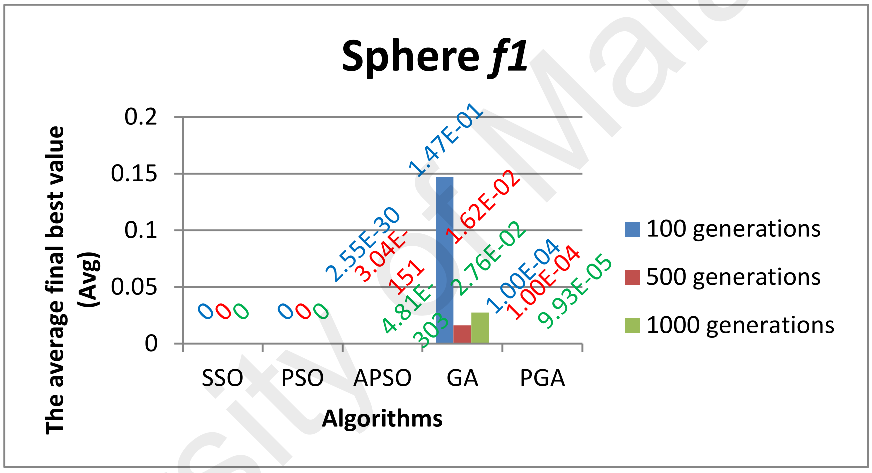


Figure 4.11: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of Sphere function

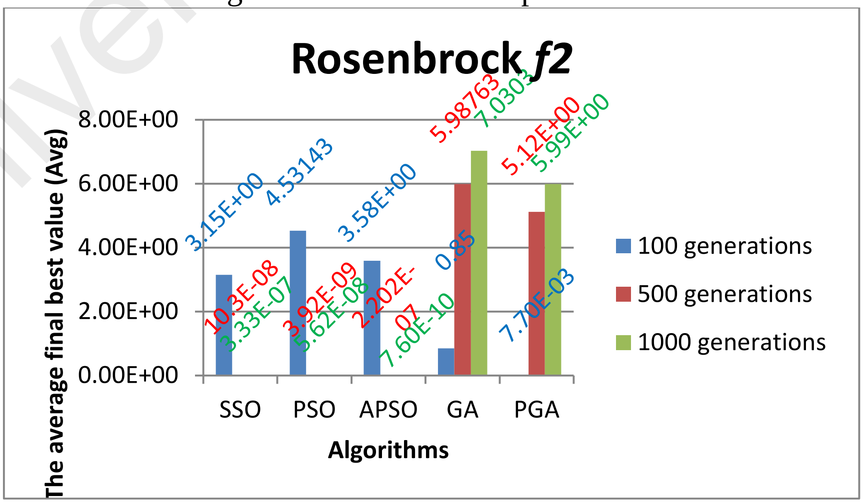


Figure 4.12: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of Rosenbrock function

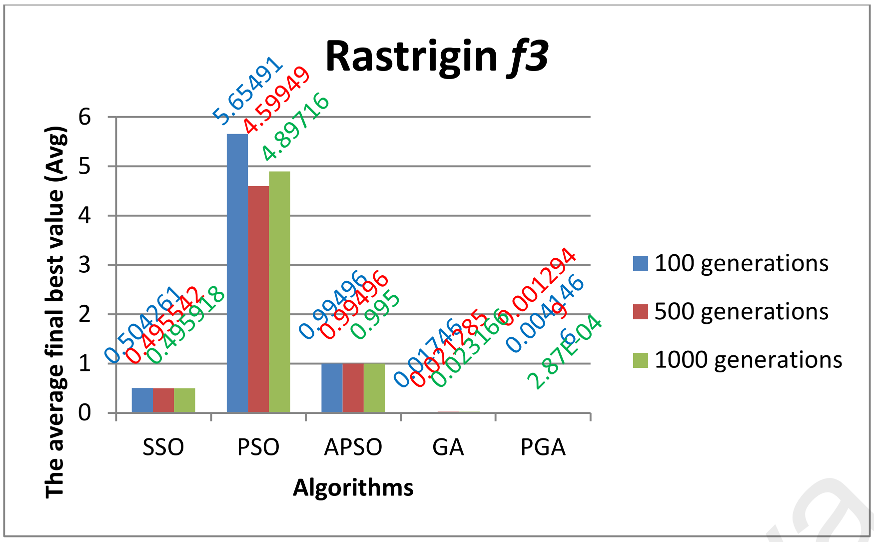


Figure 4.13: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of Rastrigin function

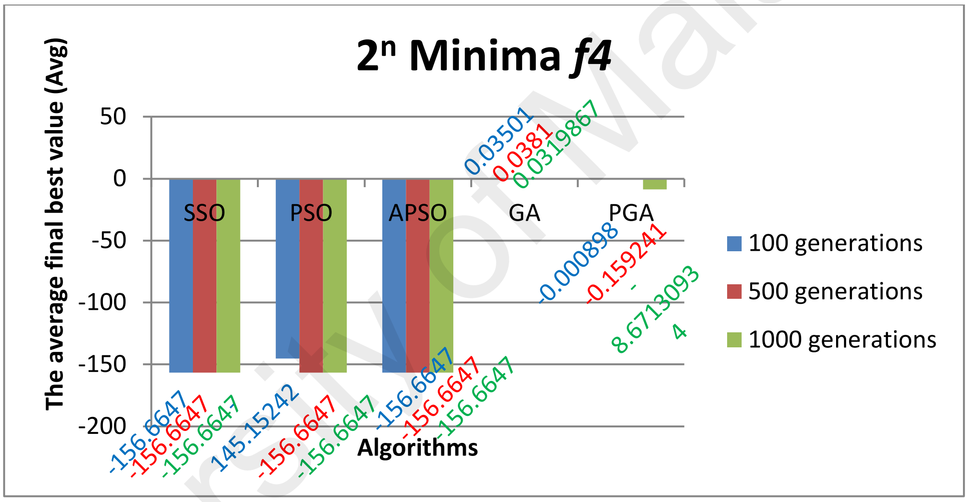


Figure 4.14: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of 2n Minima function

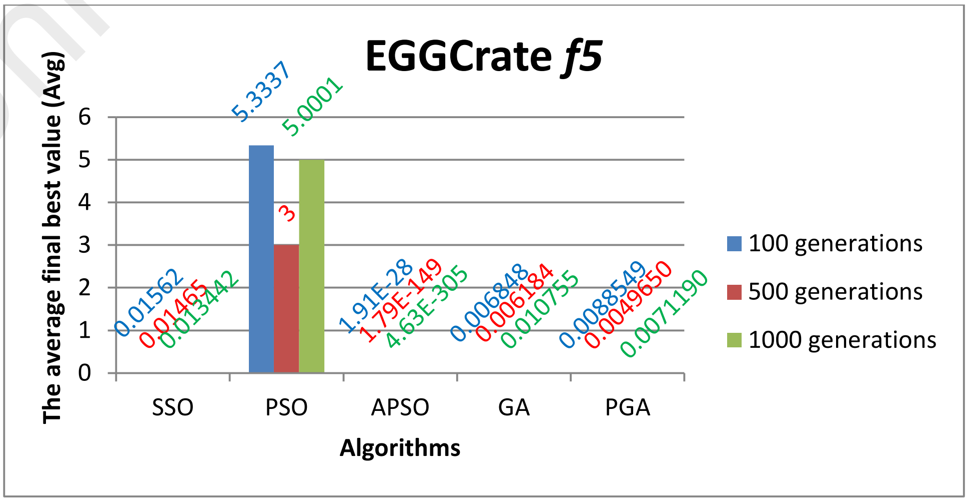


Figure 4.15: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of EGGCrate function

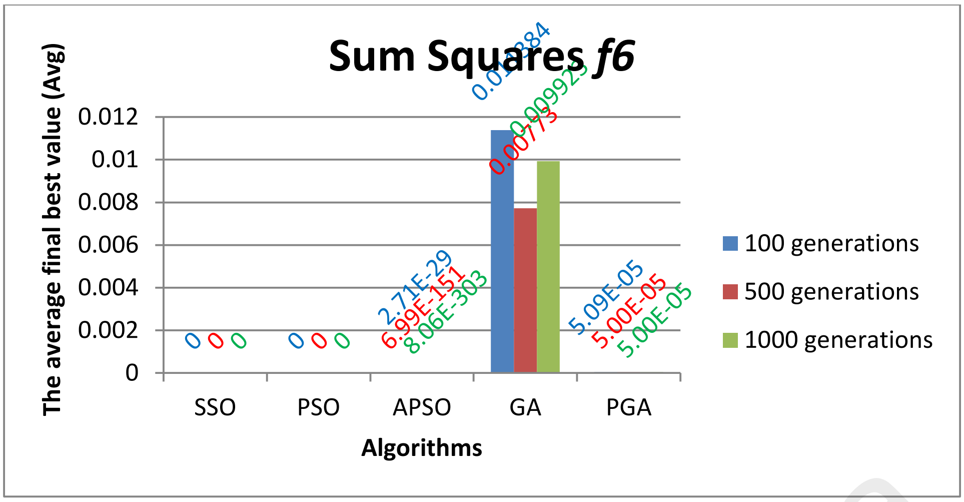


Figure 4.16: Comparison between SSO, PSO, APSO, GA, and PGA in finding the average final best value of Sum-Squares function

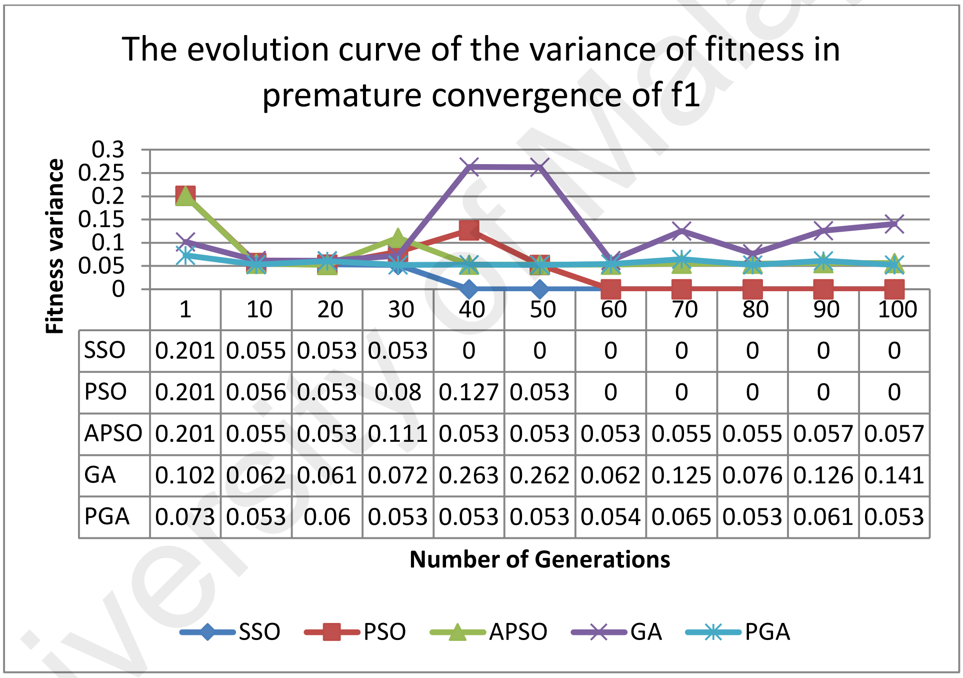
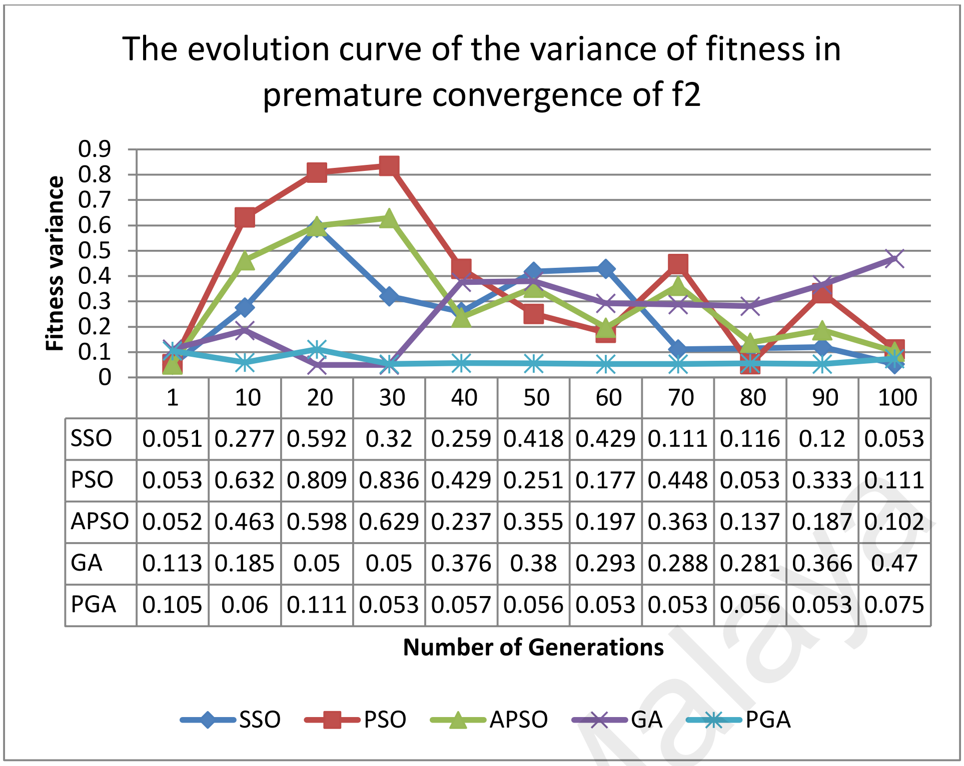
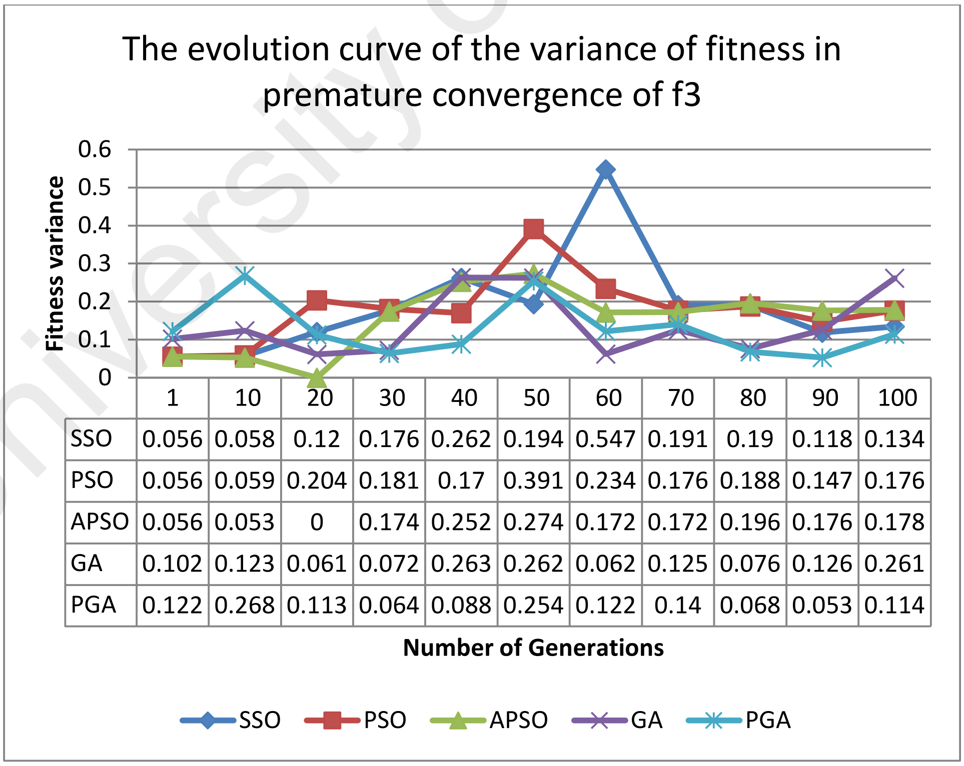


Figure 4.17: The evolution curve of the variance of fitness in premature convergence of f1



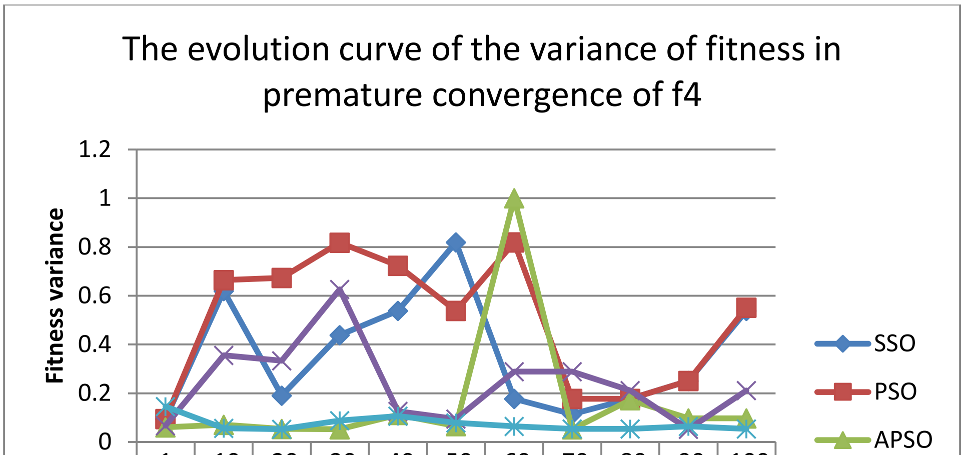
	1	10	20	30	40	50	60	70	80	90	100
SSO	0.051	0.277	0.592	0.32	0.259	0.418	0.429	0.111	0.116	0.12	0.053
PSO	0.053	0.632	0.809	0.836	0.429	0.251	0.177	0.448	0.053	0.333	0.111
APSO	0.052	0.463	0.598	0.629	0.237	0.355	0.197	0.363	0.137	0.187	0.102
GA	0.113	0.185	0.05	0.05	0.376	0.38	0.293	0.288	0.281	0.366	0.47
PGA	0.105	0.06	0.111	0.053	0.057	0.056	0.053	0.053	0.056	0.053	0.075

Figure 4.18: The evolution curve of the variance of fitness in premature convergence of f2



	1	10	20	30	40	50	60	70	80	90	100
SSO	0.056	0.058	0.12	0.176	0.262	0.194	0.547	0.191	0.19	0.118	0.134
PSO	0.056	0.059	0.204	0.181	0.17	0.391	0.234	0.176	0.188	0.147	0.176
APSO	0.056	0.053	0	0.174	0.252	0.274	0.172	0.172	0.196	0.176	0.178
GA	0.102	0.123	0.061	0.072	0.263	0.262	0.062	0.125	0.076	0.126	0.261
PGA	0.122	0.268	0.113	0.064	0.088	0.254	0.122	0.14	0.068	0.053	0.114

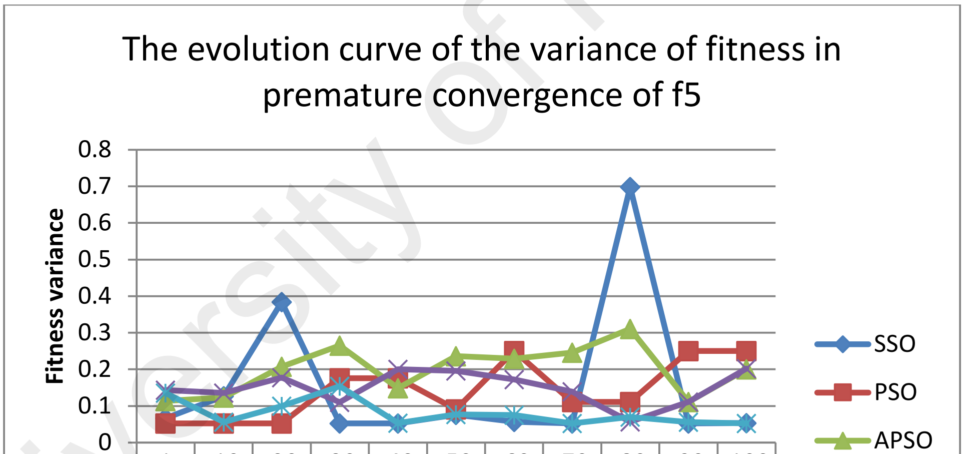
Figure 4.19: The evolution curve of the variance of fitness in premature convergence of f3



	1	10	20	30	40	50	60	70	80	90	100
SSO	0.09	0.62	0.19	0.44	0.54	0.82	0.18	0.11	0.18	0.25	0.54
PSO	0.09	0.66	0.67	0.82	0.72	0.54	0.82	0.18	0.18	0.25	0.55
APSO	0.06	0.07	0.05	0.05	0.11	0.07	1	0.05	0.17	0.1	0.1
GA	0.07	0.36	0.33	0.63	0.13	0.09	0.29	0.29	0.21	0.05	0.21
PGA	0.15	0.06	0.05	0.09	0.11	0.08	0.06	0.05	0.05	0.06	0.05

Number of Generations

Figure 4.20: The evolution curve of the variance of fitness in premature convergence of f4



	1	10	20	30	40	50	60	70	80	90	100
SSO	0.07	0.13	0.38	0.05	0.05	0.08	0.06	0.05	0.7	0.05	0.05
PSO	0.05	0.05	0.05	0.18	0.18	0.09	0.25	0.11	0.11	0.25	0.25
APSO	0.11	0.12	0.21	0.27	0.15	0.24	0.23	0.25	0.31	0.11	0.2
GA	0.14	0.14	0.18	0.11	0.2	0.2	0.17	0.14	0.06	0.11	0.2
PGA	0.14	0.06	0.1	0.15	0.05	0.08	0.08	0.05	0.07	0.06	0.05

Number of Generations

Figure 4.21: The evolution curve of the variance of fitness in premature convergence of f5

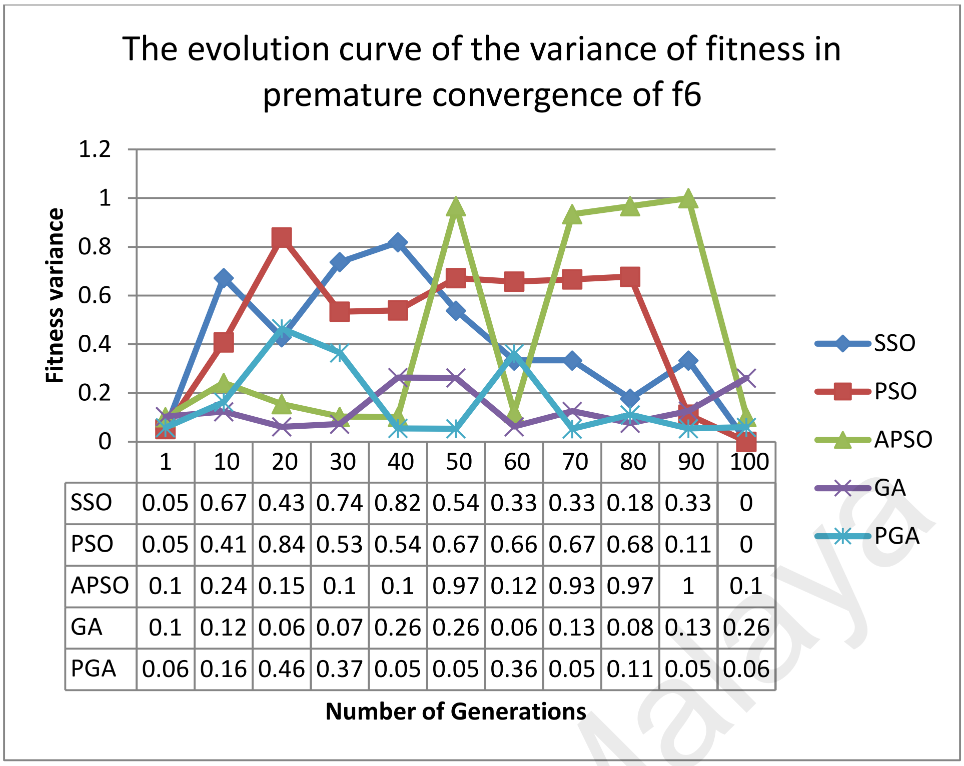


Figure 4.22: The evolution curve of the variance of fitness in premature convergence of f6

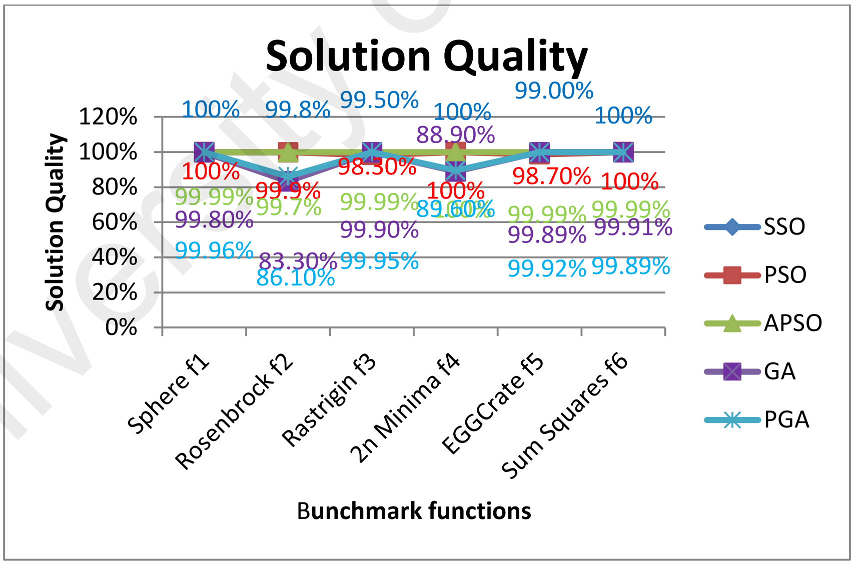


Figure 4.23: Comparison between SSO, PSO, APSO, GA and PGA in term of solution quality

4.4 Chapter Discussion

In this chapter, a SOOA called Sperm Swarm Optimization (SSO) is proposed. This algorithm is inspired by a natural procedure called fertilization procedure. Our findings based on results in Table 4.3 showed that our approach (SSO algorithm) has a good performance and convergence compared to other algorithms. In addition, the proposed SSO outperforms PGA, GA, PSO, and APSO in term of quality of results. SSO outperforms APSO in finding the average final best value of Sum squares, Rastrigin, and Rosenbrock functions for all generations while SSO outperforms PSO in solving EGGCrate function for all generations, Rosenbrock and 2^n Minima functions for 100 generations, and Rastrigin for all generations. Furthermore, SSO outperforms PGA and GA in finding the average final best value of Sum squares, 2^n Minima, and Sphere functions for all generations, Rosenbrock function for 100 generations. This proves that the proposed SSO has a good convergence especially for smaller generations such as 100 generations. Different types of mutation and the inherently continues approach in SSO help in solving test functions that have single local minima while PGA and GA are more capable at solving test functions with several local minima such as EggCrate and Rastrigin functions.

SSO is inherently continuous approach that stores past location for each sperm to determine their next location. SSO uses mutation operations to increase the convergence. This algorithm simulates the nature of the sperm through the fertilization procedure, as two factors affecting the sperm movement such as pH value and temperature value of the visited area.

Based on literature review, APSO and PGA are simplified and enhanced approaches of PSO and GA. These algorithms solve the problem of convergence in classical PSO and GA. PGA deals each chromosome separately and uses search strategies such as local hill-climbing to reach optimal solution with a good convergence while APSO uses only global best solution to increase the algorithm convergence.

4.5 Chapter Summary

This chapter can be summarized as follows:

- In this chapter, a SSO is proposed to solve optimization problems. SSO is a biological nature-inspired algorithm that based on the fertilization procedure of female reproductive system.
- SSO is tested with several benchmark functions and compared against a set of well-known algorithms in the field of SOO. Two metrics to evaluate the proposed algorithm are the convergence and quality of results.
- We can say algorithm *x* has a good quality of result if its achievable fitness value is equal to the optimal value of a benchmark function. The optimal values for Rosenbrock, Rastrigin, EggCrate and Sum Squares functions are zero.

  We can say algorithm *x* has a good convergence if it can reach the optimal value of a benchmark function in a smaller number of generations.
- A set of well-known benchmark functions have been chosen carefully to compare the proposed algorithm with other algorithms in finding the average final best value for all benchmark test functions at 1000, 500 and 100 generations.
- The results summarized in Table 4.4 shows that the proposed SSO outperforms APSO in finding the average final best value of Sum squares, Rastrigin, and Rosenbrock functions for all generations. Also, SSO outperforms PSO in solving EGGCrate function for all generations, Rosenbrock and 2^n Minima functions for 100 generations, and Rastrigin for all generations. Furthermore, SSO outperforms PGA and GA in finding the average final best value of Sum squares, 2^n Minima, and Sphere functions for all generations, Rosenbrock function for 100 generations.
- The results summarized in Table 4.5 shows that the proposed SSO has a better convergence when solving functions with single local minima. For functions with several local minima, GA and PGA have a better convergence.
- However, convergence sometimes may not reflect a good result as the algorithm may converge away from the optimal results. Therefore, the quality of results is

measured to determine the ability of an algorithm in finding the optimal or near to the optimal results. These metrics show that SSO has a higher ability to find the optimal results of functions with single local minima.

CHAPTER 5: MULTI-OBJECTIVE OPTIMIZATION ALGORITHM BASED ON SPERM FERTILIZATION PROCEDURE (MOSFP)

5.1 Introduction

Evaluationary algorithms and swarm intilligance algorithms to solve multi-objective optimization problems have extremely grown in the last decades. Evolutionary Multi-objective Optimization (EMO) and Swarm Intilligance Multi-objective Optimization (SIMO) are giving rise to a various optimization algorithms. Different types of techniques have been used by these algorithms, including, adaptive grid technique based on data structures to archive non-dominated vectors (Knowles & Corne, 2000), dominated tree (Everson ,2002), archive techniques (Coello, 2002), etc. These techniques help a wide variety of Multi-Objective Optimization Algorithms (MOOAs) to provide a solution for different Multi-Objective Optimization Problems (MOOPs) (Capitanescu, 2017). MOOPs consist of conflicting objectives in which some of them are minimization objective functions and the other are maximization objective functions. Examples of these objectives are network coverage objective function and network energy consumption objective function, in which, a maximization function and minimization function respectively. A detailed description of these objectives and their optimization is given in Section 2.5 of Chapter 2. The concept of Pareto optimality has been emerged to find a solution for the MOOP instead of applying the optimality concept of Single Objective Optimization (SOO), where the final results of MOOPs have been picked among a set of Pareto optimal solutions (Sakawa, 2013). The Pareto optimal concept is based on finding the non-dominated solutions that balances between minimization objective functions and maximization objective functions (Engelbrecht, 2006).

Based on the concept of Pareto optimality, many Single Objective Optimization Algorithms (SOOAs) have been extended to MOOPs. Some of these algorithms are inspired from the metaphor of man-made processes or natural procedure (Watanabe, 2004). For examples, Non-dominated Sorting Genetic Algorithm (NSGA) (Coello, 2013) and NSGA-II (Srinivas, 1994) are the extended versions of GA (Holland, 1992) while Multi-Objective Particle Swarm Optimization (MOPSO) and Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) (OMOPSO) are the extended versions of PSO algorithm ((Coello, 2002); (Sierra, 2005)).

These algorithms are used to search for a solution for different types of real-world MOOPs. Examples of these MOOPs include the design of mobile and telecommunication networks (Watanabe, 2001), defense applications (Hughes, 2004), rock crusher design (Barone, 2002), scheduling (Shaw, 1999), nuclear fuel management (Engrand, 1998), Yagi–Uda antenna design (Venkatarayalu & Ray, 2003), stationary gas turbine combustion process optimization (Buche, 2002), and distributing products through oil pipeline networks (Garcia, 2004). These problems have a set of objective functions that consists of maximization objective functions and minimization objective functions. Usually, there is a set of a trade-off and compromise solutions between these objectives, for these reasons, MOOA is created to find a range of solutions that offer a variety of tradeoffs between these objectives (Zitzler, 2004). SSO algorithm that proposed in Chapter 4 is a recent Single Objective Optimization Algorithm (SOOA) inspired by the fertilization procedure wherein sperms swim in swarms of flocks until reach the waited egg in the fallopian tubes. From Chapter 2, SSO has been developed to discover a solution for a set of optimization tasks, but until recently SSO has not been extended to deal with MOO problems. The performance of SSO reported and discussed in Chapter 4 indicates that SSO algorithm is suitable for Multi-Objective Optimization (MOO) due to a good quality of solutions and a high speed of convergence. Therefore, this chapter intends to extend SSO to solve MOOPs. For this purpose, this chapter will define Pareto dominance, archive operation and crowding factor. To validate the results of MOO, algorithms such as NSGA-II, OMOPSO, and SPEA2 are selected for comparison. The comparison strategy will be based on quantitative and qualitative techniques. The quality of Pareto front will be used in qualitative technique, while the three quality metrics, including, Inverted Generational Distance (IGD), Spread (SP) and Epsilon(ϵ) will be used in quantitative technique. This is organized as follows: Section 5.2 reviews the Sperm Swarm Optimization (SSO) algorithm. Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP) is presented in Section 5.3. Section 5.4 describes the details of the comparison strategy. Section 5.5 shows the experimental test and results. Section 5.6 discusses the findings and summary of this chapter is given in Section 5.7.

5.2 Literature review of the Sperm Swarm Optimization (SSO) algorithm

This section reviews the Single-Objective Optimization Algorithm (SOOA) called SSO algorithm that proposed in Chapter 4 by explaining its parts, highlighting its limitations and summarizing the SSO algorithm. SSO algorithm is the basic part of the Multi-Objective Optimization Algorithm (MOOA) that will be proposed in this chapter. The proposed MOOA will have the same rules and concepts of the SSO. This will be summarized in the following sub-sections.

5.2.1 Sperm Swarm Optimization (SSO) Algorithm

The Sperm Swarm Optimization (SSO) is an inherently continues approach proposed in Chapter 4. This approach is inspired by the fertilization procedure. SSO is represented as a distributed behavioral algorithm that can be applied to the multi-dimensional domain of search space. Through the simulation, the manner of motility of each individual in the population can be affected by either the current best local solution or best global solution of the sperm swarm. The former solution can be obtained by any individual based on the certain neighborhood, as each individual in the population remembers its past position to locate the new position. At the same time, the latter solution can be obtained by any sperm based on its position from the target, as this solution will be remarked by the sperm that has a very nearest position to the target (egg). The concept of the winner is used in this work, which refers to a sperm that has a very nearest position to the target. “Researchers found that the sperm swarms move together forming groups when they are swimming in viscoelastic fluids and their behavior of moving exhibiting similar to “flocking”. They also observed a certain movement beat and degree of synchronicity of tail movements during grouping” (Tung, 2016). The sperm swarm and winner are depicted in Figure 4.2.

Inherent discrete procedure is used by evolutionary algorithms such as GA and its enhanced versions. These versions use adaptive selection scheme, such as in (Nalepa, 2014) to deal with each chromosome independently and can easily execute discrete design variables. Therefore, the static parameters or variables can be used easily in their work. However, SSO algorithm uses the random parameters instead of static parameters, which play a significant role in updating the position of each sperm until reaching the optimal value.

The randomness has been used with most of the previous well-known algorithms ((Kennedy et al. 1995); (Geem, 2001); (Gandomi, 2013); (Karaboga & Basturk, 2007); (Rbouh & Imrani, 2014)). PSO (Kennedy, 1995) is one of the most popular swarm intelligence algorithms that uses randomness in its evaluation. C_1 and C_2 are two numerical variables in PSO that take random values in the range of 0 to 4. The adaptiveness of the SSO depends on two variables (i.e., pH and temperature). These factors take a range of numerical variables randomly based on the following rules:

1- A healthy female reproductive system has a normal pH value around 4.5–5.5 (Das Neves, 2006). However, low pH of mucus acidic may deactivate and destroy the motility of sperm. For this reason, during the ovulation, the pH value of vaginal acidic or acid is in the range of 7 to 14, which is very suitable for sperm motility and is deemed very alkaline non-toxic to spermatozoa (Rodrigues, 2009). There are many factors affecting the pH value of the female reproductive system, including, type of food consumed (Borges, 2011) and mood or emotional status of the female such as sadness or happiness (Edmunds, 2013). Based on this observation, we can estimate that the value of pH will be varied in the range of 7–14.

2- The direction of sperms movement is affected by the temperature inside the female reproductive system. The scientists found that the sperm searches for a warmer area (the egg location), which acts like a temperature sensor (Bahat, 2012). In addition, the sperm head can sense and response to a temperature difference of <0.0006 degrees Celsius (Bahat, 2012). Researchers found that the temperature inside the vagina can be varied based on female status. The temperature in the range of 35.1 °C to 37.4°C is considered as a normal temperature inside the vagina (Christian, 2013). However, due to vaginal blood pressure circulation, the temperature may reach 38.5 °C (Health, 2011). Based on this observation, we can appreciate that the value of temperature will be varied in the range of 35.1–38.5.

Based on the prior rules, we can observe that SSO algorithm is different than evaluation algorithms such as GA and its adaptive methods such as in (Nalepa, 2014) (the version that uses adaptive selection scheme as enhanced versions of classical GA), which they are an inherently discrete procedure. These procedures deal with each individual in the population independently; therefore, they can easily use discrete design variables in their evaluation. In other meaning, GA and its enhancement versions use selection operation to select the best solutions and discard the other solutions. The GA procedure and its enhancement versions do not use velocity update rules to generate new solutions instead use a set of operations to find an optimal result of an objective function. Examples of these operations are selection, crossover, and mutation. Details procedure of GA can be found in Section 2.2.1 of Chapter 2 while the types of crossover and mutation operations in Section 2.5. SSO algorithm performs the inherently continuous procedure to update the sperm location. Each individual in the swarm remembers its past position and based on it produces a new position. SSO algorithm uses random parameter rather than the static parameter (i.e., pH and temperature) to update the position of each sperm in the population until reaching the optimal value. The history of samples is not cached for each individual in the swarm such as temperature value and the pH value of the visited location. The procedure of SSO cached only the location of each sperm. This is because the location is the outcome of multiplying some numerical variables with each other, such as the temperature value, pH value, and sperm best solution, etc. Therefore, the location of each individual is very important to be cached. SSO compares the old location (cached location) with the new location and update it with the new location if it is better than the old one. This is clear in the SSO procedure (Algorithm 4.1) in Chapter 4. Furthermore, mutation operations are used in this algorithm to increase the convergence. SSO uses mathematical rules (velocity rules) to update the sperm velocity. These rules are summarized in Section 4.2 and Equations from 4.1 to 4.5.

The proposed extension of SSO algorithm as a MOOA uses winner and Pareto dominance concepts. The winner is the closest sperm to the target (egg). The value of winner is considered as the best value in the whole swarm. This value is the global best value and used as a reference value for other sperms in the swarm to adjust their velocities on the search space domain.

The advantages of MOOA over SOOA is the ability of MOOAs of searching a set of possible solutions that manage tradeoffs between a set of conflicting terms to find an optimal solution. Based on this, the SSO can be utilized easily in solving many MOOPs such as the aforementioned MOOPs in Section 5.1.

5.3 Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP)

The main idea of MOOAs is to discover Pareto-optimal set to manage the tradeoffs between a set of conflicting objective functions and also to balance them. The popular Italian economist (Vilfredo Pareto) is the founder of the hypothesis of Pareto optimality, which is conceptualized in his study called Manual of Political Economy (MPE) in 1906 (Engelbrecht, 2006).

The definition of Pareto optimality and Pareto optimal set are based on some basic notions introduced as follows:

- Domination: v_1, may dominate a position vector, v_2 ($v_1 \prec v_2$), if and only if $f_m(v_1) \le f_m(v_2), \quad \forall \mathrm{m} = 1, \cdots \mathrm{n_m}$ and $f_m(v_1) < f_m(v_2)$, for at least one m (Chamaani, 2007).
- Pareto optimal: A vector $v* \in$ F is defined as Pareto optimal if there no vector $v \in F$ such that $f_m(v) \le f_m(v^*), \quad m \in N, and \quad f_m(v) < f_m(v^*)$ for at least one $m \in N$. An objective vector $z^* = \mathrm{f}(v^*)$ is called Pareto optimal if the corresponding vector v^* is the Pareto optimal. The set of Pareto optimal decision vectors $v^* \in$ F is denoted by $P \subseteq F$ (Lindroth, 2010).
- $v^* \in F$ is Pareto optimal of a position vector if no position vector dominates it, $v \ne v^* \in F$. On the other meaning, the Pareto optimal solution is non-dominated solution, which not dominated by other solutions (Lindroth, 2010).
- Pareto-optimal set: is a set containing all the Pareto optimal vectors $\mathrm{P_S} = \{v_\circ \mid \neg \exists v_1 \prec v_\circ\}$ (Zheng, 2010).
- Pareto front (P): can be defined as $\mathrm{P} = \{(f_1(v), f_2(v), \cdots f_N(v)) \mid v \in p_S\}$ (Zheng, 2010).

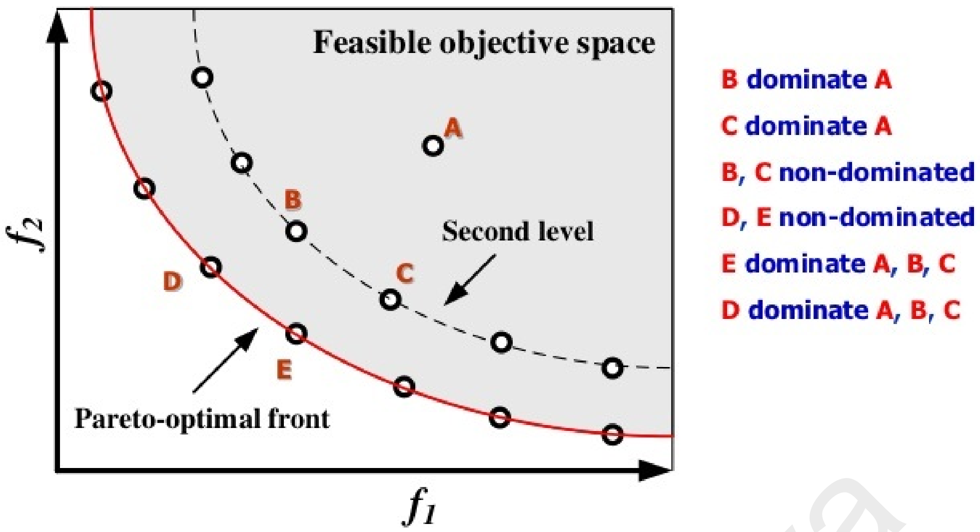


Figure 5.1: Non-dominated solutions (Goldberg, 1989)

The concept of the Pareto front and the Pareto optimal set can be simplified as in Figure 5.1 (Goldberg, 1989). From this figure, it can be noted that solutions B and C dominate solution A while solutions D and E dominate all the other solution. Solutions D and E are not-dominated by any other solutions. As solutions D and E are non-dominated solutions, they will form the Pareto front set.

The aim of any MOOA is to minimize the distance between the solution and the true Pareto front. To fulfill this objective, appropriate objective functions must be defined. Many classical methods and approaches have been developed to assign fitness function. Example of this method is aggregation–based method that defines the fitness function based on the weighted sum of the objective functions (Lei, 2007).

The classical methods have many limitations such as tend to be inefficient and very susceptible to precise accumulation of goals (Engelbrecht, 2006). For these reasons, various MOOAs have been proposed, where some of them are complex methods that based on neural network to obtain optimal weights of the fitness functions (Lee, 2004). Some of these MOOAs are based on Pareto dominance uses for fitness assignment, where the fitness value is proportional to the dominance rank of solutions. The concept of Pareto dominance is used in many algorithms to find a set of best solutions or optimal solution for MOOPs. MOPSO is an example of these algorithms, which is used to find a solution for a wide variety of MOOPs based on Pareto dominance (Coello, 2004).

In this chapter, Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP) is proposed, the Pareto dominance and crowding factor (Deb, 2002) have been used in the algorithm. Crowding factor helps to build criterion for a second discrimination (additional to Pareto dominance). This factor also helps to decide the best solutions (winners) that should be reserved when the algorithm performs the maximum list size. In MOSFP algorithm, the winner is selected based on means a binary tournament that is referred to crowding value of the winners. This algorithm crowds the best solutions in a set called Set of Winners (SoW), which the maximum size of SoW equal to the size of the population. At the end of each generation, the algorithm updates SoW and the corresponding crowding values related to them. If the size of SoW is greater than the maximum defined size, the algorithm excreted the best winners based on their crowding value and reserves them, moreover, the worst winners will be eliminated. This helps to select the best solutions without the need for any kind of selection criterion. That is why the crowding factor is used in many researches such as ((Sierra, 2005); (Ray, 2002); (Li, 2003)).

The mutation operations have been used in MOSFP algorithm as well as in many well-known optimization algorithms such as MOPSO, OMOPSO, and NSGA-II ((Coello, 2013); (Coello,2002); (Sierra, 2005)). Examples of mutation operations are non-uniform mutation and uniform mutation. The non-uniform mutation provides a non-constant variability range for each decision variable while uniform mutation provides a constant of variability range for each decision variable. In MOSFP algorithm, a combination of both non-uniform mutation and uniform mutation as in (Sierra, 2005) is proposed. This helps to modify the values of the decision variables of a sperm. Furthermore, the algorithm is also motivated by not using mutation at all, which, if the mutation operations fail to find the convenient solutions, the part without mutation will be reserved on proper results. For this purpose, a population size is divided into three equal parts by taking the modulus of 3 on population size. In the first part of the population, no mutation will be applied at all. The non-uniform mutation will be applied to the second part of the population and uniform mutation to the third part of the population. Algorithm (5.2) summarizes the mutation part of the MOSFP algorithm. Details descriptions of uniform mutation and non-uniform mutation operations are given in Section 2.4 of Chapter 2.

MOSFP also uses the concept of $\in -$ dominance employed in many researches such as ((Sierra, 2005); (Laumanns, 2002); (Yue, 2016)). This concept performs an external archive to fix the size of non-dominance solutions. External archive is the process of storing the non-dominated solutions in the memory. Usually, a decision vector v_1 $\in -$ dominance a decision vector v_2 for some $\in > 0$ if and only if: $f_m(v_1)/(1+\in) \leq f_m(v_2), \forall m = 1,.....,n$ and $f_m(v_1)/(1+\in) \leq f_m(v_2),$ for at least $m = 1,.....,n$. The size of final external archive ($\in -$ value) can be defined by the user manually. However, the same value of all objective functions is used, which are changed based on the amount of points in the final Pareto-front. This technique is used in many researches such as (Sierra, 2005).

Algorithm 5.1 Mutation

```
Begin
Step 1:  for i = 0  to population size do
        If (i % 3 == 0) then
            Sperms_ mutated with a non-uniform mutation operator
              Else if (i % 3 == 1) then
                 Sperms_ mutated with a uniform mutation operator
                    Else
                       Sperms_ without mutation
        End if
     End for
End procedure.
```

Algorithm (5.2) summarizes the full procedure of MOSFP. MOSFP begins by initializing the sperm swarm. SoW is set based on the non-dominated individuals. Later, the crowding factor will be calculated. For each sperm in the population at each generation, the swim (velocity update rule) and mutation operations that are described in Algorithm 5.1 is executed. Velocity update rule (swim) as described in Equation 5.1 is performed for each sperm in the population.

$$V_i(t) = D \cdot Log_{10}(pH\_Rand_1) \cdot V_i + Log_{10}(pH\_Rand_2) \cdot Log_{10}(Temp\_Rand_1) \cdot (x_{sbest_i} - x_i(t)) + Log_{10}(pH\_Rand_3) \cdot Log_{10}(Temp\_Rand_2) \cdot (x_{sgbest} - x_i(t)), \quad (5.1)$$

Where:

- D is the velocity damping factor, a random number in the range of 0 to 1;
- $pH\_Rand_{1,2,3}$ are random numbers between (7, 14), representing the pH value;
- V_i is the sperm cell velocity;
- $Temp\_Rand_{1,2}$ are the area temperature, a random number in the range of 35.1 to 38.5;
- Personal sperm best solution (X_{sbest}) is the best solution achieved so far by a sperm;
- Sperm global best solution (X_{sgbest}) is the global best solution achieved so far by a swarm;
- $x_i(t)$ is the current best solution denoted by the following formula.

$$x_i(t) = x_i(t) + v[\,], \quad (5.2)$$

Algorithm 5.2 Multi-objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP)
1: **Begin** 2: ***Step 1:*** *initialize positions for all sperms.* 3: ***Step 2:*** *initialize Winners (sperms with the best solutions).* 4: ***Step 2:*** *archive the Winners in* $\in -archive$ 5: ***Step 3:*** *crowd the winners using crowding operation.* 6: ***Step 4:*** *define counter (i) and define number of maximum iterations* (i_{Max}).

```
Step 5: do //this do is a do - while
        For <each sperm> do
              Select winner from the sperm swarm
              Update sperms positions using the predefined sperm velocity update
              rule (perform swim)
              Perform mutation procedure (Algorithm 5.1)
              Evaluate the fitness for each sperm (result of multi-objective
              optimization problem)
              Update personal sperm current best solution
        End for
        Update Set of Winners (SoW)
        Archive winner in ∈−archive
        Crowd the SoW using crowding operation
        Update value of the counter (i)
Step 6: While i < i_Max
Step 7: archive results in ∈−archive
End procedure.
```

The process of updating personal sperm current best solution is based on the two cases; first, dominated by the new sperm; second, with the new sperm value that are non-dominated with respect to each other. Then, the SoW will be updated. If all the sperm are updated, the individuals with new locations that are better than the old locations will have the possibility to join the SoW. Next, $\in -archive$ will be updated. Finally, the procedure crowds the individuals based on crowding values of SoW, which many of the winners are discarded if exceeding the determined size of the SoW. The procedure is repeated until the determined number of maximum iterations (i_{max}) is reached. The parameters used in this algorithm are swarm size (*S_size*), number of iterations (i), mutation rate (*mutRate*), which is automatically computed, and the value of bounding the size of the $\in -archive$ ($\in$).

5.4 Comparison Strategy

Benchmark functions are used to compare MOSFP with the most popular algorithms in MOO. To perform the quantitative evaluation on the performance and efficiency of MOOA, three issues are taken into consideration ((Coello, 2004); (Guzek, 2012)).

- Minimize the distance between the Pareto front that generated by our algorithm and the global Pareto front of any problem. Global Pareto front is the well-known Pareto front of any problem. Examples of these well-known Pareto fronts are the Pareto fronts of Zitzler-Deb-Thiele (ZDT) test suite, and Walking-Fish-Group (WFG) test suite. Evaluation metric: Inverted Generational Distance (*IGD*).
- Maximize the spread of the solution that generated by our algorithm, so uniform distribution of vectors can be obtained. Evaluation metric: Spread (*SP*).
- Maximize the convergence of our proposed approach to achieve a good quality of the Pareto optimal set found. Evaluation metric: Epsilon ($\in$)

To evaluate the above issues, three frequently used metrics are employed as in ((Coello, 2004); (Guzek, 2012); (Guzek, 2014)). These metrics are Inverted Generational Distance (*IGD*), Spread (*SP*) and Epsilon ($\in$) described as follows:

1. Inverted Generational Distance (*IGD*): Introduced by Van Veldhuizen et al. ((Dai, 2016); (Bezerra, 2017); (Yang, 2007)) to measure the distance between the Pareto front that generated by an algorithm to the true Pareto front of any problem. *IGD* can be measured by using the following equation:

$$IGD = \left(\sqrt{\sum_{t=1}^{n} d_i^2} \right) / n, \tag{5.3}$$

Where:

- n is the number of non-dominated vectors that produced by an algorithm;
- d_i is the Euclidean distance that measured in object space between vectors found by an algorithm and the true Pareto front of a problem.

IGD with a zero value indicates that all solutions found by an algorithm are in the Pareto front of the problem ((Monroy, 2004); (Jiang & Cai, 2009)). Therefore, the *IGD* value for an algorithm is preferred to be equal or near to zero. Again, *IGD* in this chapter is used to compare a Pareto front that generated by an algorithm with a reference Pareto front (the true Pareto front) of the same problem.

2. Spread (*SP*): A diversity metric, which calculate the distribution of solutions and the extent of spread is the set of optimal solution (*S*). The concept of spread as illustrated in Figure 5.2 shows five non-dominated solutions of *S*. These solutions are spread in two cases. The first case shown in Figure 5.2 (a) has a good distribution of solution but a poor spread, as the optimal set (*S*) does not have the radical points such as (0, 1), (1, 0) on the 2-dimensional Pareto front. The second case shown in Figure 5.2 (b) has a very good spread but unfavorable distribution of an *S*, which the solutions are distributed on the whole solution domain (Jiang, 2014). *SP* can be defined according to (Riquelme, 2015):

$$\Delta(S,P) = \frac{d_f + d_l \sum_{i=1}^{|S|-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (|S|-1) \cdot \bar{d}}, \tag{5.4}$$

Where:

- d_i is the Euclidean distance between consecutive solutions;
- $\bar{d}$ is the average of d_i;
- df and d_l are the minimum Euclidean distance measured based on the distance between solutions in the optimal set (*S*) to the Pareto front (P) radical (bounding) solutions.

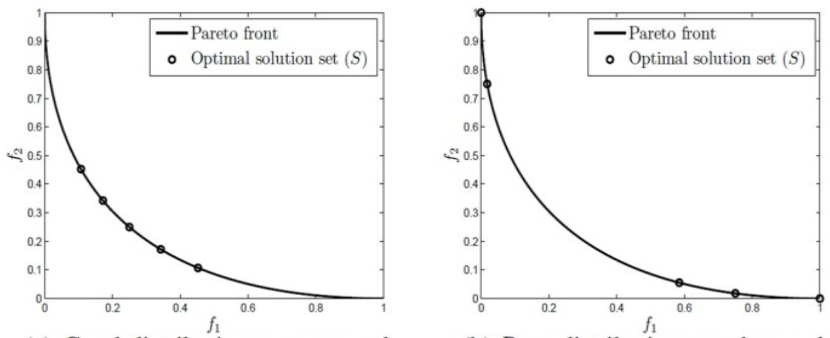


Figure 5.2: Diversity metrics for two components (spread and distribution) (Lee, 2004)

3. Epsilon ($\in$): A binary indicator that considers all the objectives to provide a factor by which an approximation set is worse than another. Let V and U be two approximation sets, then $\in(V,U)$ is the minimum factor $\in$. For any solution in set *V* there is at least one solution in *U* that is not considered as worse by a factor of $\in$ considering all the objects (Hamdy, 2016). This metric is used to estimate the quality of the archived solution set by each algorithm (Acampora, 2014). $\in$ is used is used to test the convergence of MOOA·

These three metrics are chosen in this chapter as in ((Coello, 2004); (Guzek, 2012); (Guzek, 2014)), in which, spread measures the *SP* of the obtained solution, epsilon measures the convergence of the algorithm, while *IGD* combines both of these components (Guzek, 2012).

In this chapter, we choose the most used multi-objective optimization algorithms by consensus the specialists in the field of MOO to evaluate the performance of MOSFP algorithm. These algorithms are OMOPSO, NSGA-II, and SPEA2. Among swarm intelligence approaches, OMOPSO is the most popular approach because OMOPSO has very good quality of results and high convergence and performance (Hamdan, 2017). For evolutionary algorithms, SPEA2 and NSGA-II are the most commonly used optimization methods ((Gharari, 2016), (Acampora, 2014)). Therefore, these algorithms are chosen in this chapter to compare their results with the proposed MOSFP using the same environment, hardware, and platform for each algorithm. The full procedure of these algorithms are summarized in Section 2.3 of Chapter 2.

To compare MOSFP algorithm with the aforementioned algorithms, qualitative and quantitative tests are performed. For the qualitative test, the quality of achieved Pareto fronts of the algorithms are compared, while for the quantitative test, $\in$, *SP*, and *IGD* metrics are adopted. For these purposes, two benchmark suites of Multi-Objectives Optimization Problems (MOOPs) called Zitzler-Deb-Thiele (ZDT), and Walking-Fish-

Group (WFG) suites are used in this chapter. These test suites (MOOPs) are summarized as follows:

Zitzler-Deb-Thiele (ZDT) Test Suite: Zitzler et al. (Huband, 2006) proposed this suite to be a standard test suite for evaluating the MOOAs. ZDT is widely used as benchmark functions for Swarm Intelligence Algorithms (SIA) and Evolutionary algorithms (EA). Table 5.1 analyses the characteristics of this suite. From the table, ZDT has a set of test functions that are difference in their geometry. For example, ZDT1 has a convex geometry, while ZDT2 and ZDT6 have concave geometry. ZDT3 is disconnected on both *S* and Pareto front. This suite shares many characteristics, including, how multimodality can produce many-to-one *P* as in (ZDT6), and a disconnected *P* such as in (ZDT3). These functions have only one parameter, which is used in their calculation. ZDT suite has many advantages, including, used in estimating a wide variety of MOOAs. Also, ZDT is well defined in the literature, which facilitate the comparisons with new MOOAs. *P* of each benchmark function in this suite is easy to understand and apply (Huband, 2005). For these reasons, ZDT is chosen to evaluate our proposed algorithm. Details mathematical formulation of Zitzler-Deb-Thiele (ZDT) test suite is shown in Part 1 of Appendix A.

Walking-Fish-Group (WFG) Test Suite: This suite is proposed by (Deb, 2002) and consists of nine problems. Table 5.2 outlines the characteristics of this suite (Huband, 2006). WFG1 utilizes different variables by using dissimilar weights in its parameters weighted sum reduction. WFG suite has unimodel and separable functions such as WFG1 and WFG7. In addition, it has non-separable reduction problems such as WFG6 and WFG9, which their models are more difficult than both of WFG3 and WFG2. A multimodality problem is also applicable in WFG suite such as WFG4 as it has a large “hill sizes” and its models are more difficult than that of WFG9. WFG5 is highly deceptive benchmark function and can be more difficult than that of WFG9. In WFG7, the parameters are dependent on the distance and position-related parameters. WFG9 is more complex than WFG7 and its distance-related parameters depend on other distance and position-related parameters. WFG8 is different and more complex than WFG9 wherein its distance-related parameters depend on other distance and position-related parameters.

Table 5.1: Analysis of ZDT benchmark functions (Huband, 2006)

Name	ZDT1		ZDT2		ZDT3		ZDT6	
Objective	f_1	f_2	f_1	f_2	f_1	f_2	f_1	f_2
R3: # Parameters	1	✓	1	✓	1	✓	1	✓
F2: Separability	S	S	S	S	S	S	S	S
F5: Modality	U	U	U	U	U	M	U	U
R1: No Extremal	×		×		×		×	
R2: No Medial	✓		✓		✓		✓	
R5: Diss. Domains	×		×		×		×	
R6: Diss. Ranges	×		×		✓		✓	
R7: Optima known	✓		✓		✓		✓	
F1: Geometry	convex		concave		disconnected		concave	
F3: Bias	-		-		-		+	
F4a: Pareto Many-to-one	-		-		-		+	
F4b: Flat Regions	-		-		-		-	

The WFG benchmark test suite is a comprehensive problem consisting a wide range of different problems. The problems include a mixed shape of Pareto front problems, deceptive problems, a truly degenerate problems, non-separable problems, problems scalable in the number of position related parameters, and problems with dependencies between both of position-related parameters and distance-related parameters. This suite is one of the most commonly used benchmark suites to provide truer means of evaluating the efficiency and performance of MOOAs (Chase, 2009). Therefore, WFG test suite is chosen to evaluate our proposed approach and the detailed of the WFG test suites is defined in Part 2 of Appendix A.

Table 5.2: Analysis of WFG benchmark functions (Huband, 2006)

Name	Objective	F2: Separability	Modality	Bias	Geometry
WFG1	$f_{1:M}$	S	U	Polynomial, flat	Convex, Mixed
WFG2	$f_{1:M-1}$ f_M	NS NS	U M	-	Convex, disconnected
WFG3	$f_{1:M}$	NS	U	-	Linear, degenerate
WFG4	$f_{1:M}$	S	M	-	Concave
WFG5	$f_{1:M}$	S	D	-	Concave
WFG6	$f_{1:M}$	NS	U	-	Concave
WFG7	$f_{1:M}$	S	U	Parameter dependent	Concave
WFG8	$f_{1:M}$	NS	U	Parameter dependent	Concave
WFG9	$f_{1:M}$	NS	M, D	Parameter dependent	Concave

The abbreviation in Table 5.1 and Table 5.2 are "M" for multimodal, "NS" for non-separable, "S" for separable, "D" for deceptive, "U" for unimodal, the symbols "✓" and "+" indicate whether a given recommendation is adhered to, whereas the symbols "–" and "+" indicate the absence or presence of some features.

5.5 Experimental Test and Results

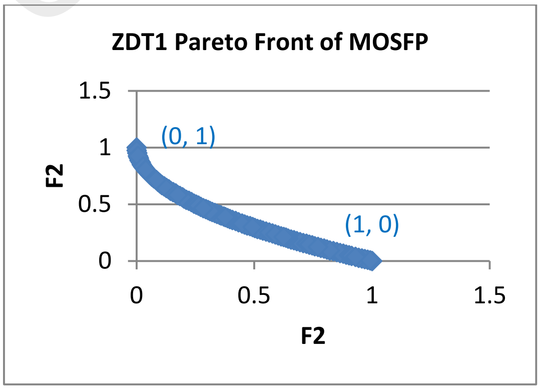
MOSFP algorithm has been implemented in Java language using the JMetal tool and integrated the implementation into the platform of NetBeans IDE 8.1. All the evaluation tests are carried out on a desktop with 3 GB RAM with Intel dual-core CPU T3200 running Windows 7. This chapter used standard parameters and settings as recommended in (Chase, 2009) to compare between our MOSFP and the other algorithms. Table 5.3 summarizes these parameters. These parameters and settings of all algorithms are kept in all evaluation tests presented next. All the algorithms are evaluated by reporting the results achieved from executing 100 independent runs of each algorithm for each benchmark function. In each benchmark function, the total limit of objective function evaluations is set to 5000. Three evaluation metrics namely, *IGD,* $\in$ and *SP* are evaluated, which the worst, average, best and median of each metric based on five-thousand runs are presented.

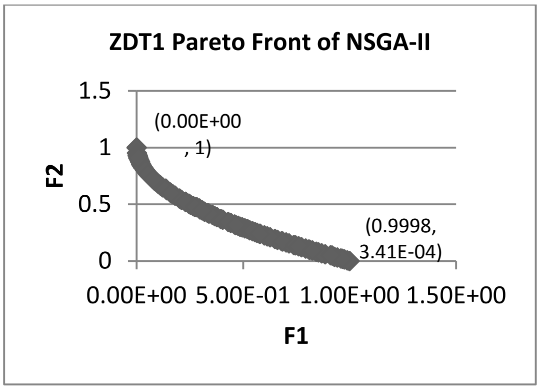
Table 5.3: Parameters and settings of the algorithms

Parameters	MOSFP	OMOPSO	NSGA-II	SPEA2
Population size	100	100	100	100
Archive size	(winner) 100	100	(Elite) 100	100
Mating pool size	-	-	-	100
Maximum generation	5000	5000	5000	5000
Crossover probability	-	-	0.9	0.9
Mutation probability	1/d where d is the variable code size			

A) ZDT Test Suites

Overall, the OMOPSO algorithm obtained the best average of *IGD*, *SP*, and ∈ measurements for ZDT test suites, while the proposed MOSFP is in the second followed by NSGA-II and SPEA2 as outlined in Table 5.4. However, for ZDT3 test function, the proposed MOSFP attained the best average ∈ among all algorithms. Despite the rank, the proposed MOSFP has shown a comparable performance with the OMOPSO algorithm as their difference of average ∈, *SP*, and *IGD* measurements are minimal. Both MOSFP and the OMOPSO obtained extreme superior points of the Pareto front such as (1, 0) and (0, 1) for ZDT1 and ZDT2 functions as depicted in Figures 5.3 and 5.4. Also, both MOSFP and the OMOPSO preserved the Pareto front extreme superior points of (0, 1) for ZDT3 function and (1, 0) for ZDT6 function, as shown in Figures 5.5 and 5.6. In opposite, SPEA2 and NSGA-II failed to obtain these optimal extreme values of the Pareto front for all ZDT functions.





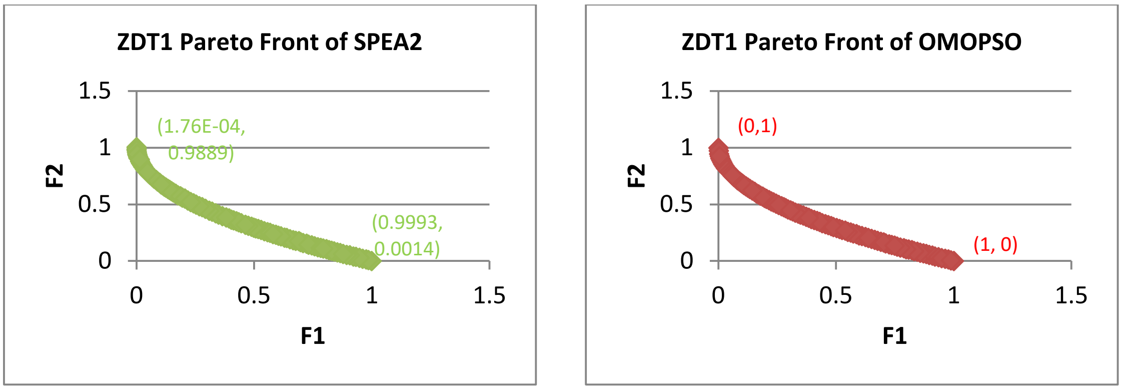


Figure 5.3: ZDT1 Pareto fronts achieved by all the algorithms

Table 5.4: Comparison of results between OMOPSO, NSGA-II, SPEA2, and our algorithm (denoted by MOSFP) in term of IGD, Spread, Epsilon and for ZDT problems

	Epsilon				Spread				IGD			
	Best	Worst	Aver.	Med.	Best	Worst	Aver.	Med.	Best	Worst	Aver.	Med.
ZDT1												
MOSFP	5.01E-03	5.57E-03	5.27E-03	5.25E-03	2.76E-02	9.85E-02	6.23E-02	6.22E-02	3.44E-05	3.61E-05	3.51E-05	3.51E-05
OMOPSO	4.92E-03	5.40E-03	**5.15E-03**	5.15E-03	3.29E-02	8.74E-02	**5.54E-02**	5.52E-02	3.39E-05	3.60E-05	**3.48E-05**	3.48E-05
NSGA-II	1.11E-01	2.67E-01	1.61E-01	1.51E-01	5.84E-01	8.17E-01	6.80E-01	6.87E-01	7.65E-04	1.92E-03	1.18E-03	1.14E-03
SPEA2	1.60E-01	4.58E-01	2.82E-01	2.74E-01	8.92E-01	6.11E-01	7.43E-01	7.35E-01	1.04E-03	2.71E-03	1.76E-03	1.72E-03
ZDT2												
MOSFP	5.018E-03	5.65E-03	5.27E-03	5.28E-03	2.75E-02	9.95E-02	5.62E-02	5.41E-02	3.41E-05	3.60E-05	3.49E-05	3.49E-05
OMOPSO	4.93E-03	5.32E-03	**5.13E-03**	5.14E-03	2.45E-02	8.30E-02	**5.53E-02**	5.56E-02	3.43E-05	3.56E-05	**3.48E-05**	3.47E-05
NSGA-II	3.46E-01	1.457	6.32E-01	5.88E-01	6.38E-01	1.011	8.33E-01	8.32E-01	1.54E-03	9.44E-03	3.10E-03	2.75E-03
SPEA2	5.21E-01	1.775	1.107	1.051	8.14E-01	1.0162	9.38E-01	9.51E-01	2.50E-03	1.16E-02	6.34E-03	5.31E-03
ZDT3												
MOSFP	3.37E-03	5.28E-03	**4.13E-03**	4.11E-03	6.99E-01	7.02E-01	7.01E-01	7.01E-01	2.78E-05	3.08E-05	2.91E-05	2.91E-05
OMOPSO	3.54E-03	5.25E-03	4.18E-03	4.06E-03	6.99E-01	7.01E-01	**7.00E-01**	7.00E-01	2.74E-05	3.04E-05	**2.86E-05**	2.86E-05
NSGA-II	1.51E-01	3.36E-01	2.10E-01	2.01E-01	7.82E-01	9.39E-01	8.63E-01	8.62E-01	3.96E-04	1.40E-03	8.73E-04	8.87E-04
SPEA2	1.91E-01	6.06E-01	2.92E-01	2.86E-01	7.88E-01	9.37E-01	8.64E-01	8.62E-01	7.43E-04	1.98E-03	1.30E-03	1.31E-03
ZDT6												
MOSFP	5.32E-03	6.53E-03	5.90E-03	5.91E-03	1.33E-01	1.81E-01	1.55E-01	1.55E-01	3.21E-	3.35E-	3.27E-	3.27E-

									05	05	05	05
OMOPSO	4.73E-03	9.00E-03	**5.59E-03**	5.42E-03	5.59E-02	2.53E-01	**1.13E-01**	1.08E-01	3.09E-05	3.41E-05	**3.19E-05**	3.17E-05
NSGA-II	2.34E-02	3.53E-02	2.71E-02	2.65E-02	6.04E-02	1.36	7.41E-01	9.53E-01	4.67E-05	1.37E-03	2.03E-04	1.60E-04
SPEA2	1.85E-02	3.84E-01	6.98E-02	6.09E-02	9.03E-01	1.436	1.332	1.345	1.99E-04	2.29E-04	2.04E-04	2.03E-04

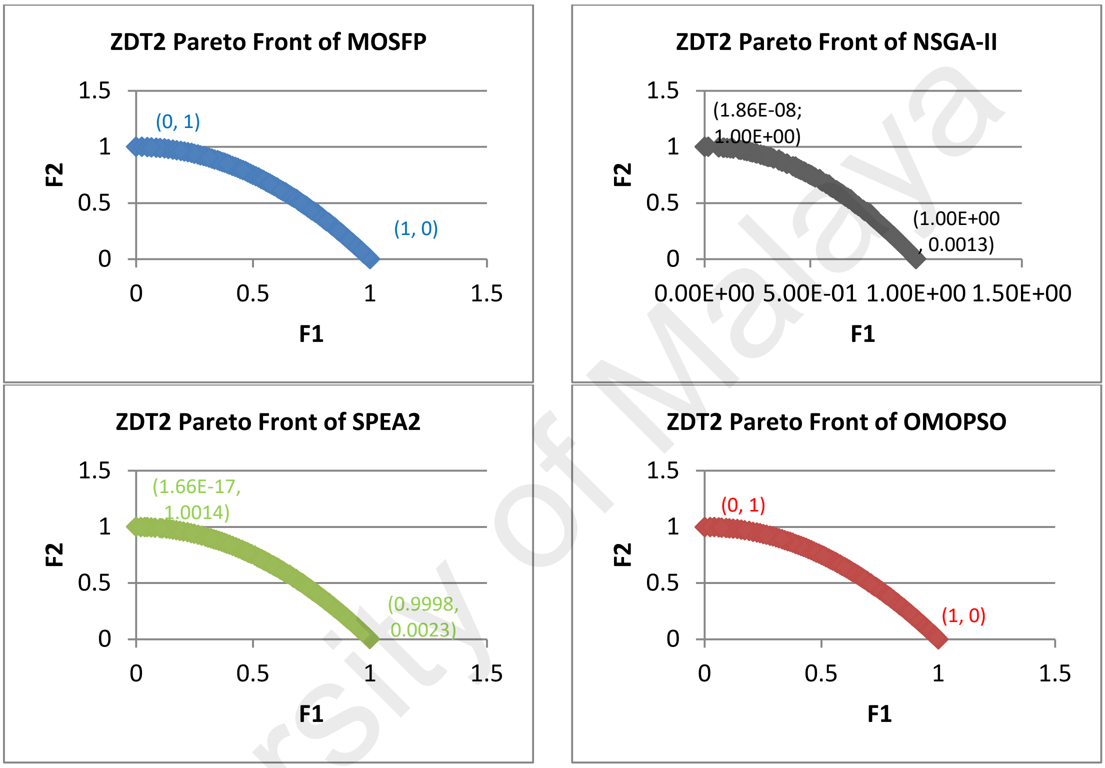
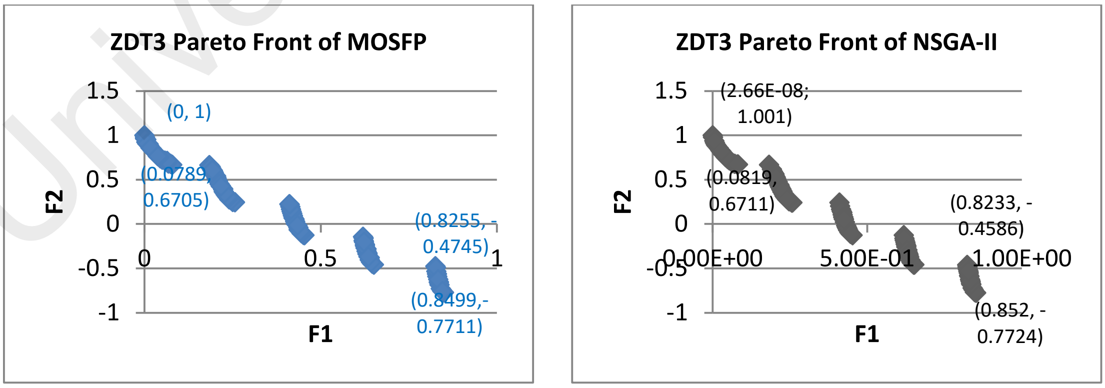


Figure 5.4: ZDT2 Pareto fronts achieved by all the algorithms



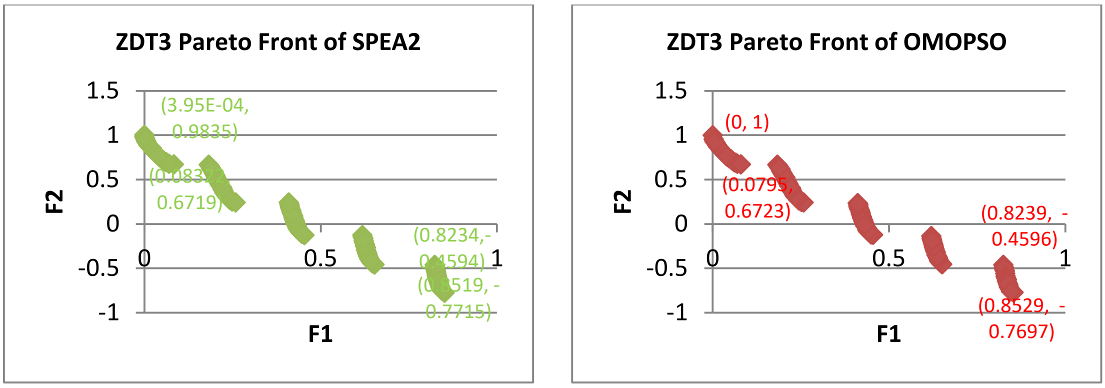


Figure 5.5: ZDT3 Pareto fronts achieved by all the algorithms

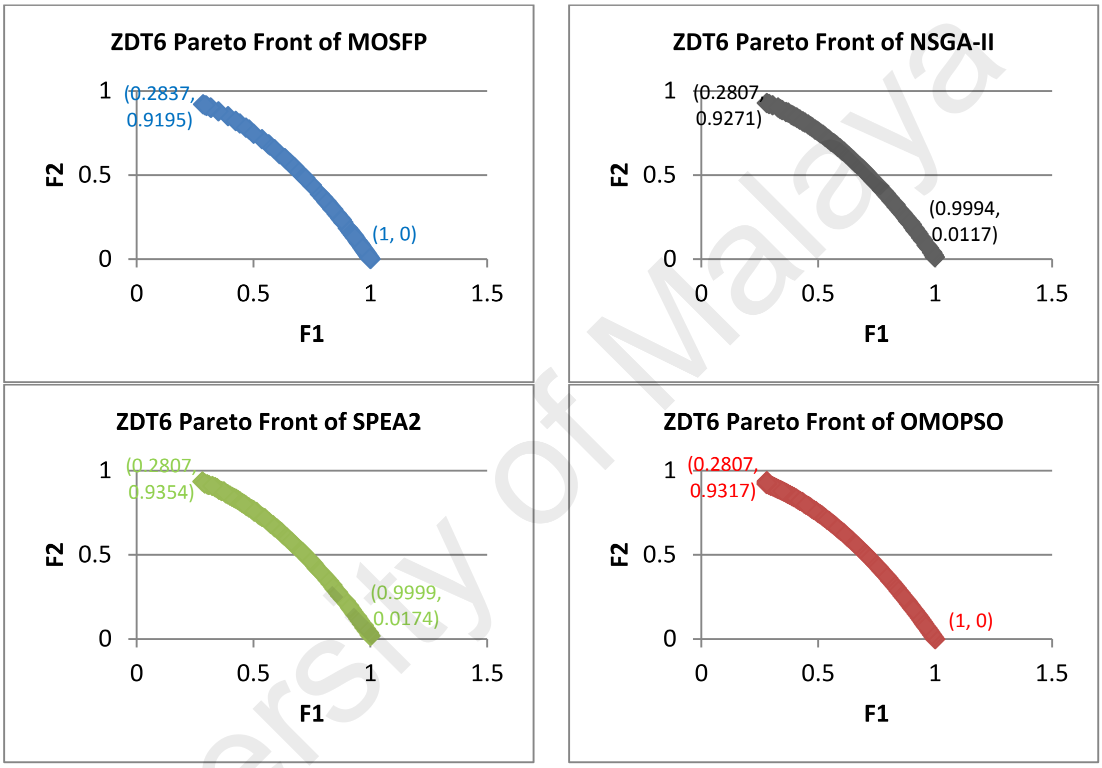


Figure 5.6: ZDT6 Pareto fronts achieved by all the algorithms

B) WFG Test Suites:

Table 5.5: Comparison of results between OMOPSO, NSGA-II, SPEA2, and our algorithm (denoted by MOSFP) in term of IGD, Spread, Epsilon and for WFG problems

	Epsilon				Spread				IGD			
	Best	Worst	Aver.	Med.	Best	Worst	Aver.	Med.	Best	Worst	Aver.	Med.
WFG1												
MOSFP	8.59E-01	1.199	1.092	1.094	8.01E-01	1.057	9.31E-01	9.31E-01	2.72E-03	4.93E-03	4.33E-03	4.37E-03
OMOPSO	7.68E-02	1.022	**1.61E-01**	1.45E-01	7.41E-01	1.238	**8.42E-01**	8.33E-01	3.04E-05	3.49E-03	**1.05E-04**	4.81E-05

NSGA-II	1.441	2.020	1.740	1.746	8.63E-01	1.030	9.37E-01	9.33E-01	3.93E-03	6.17E-03	5.06E-03	5.07E-03
SPEA2	1.562	2.227	1.923	1.922	8.08E-01	1.266	1.080	1.070	4.46E-03	7.02E-03	5.77E-03	5.73E-03
WFG2												
MOSFP	1.14E-02	1.92E-02	1.42E-02	1.39E-02	7.57E-01	7.96E-01	7.77E-01	7.74E-01	5.07E-05	7.37E-05	5.91E-05	5.90E-05
OMOPSO	7.48E-03	1.04E-02	**9.01E-03**	8.94E-03	7.56E-01	7.59E-01	**7.57E-01**	7.57E-01	3.81E-05	4.22E-05	**4.01E-05**	4.00E-05
NSGA-II	1.59E-02	8.15E-01	4.83E-01	7.99E-01	7.86E-01	1.006	8.59E-01	8.50E-01	6.21E-05	1.77E-03	1.06E-03	1.73E-03
SPEA2	1.64E-02	8.18E-01	5.03E-01	8.01E-01	8.01E-01	1.076	9.27E-01	9.22E-01	6.85E-05	1.77E-03	1.11E-03	1.73E-03
WFG3												
MOSFP	1.55E-02	1.81E-02	1.81E-02	1.66E-02	2.40E-02	6.63E-02	4.22E-02	4.29E-02	4.48E-05	4.81E-05	4.62E-05	4.61E-05
OMOPSO	1.39E-02	1.51E-02	**1.43E-02**	1.43E-02	1.54E-02	4.85E-02	**2.95E-02**	2.89E-02	4.37E-05	4.64E-05	**4.49E-05**	4.5E-05
NSGA-II	3.37E-02	7.05E-02	4.60E-02	4.43E-02	2.62E-01	4.10E-01	3.44E-01	3.44E-01	7.06E-05	1.35E-04	8.55E-05	8.34E-05
SPEA2	2.74E-02	1.12E-01	4.87E-02	4.44E-02	1.70E-01	3.09E-01	2.16E-01	2.16E-01	5.99E-05	1.32E-04	8.58E-05	8.60E-05
WFG4												
MOSFP	2.87E-02	1.91E-01	4.71E-02	4.16E-02	3.10E-01	4.38E-01	3.72E-01	3.74E-01	1.13E-04	1.55E-04	1.25E-04	1.24E-04
OMOPSO	1.25E-02	4.47E-02	**1.90E-02**	1.52E-02	9.02E-02	2.43E-01	**1.38E-01**	1.27E-01	8.13E-05	1.14E-04	**8.86E-05**	8.51E-05
NSGA-II	6.26E-02	1.55E-01	1.09E-01	1.10E-01	4.55E-01	6.69E-01	5.56E-01	5.53E-01	2.90E-04	4.30E-04	3.60E-04	3.54E-04
SPEA2	3.01E-02	4.17E-1	1.04E-1	7.02E-2	2.46E-01	3.51E-01	3.00E-01	3.02E-01	1.01E-04	1.74E-04	1.16E-04	1.14E-04
WFG5												
MOSFP	4.99E-02	6.52E-02	**5.61E-02**	5.59E-02	9.78E-02	1.76E-01	1.40E-01	1.40E-01	2.66E-04	4.24E-04	**3.00E-04**	2.95E-04
OMOPSO	4.93E-02	9.35E-02	5.73E-02	5.71E-02	9.05E-02	3.47E-01	**1.19E-01**	1.18E-01	2.57E-04	6.84E-04	3.05E-04	2.95E-04
NSGA-II	5.30E-02	1.33E-01	1.04E-01	1.04E-01	3.15E-01	4.55E-01	3.88E-01	3.85E-01	3.14E-04	1.15E-03	8.11E-04	8.26E-04
SPEA2	9.73E-02	1.50E-01	1.19E-01	1.17E-01	2.29E-01	3.34E-01	2.82E-01	2.81E-01	7.32E-04	1.30E-03	9.75E-04	947E-04
WFG6												

MOSFP	1.55E-02	2.08E-02	1.70E-02	1.69E-02	8.58E-02	1.52E-01	1.20E-01	1.22E-01	4.98E-05	5.46E-05	5.21E-05	5.21E-05
OMOPSO	1.29E-02	1.45E-02	**1.36E-02**	1.36E-02	7.70E-02	1.42E-01	**1.05E-01**	1.06E-01	4.71E-05	5.16E-05	**4.90E-05**	4.88E-05
NSGA-II	3.49E-02	1.30E-01	6.10E-02	5.75E-02	3.19E-01	5.17E-01	3.92E-01	3.86E-01	7.59E-05	5.32E-04	1.64E-04	1.45E-04
SPEA2	3.41E-02	2.13E-01	8.74E-02	7.79E-02	2.52E-01	6.56E-01	3.61E-01	3.44E-01	7.17E-05	4.08E-04	1.76E-04	1.67E-04
WFG7												
MOSFP	1.47E-02	1.89E-02	1.57E-02	1.56E-02	9.38E-02	1.44E-01	1.20E-01	1.20E-01	4.94E-05	5.38E-05	5.14E-05	5.14E-05
OMOPSO	1.29E-02	1.45E-02	**1.36E-02**	1.36E-02	8.68E-02	1.38E-01	**1.10E-01**	1.09E-01	4.71E-05	5.10E-05	**4.93E-05**	4.93E-05
NSGA-II	2.98E-02	7.71E-02	4.23E-02	3.98E-02	3.13E-01	4.48E-01	3.82E-01	3.79E-01	6.75E-05	8.45E-05	7.47E-05	7.42E-05
SPEA2	2.85E-02	3.34E-01	8.82E-02	7.23E-02	2.41E-01	4.41E-01	2.92E-01	2.89E-01	6.00E-05	3.34E-04	7.35E-05	6.85E-05
WFG8												
MOSFP	3.05E-01	4.82E-01	**3.66E-01**	3.66E-01	6.30E-01	1.071	7.54E-01	7.34E-01	1.73E-03	2.18E-03	2.02E-03	2.02E-03
OMOPSO	5.31E-02	4.89E-01	3.78E-01	4.87E-01	4.05E-01	7.83E-01	**5.06E-01**	4.89E-01	2.33E-04	2.11E-03	**1.79E-03**	2.08E-03
NSGA-II	1.94E-01	7.27E-01	4.62E-01	4.97E-01	6.60E-01	9.34E-01	7.95E-01	7.94E-01	1.10E-03	2.61E-03	2.26E-03	2.37E-03
SPEA2	3.99E-01	8.15E-01	6.14E-01	6.00E-01	6.45E-01	9.09E-01	7.67E-01	7.58E-01	2.04E-03	2.68E-03	2.44E-03	2.46E-03
WFG9												
MOSFP	3.59E-02	1.05E-01	9.26E-02	9.28E-02	1.37E-01	2.27E-01	1.84E-01	1.83E-01	8.15E-05	9.25E-05	8.69E-05	8.70E-05
OMOPSO	1.53E-02	8.35E-02	**7.58E-02**	7.81E-02	6.31E-02	1.20E-01	**9.51E-02**	9.56E-02	5.32E-05	5.77E-05	**5.50E-05**	5.50E-05
NSGA-II	8.76E-02	1.76E-01	1.09E-01	1.04E-01	3.05E-01	4.39E-01	3.63E-01	3.65E-01	8.17E-05	1.19E-04	9.87E-05	9.81E-05
SPEA2	8.86E-02	2.44E-01	1.30E-01	1.21E-01	2.29E-01	3.25E-01	2.86E-01	2.86E-01	7.76E-05	1.54E-04	9.38E-05	9.32E-05

In this experiment, the OMOPSO has the best overall performance among all of the algorithms. However, the proposed MOSFP outperformed the OMOPSO, SPEA2, and NSGA-II algorithms in solving WFG8 and WFG5 test functions in terms of measurement and the best average of *IGD* for WFG5 function. Overall, based on the prior Figures of WFG functions, the proposed MOSFP demonstrates a high ability in obtaining the optimal extreme superior points of the true Pareto front, i.e., (2, 0) and (0, 4) for WFG test functions. The proposed MOSFP obtained a close approximation of extreme superior points to the optimal point of the true Pareto front in solving WFG9, WFG7, WFG6, WFG5, WFG4, WFG3, and WFG2 test functions. For WFG8 function, all of the algorithms obtained inaccurate approximation of extreme superior points of true Pareto front. However, both the MOSFP and OMOPSO algorithms exhibit a better spread than SPEA2 and NSGA-II when their Pareto front is distributed into three parts when compared to SPEA2 and NSGA-II that are distributed into two parts. The same can also be observed in WFG1 test function when OMOPSO and MOSFP are more spread than SPEA2 and NSGA-II.

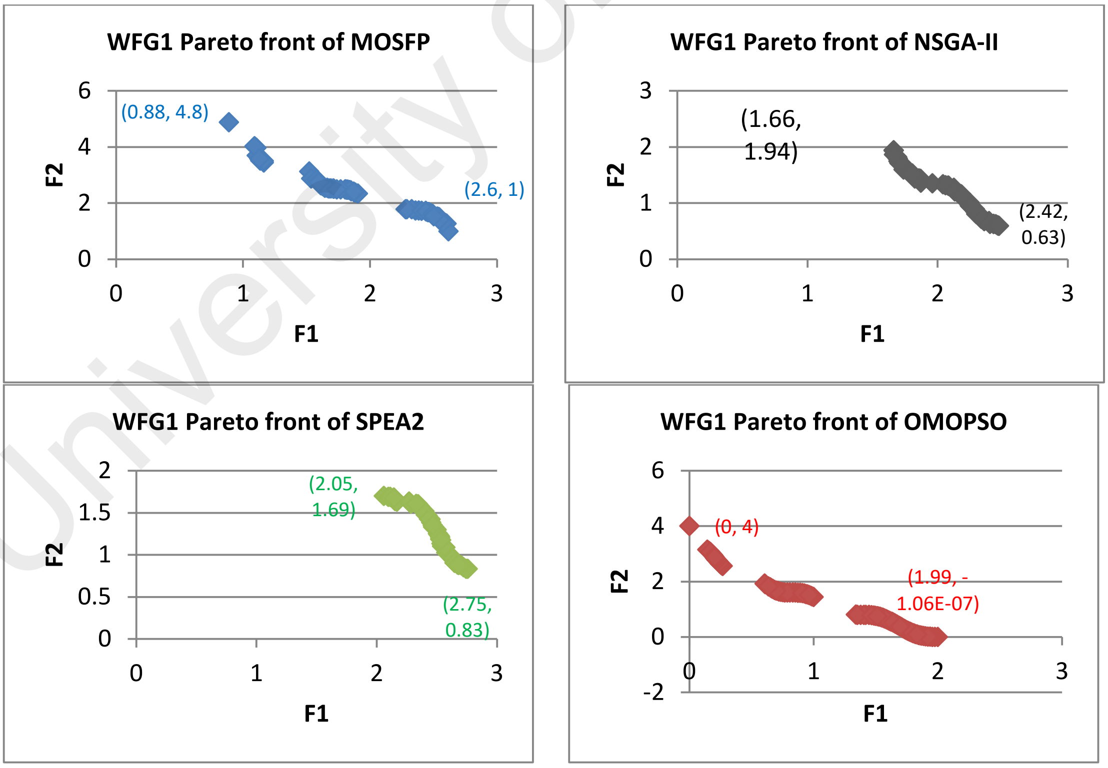


Figure 5.7: WFG1 Pareto fronts achieved by all the algorithms

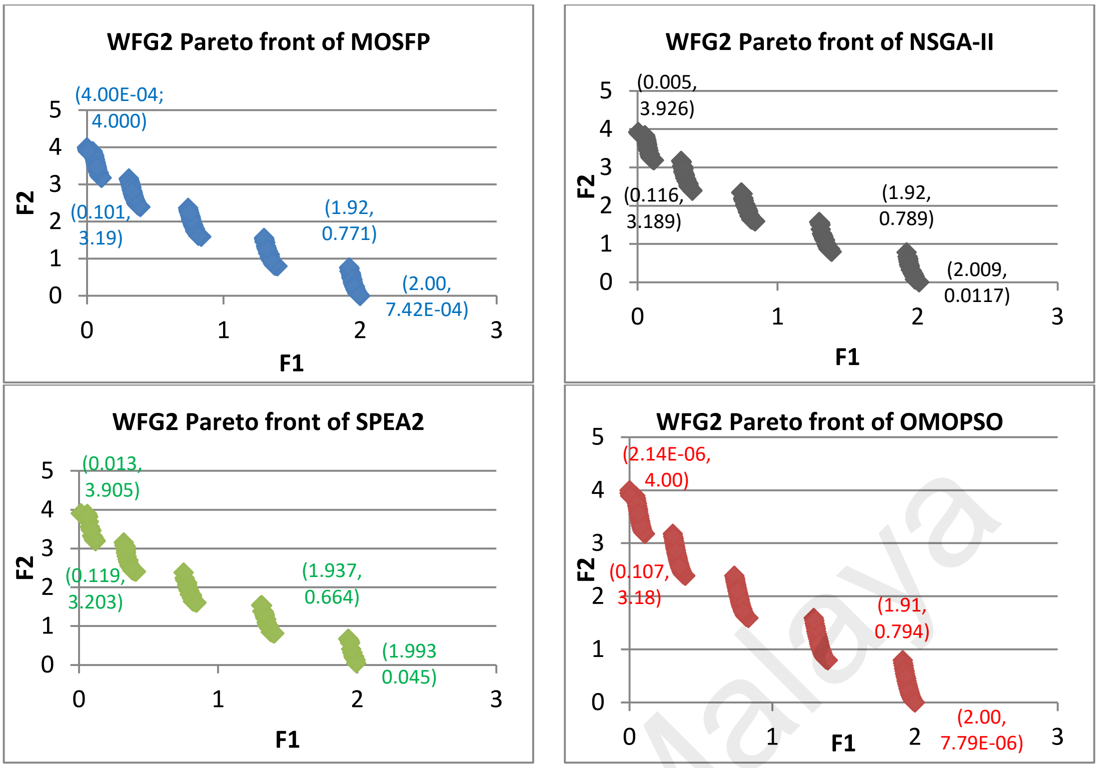


Figure 5.8: WFG2 Pareto fronts achieved by all the algorithms

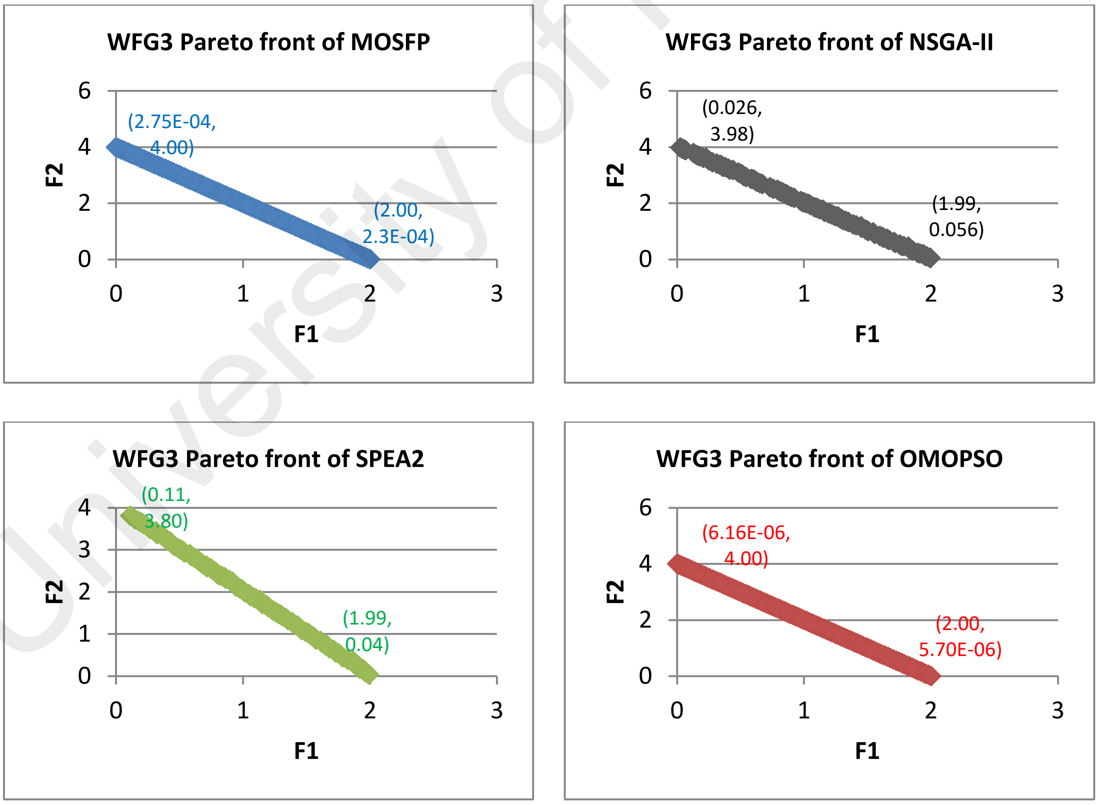


Figure 5.9: WFG3 Pareto fronts achieved by all the algorithms

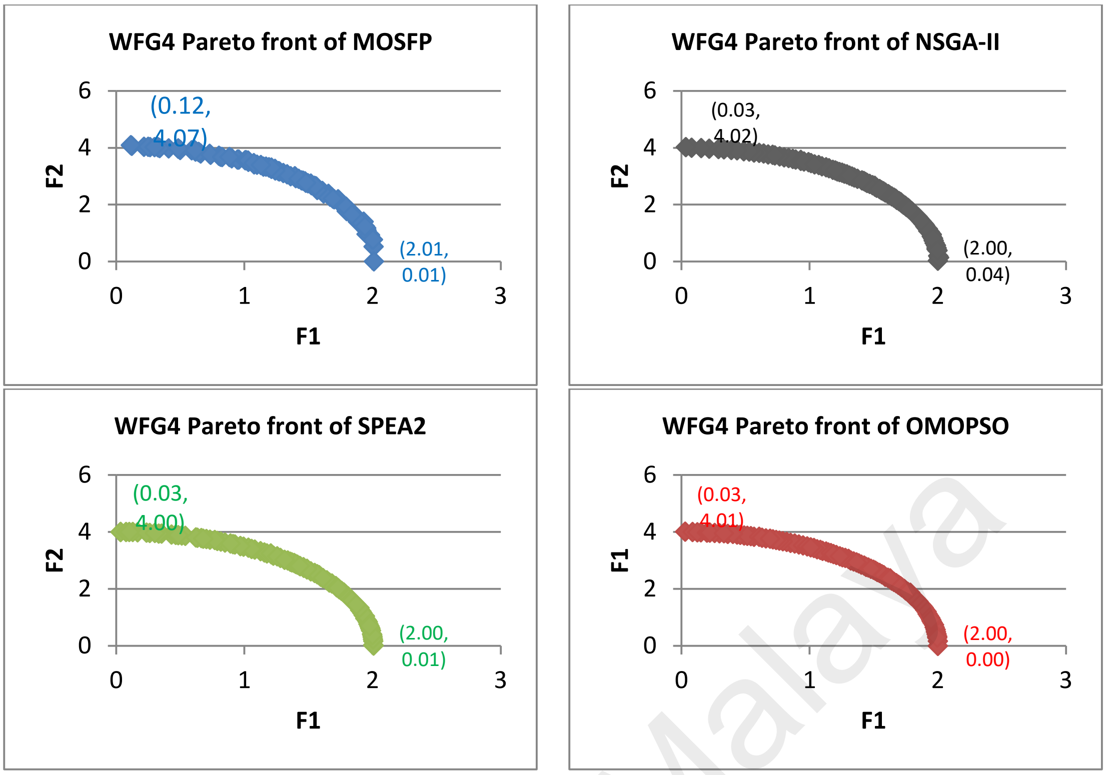


Figure 5.10: WFG4 Pareto fronts achieved by all the algorithms

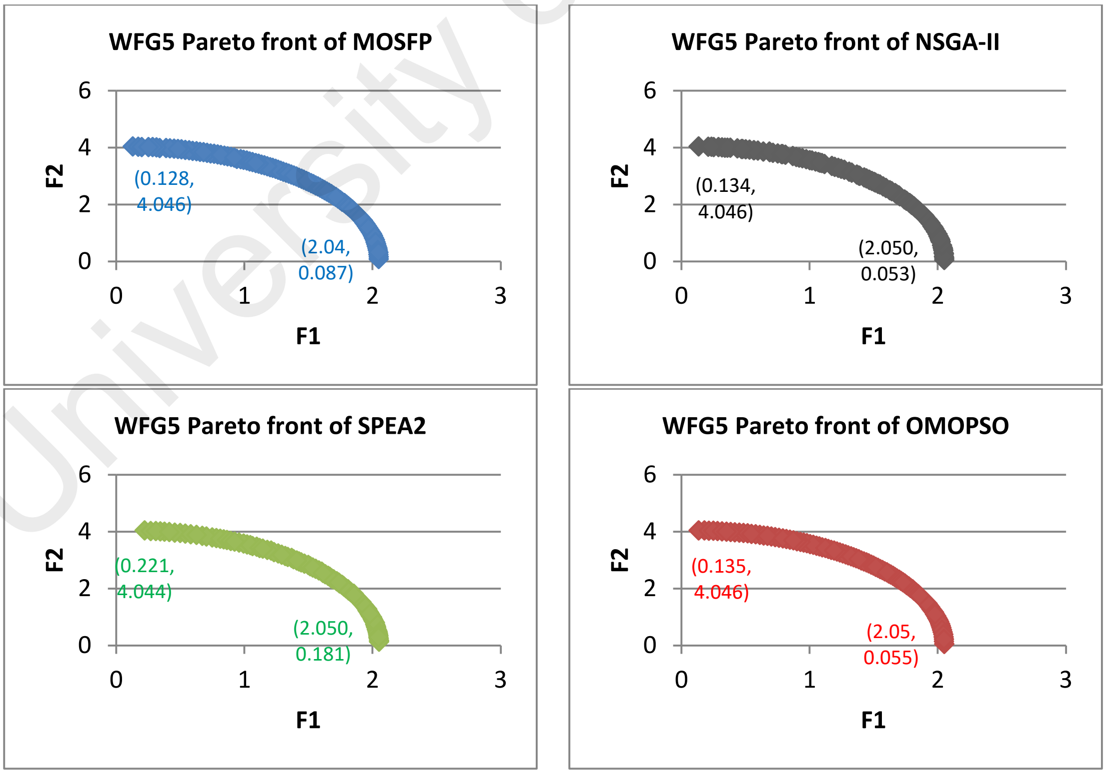


Figure 5.11: WFG5 Pareto fronts achieved by all the algorithms

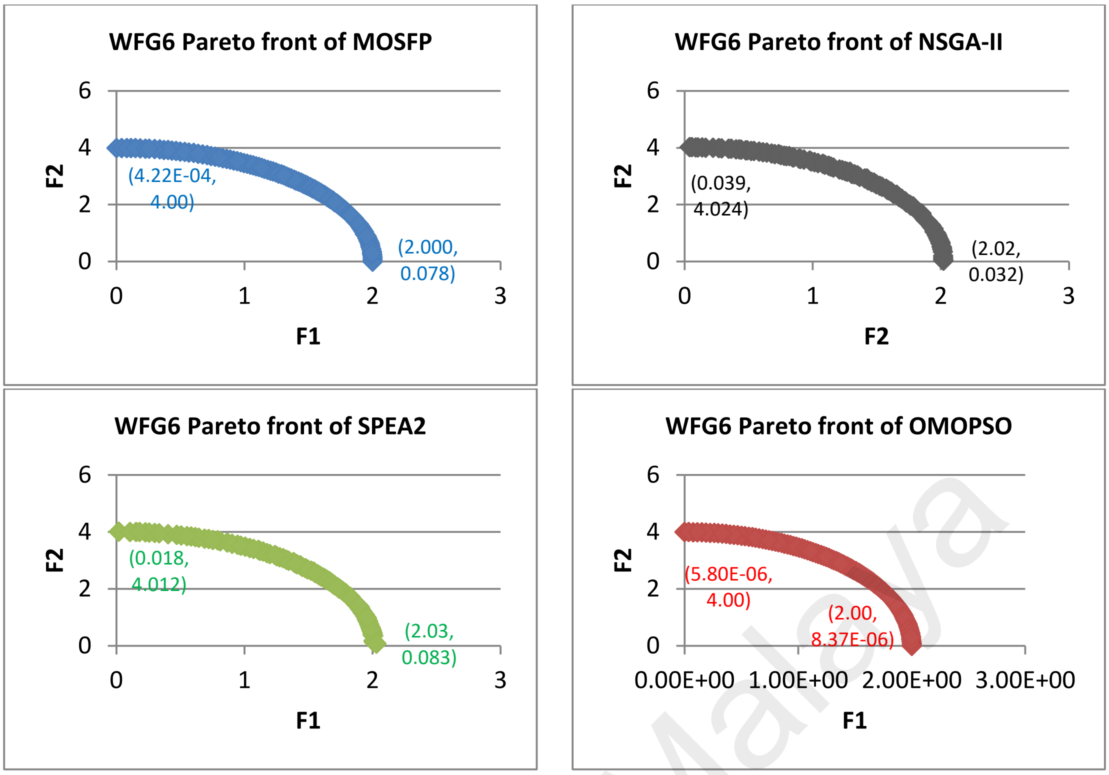


Figure 5.12: WFG6 Pareto fronts achieved by all the algorithms

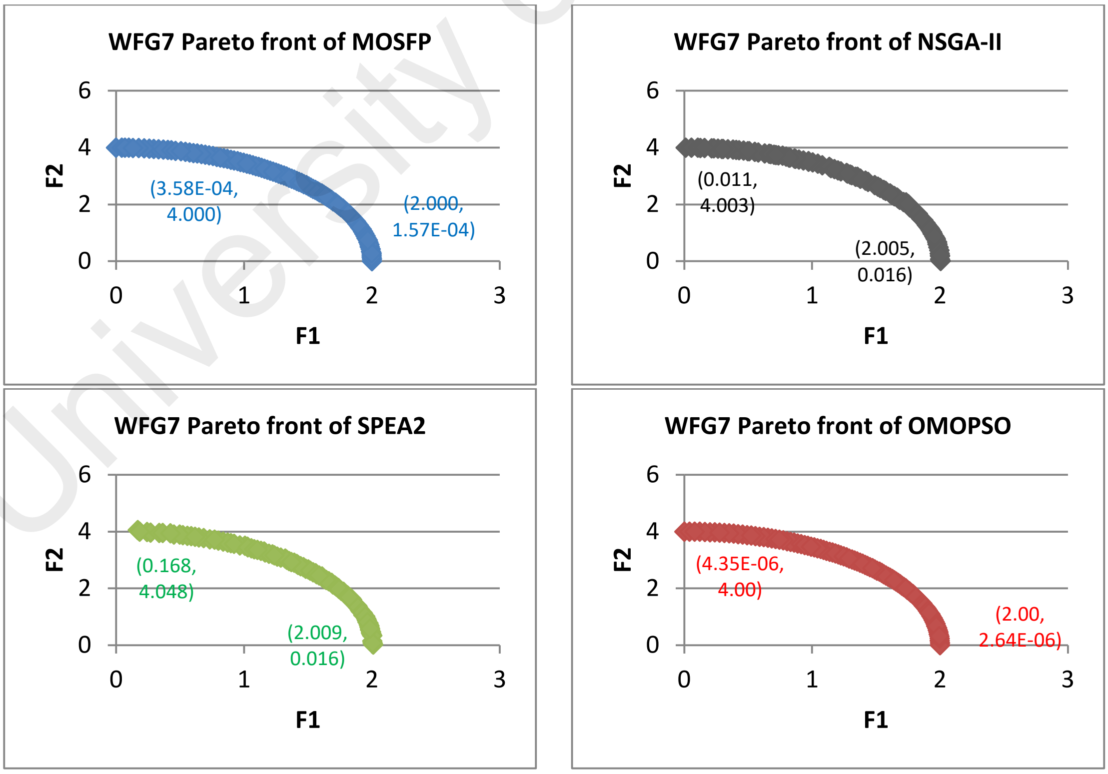


Figure 5.13: WFG7 Pareto fronts achieved by all the algorithms

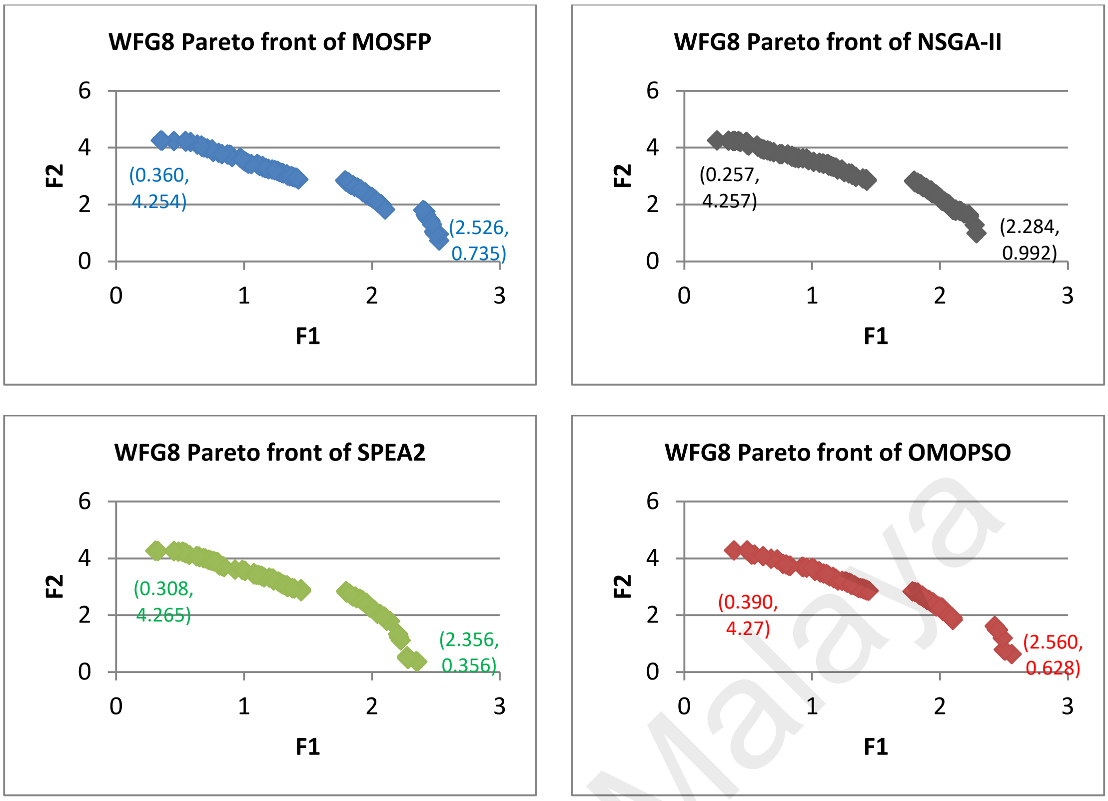


Figure 5.14: WFG8 Pareto fronts achieved by all the algorithms

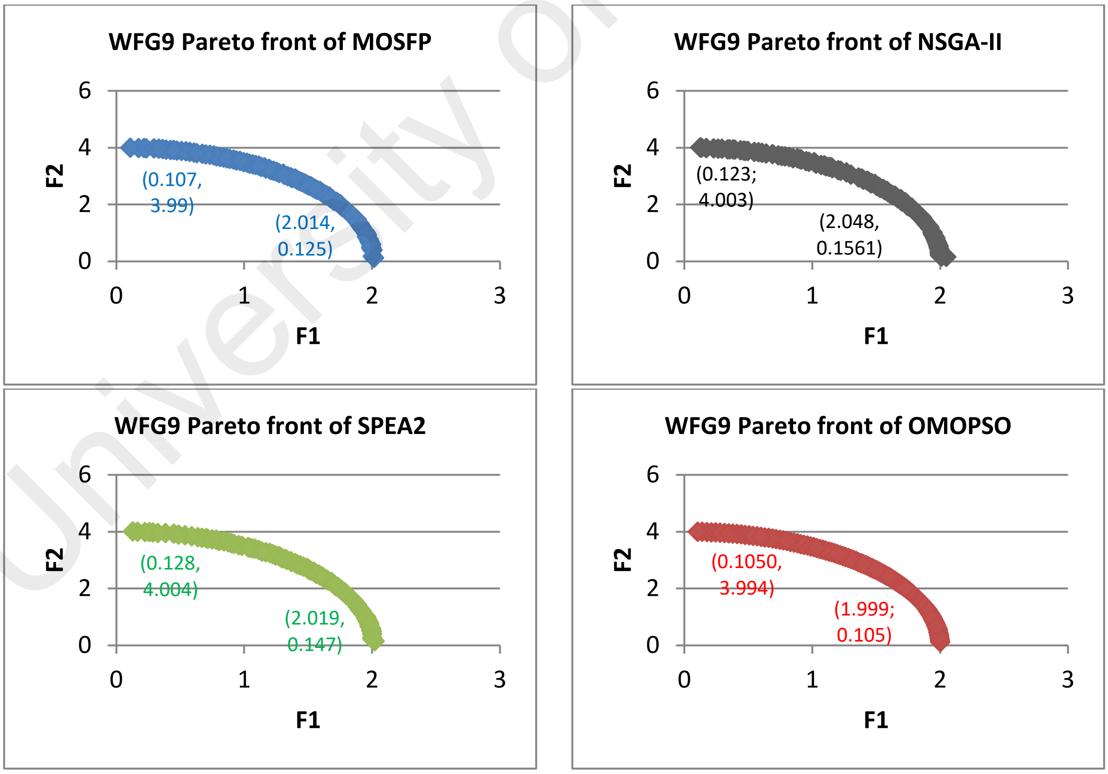


Figure 5.15: WFG9 Pareto fronts achieved by all the algorithms

5.6 Chapter Discussion

In this chapter, MOSFP, NSGA-II, OMOPSO, and SPEA2 were tested on WFG and ZDT benchmark functions. Three standard metrics have been used for this purpose, namely Epsilon, Inverted Generational Distance and Spread. For each algorithm, the maximum generation for each benchmark function is set to 5000. The experimental test is repeated 100 times for each objective to ensure the quality of the results.

Overall, our algorithm (denoted by MOSFP) outperformed both SPEA2 and NSGA-II algorithms in all benchmark function suites. Furthermore, MOSFP algorithm attained a good approximation and a high amount of points related to the true Pareto front of all test suites. Additionally, MOSFP outperformed OMOPSO in solving the WFG5 problems, and achieved better solution sets than OMOPSO for true Pareto front of both WFG8 and ZDT3. The high-quality performance and efficiency of MOSFP was reflected on the metrics of IGD and $\in$ of WFG5, and $\in$ of both WFG8 and ZDT3. This proves that the MOSFP has a better convergence than OMOPSO approach to discover the search space domain. Particularly, MOSFP has a high-quality performance in solving test suites that include more than two objective functions such as WFG8 and WFG 5 and the very complex disconnected benchmark functions such as ZDT3.

Based on that, MOSFP algorithm has the potential in solving the problems that need an algorithm with a good convergence such as coverage issue in WSN. Finding the optimal coverage in WSN requires determining the optimal distribution of sensor nodes on topology area estimated by kilometers or hectares (Bara'a, 2015). Furthermore, MOSFP has the ability more than other algorithms in solving the real problems with more than two objective functions such as finding the optimal task allocation of stationary gas turbine (Buche, 2002), finding the optimal length and spacing of Yagi–Uda antenna design (Venkatarayalu, 2003), solving engineering applications (Chase, 2009) finding the optimal Quality of Services (QoS) of wireless, mobile and telecommunication networks ((Bara'a, 2015); (Ibdah, 2017); (Hamdan, 2015)) and finding the optimal products distribution through oil pipeline networks (Garcia, 2004).

5.7 Chapter Summary

Chapter 5 could be summarized in following points:

1. Multi-Objective Optimization (MOO) manages the tradeoffs between a set of conflicting objective functions that consists of maximization problems and minimization problems. Based on this management, the optimal solution can be determined.
2. Examples of Multi-Objective Optimization Problems (MOOPs) are Walking-Fish-Group (WFG) test suites and Zitzler-Deb-Thiele (ZDT) test suites. WFG and ZDT suites are the mostly used benchmark functions to test the performance of Multi-Objective Optimization Algorithms (MOOAs). These suites consist of minimization and maximization objective functions. The mathematical formulations of these suites are summarized in Appendix A.
3. Based on the MOO concept, many SOOAs have been extended to solve MOO problems. For examples, PSO algorithm is extended to various approaches such as OMOPSO, and GA is extended to NSGA-II.
4. SSO algorithm is Single Objective Optimization (SOO) heuristic-based algorithm that inspired by sperm motility to fertilize the egg. Based on the results in Chapter 4, this algorithm is suitable to solve various types of SOO problems, but requires enhancement to deal with MOO problems.
5. This chapter proposes a MOO version of SSO algorithm based on crowding, mutation operations, and archive operation. Crowding operator is used to enhance the spreading and distribution of non-dominated solutions along the Pareto front, while the mutation operation is used to increase the algorithm convergence. Moreover, the archive operator is used to increase the speed of the algorithm by fixing the size of non-dominance solutions.

6. The proposed algorithm is evaluated in two ways; qualitative and quantitative tests. In the qualitative test, the quality of the Pareto front of each algorithm has been tested for each algorithm. Figures 5.3 to 5.13 show these Pareto fronts for both ZDT and WFG test suites. For the quantitative test, three measurement metrics have been adopted such as Inverted Generational Distance, Epsilon, and Spread. Table 5.4 and Table 5.5 show these metrics for ZDT and WFG test suites respectively.
7. The experimental results of the proposed algorithm based on comparisons with NSGA-II, OMOPSO, and SPEA2 show that the proposed algorithm has a good ability to solve the problems with more than two objective functions such as WFG8 and WFG 5 and the very complex disconnected benchmark functions such as ZDT3.

CHAPTER 6: MOSFP METHOD FOR SOLVING WIRELESS SENSOR NETWORKS OPTIMIZATION PROBLEMS IN SMART GRID APPLICATIONS

6.1 Introduction

Wireless Sensor Network (WSN) consist of a large number of sensors embedded with various kinds of devices to detect and monitor the physical phenomena such as heat, pressure, light, etc. These sensors were first employed in military applications. For example, video surveillance in tricky areas such as forests (Akyildiz, 2007). Nowadays, a wide variety of short-range communication technologies such as Wi-Fi, ZigBee, etc. are developed to support sensor-based devices. These technologies can operate on Industrial, Scientific and Medical (ISM) band ((Abu-Sharkh, 2015); (Aguirre, 2016)) with various communication ranges. WSN is rapidly gaining popularity for sensing, detecting and monitoring in many applications such as industrial infrastructure, automation, traffic, health, and various consumer areas (Doudou, 2016).

However, sensor nodes in WSN are susceptible to various challenges due to their limited communication ranges, limited memory size, and limited power in the battery (Singh, 2016). The misuse and mismanagement of these devices will reduce the Quality of Service (QoS), and the network lifetime especially in dense networks. As an instance, if the packet payload size is increased, the probability of dropping data packet will be increased. The retranslation of these packets will require reallocation of the dropped packets in the memory and consumes more power of the battery. In addition, this procedure will consume more time and increase the network delay.

To mitigate the effect of these problems and challenges, different Multi-Objective Optimization Algorithms (MOOAs) have been used such as algorithms in Section 2.3 of Chapter 2. As mentioned in Section 2.5 of Chapter 2, many researchers used various

MOOAs to optimize a set of mathematical network models (objective functions) of WSN such as network coverage, throughput, network energy consumption. ((Jia, 2009) – (Hamdan, 2017)). These network models consist of maximization and minimizations. An example of minimization network model is energy consumption while an example of the maximization network model is network coverage. These models also include the effect of different network parameters such as frequency range, packet payload size, and the distance between transmitter and receiver. Furthermore, these models involve the parameters of interference such as Packet Error Rate (PER) and depend on the interference from different devices that work on the same frequency band. The study of these objective functions (mathematical network models) is very important to understand the challenges in the network during the implementation phase.

In prior work, limited attention is given to find the optimum value for some parameters in network physical layer such as packet payload size (see Section 2.5 of Chapter 2, Table 2.6). The size of packet payload can affect the network objective functions such as end-to-end latency, end-to-end delay, network throughput, and energy efficiency. These models are very important especially for critical real time WSN applications that can be affected by end-to-end delay such as smart grid network, health monitoring network and disaster monitoring network, which the data of these networks should be received without delay. Therefore, optimization algorithms are very important to determine the optimal value of different parameters that affecting network QoS.

Sensors in smart grid send various information about voltage stability, power quality, and power consumption to control centre (power generator). This helps to monitor and generate the power in a real-time, control a power outage, monitor the power quality, and control power load. The power companies can also use this information to develop a real-time pricing. The information of real-time power pricing can guide the consumers in their power consumption by reducing it during peak time (power selling price at the highest) ((Fadel, 2015); (Xiong, 2011)). Therefore, this critical information should be received with minimum delay. To minimize the latency and delay, the amount of data packet that generated by these devices should be optimized to minimize the process of packetizing and transmitting the data to the control center. Consequently, the data can be received in a real-time manner and reduce the energy consumption in the network.

In this chapter, four MOOAs namely, Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Deb, 2002), Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Eckart, 2001), Optimized Multi-Objective Particle Swarm Optimization (OMOPSO) (Sierra, 2005), and our proposed Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP) are applied. The chapter aims to search optimal value of packet payload size that manages and balances the trade-offs between a set of objective functions. The optimal value considers four important objective functions, including packet throughput, energy efficiency, end-to-end latency and end-to-end delay. The inclusion of these objective functions is believed to enhance QoS of a communication link in the smart grid network ((Al-Anbagi, 2015); (Brown, 2012)) as previous studies only concentrate on minimizing energy consumption and maximizing network coverage.

This chapter is organized as follows: Section 6.2 presents the quality of services models in WSN. A case study is demonstrated in Section 6.3. Section 6.4 describes the methodology and experimental setup. The experimental results are presented in Section 6.5 and discussed in Section 6.6. Section 6.7 summarizes the work in this chapter.

6.2 The Quality of Services Models of WSN

Various challenges in WSN have been studied in prior work and lead to the development of new enhanced versions of MOOAs (see Section 2.5 of Chapter 2).

Some of these works are evaluated in optimizing challenges and problems in WSN, while others used the exact MOOAs to optimize a set of objective functions that affect network QoS. However, important network metrics such as end-to-end delay, end-to-end latency, and network throughput have not yet been studied in these works as shown in Table 2.3. These metrics are very important to determine the whole QoS of any WSN. For instance, if the network end-to-end delay is increased, the probability of dropping packets will be increased and tasks of retransmission of these packets will consume more energy and time. Therefore, we are going to fill this gap by optimizing the network end-to-end delay and other network models using our propose algorithm (MOSFP) and other well–known MOOAs such as OMOPSO, SPEA2, and NSGA-II.

The following subsections describe important objective functions that are used to estimate the quality of communication link for WSN. These objective functions are end-to-end latency, end-to-end delay, energy efficiency, and packet throughput.

6.2.1 End-to-End Delay Model

This model estimates the time needed to successfully transfer the data packet from source to destination, including, the transition time of packet (T_{packet}), backoff time (T_{bo}), inter-frame space-time (T_{IFS}), turnaround time of transceiver's (T_{TA}), and acknowledgment of packet receipt time (T_{ACK}). The end-to-end delay can be given by (T_l) (Liang, 2007):

$$T_l = T_{packet} + T_{IFS} + T_{bo} + T_{TA} + T_{ACK}, \tag{6.1}$$

T_{packet} is the transmission time for any data packet to reach the sink node. It can be defined as follows:

$$T_{packet} = \frac{L_{PHY} + L_{MHR} + payload + L_{MFR}}{R_{data}}, \tag{6.2}$$

Where:

- R_{data} is the data transmission rate;
- L_{PHY} is the size of the physical header in byte;
- L_{MHR} is the size of MAC header in byte;
- $payload$ is the size of data in the packet in byte;
- L_{MFR} is the size of MAC footer in byte.

Backoff periods should be defined for the sensor that wants to transmit the data packet through the medium. This formula can be calculated by determining the probability of any node (p_S) for accessing the network medium in a successful way. p_S can be defined as follows:

$$P_S = \sum_{a=1}^{a=b} P_C(1-P_C)^{(a-1)}, \tag{6.3}$$

Where p_C is the estimation probability of the ideal channel that calculated by any sensor at the end of any backoff period while b is the maximum number of backoff periods. The probability of the ideal channel (p_C) can be given by:

$$P_C = (1-q)^{n-1}, \tag{6.4}$$

Where q is the probability of node to transmit a data packet at any time while n is the number of nodes that operate on the network. The average of backoff periods (R) can be given as:

$$R = (1-P_S)b + \sum_{a=1}^{a=b} aP_C(1-P_C)^{(a-1)}, \tag{6.5}$$

Hence, (T_{bo}) is the total of backoff time, which can be expressed as:

$$T_{bo} = FractionalPart[R]T_{bop}(IntegerPart[R]+1) + \sum_{a=1}^{a=IntegerPart[R]} T_{bop}(a). \tag{6.6}$$

Where T_{bop} is the average backoff period, which can be given as:

$$T_{bop(a)} = \frac{2^{macMinBe+a-1}-1}{R_{data}} T_{boslot}, \tag{6.7}$$

Where:

- macMinBE is the initial value of backoff;
- T_{boslot} is the backoff time at one slot duration. For Zigbee/ IEEE 802.15.4 one slot duration is equal to the duration of twenty symbols.

6.2.2 End-to-End Latency Model

Mainly, the output of any node in WSN is an analog signal. The node digitizes the data and stores it in the buffer (main memory). These data will be packetized as a form of data packets and transmitted periodically. Figure (6.1) summarizes the sampling cycle and transmitting cycle of any node in WSN (Liang, 2007).

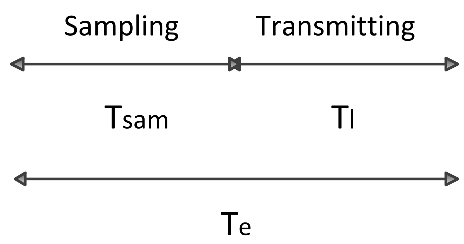


Figure 6.1: Sampling and transmitting cycle of a sensor node (Liang, 2007)

End-to-end latency (*Te)* is the amount of time between the data packet is produced at the node and received by the destination node defined as follows (Liang, 2007):

$$Te = T_{sam} + T_l , \tag{6.8}$$

Where T_{sam} is a sampling time to measure the amount of time needed by any sensor for sampling a signal until the number of samples reaches a certain size. T_l is the end-to-end delay. T_e is a dependent model that depends on the parameters of end-to-end delay model. The end-to-end delay will play a significant role in determining T_e model. The sampling time of Zigbee/IEEE 802.15.4 standard can be calculated as follows:

$$T_{sam} = \frac{payload}{Sampling\ rate}, \tag{6.9}$$

6.2.3 Energy Efficiency Model

Energy efficiency (η) model an important metric in estimating various types of networks especially for the network powered by battery such as WSN. This metric is affected by two parameters namely, PER and packet payload length. This metric should be maximized to increase the network QoS and can be calculated as in Equation (6.10) (Hamdan, 2015):

$$\eta = \frac{E_C \cdot payload}{E_C \cdot (payload + h_{(L_{MHR} + L_{MAC})}) + E_S} \cdot (1 - PER), \tag{6.10}$$

Where

- E_C is the energy consumption during the process of communication;
- E_S is the energy consumption in start-up mode;
- $payload$ is the size of data in the packet in byte;
- $h_{(L_{MHR}+L_{MAC})}$ is the packet header length contributed by the summation of L_{PHY} and L_{MHR}. L_{PHY} is the size of physical header in byte while L_{MHR} is the size of MAC header in byte;
- PER is the Packet Error Rate.

6.2.4 Network Throughput Model

Network throughput (u_{tput}) is an important network model to measure the rate of successful data packets that are transferred over the communication medium. The QoS of the network can be determined by measuring this model, which if the u_{tput} increases, the whole QoS of the network will be increased. u_{tput} is affected by two parameters such as packet payload length and PER. The u_{tput} can be expressed as follows (Hamdan, 2015):

$$u_{tput} = \frac{payload \cdot (1 - PER)}{T_{flow}}, \tag{6.11}$$

Where

- T_{flow} is the transmission latency;
- $payload$ is the packet payload size;
- PER is the packet error rate and can be calculated as in equation (6.12) (Ganhão, 2013):

$$PER = 1 - (1 - BER)^{(Length-of-packet-in-bits)}, \tag{6.12}$$

Where BER is the bit error rate.

6.3 Case Study of Smart Grid Network

In this chapter, smart grid (Kabalci, 2016) is considered as a case study. The smart grid is a modern generation electricity grid (Kabalci, 2016) and compatible with different systems. This allows information to spread in the areas of adaptive control, wireless communication, embedded sensing, and pervasive computing, to significantly improve the sustainability, security, stability, and performance of the electrical grid. The smart grid has a hierarchical communication infrastructure, which provides three fundamental functionalities, including, monitoring for control, sensing, and transmitting. The first two functionalities are carried out by various types of embedded sensor nodes and smart meters to detect and monitor the status of the various parts of the grid in a real-time manner. The smart grid is developed to support two-way links of packet transmission between the embedded sensors and the control centers (Joshi, 2016). The control instructions can be transmitted from/to the embedded sensors, gateways, or smart meters fixed in various areas to support stable and reliable access to grid components. Also, guarantees the efficient and high-performance operations of the smart grid. The infrastructure of the smart grid consists of three parts. These parts are different in their size and location in which, form a hierarchical communication infrastructure (Keyhani, 2016). These parts are described as follows:

- **Home Area Network (HAN):** This part of smart grid operates based on a local area wireless network or short-range communication protocols (e.g., ZigBee). HANs supports real-time data transmission of a power load control, dynamic pricing, and smart meter by connecting various types of devices with actuators, in-home display, sensors, and smart meter. Short-range technologies are suitable for HANs because of low power consumption, low installation cost, high performance of control, and flexibility. An example of this technology is ZigBee and suitable for HANs due to its high interoperability ((Niyato, 2012); (Sethi, 2017)). The important part of HAN is the HAN gateway to send data to an external entity such as Data Aggregator Unit (DAU). The main aim of DAU is to aggregate the data of smart meters. Moreover, it is used to retransmit these data to control center. The HAN gateway can be standalone within various devices of home (e.g. in-home display, programmable thermostat, etc.) or alternatively integrated with a HAN smart meter.
- **Neighbourhood Area Network (NAN):** The NAN plays a significant role in connecting a set of Home Area Networks (HANs) together and connecting HANs with the control center. The aim of HAN gateway is to send the data of smart meters from HANs to Data Aggregation Unit (DAU) through NAN. The HAN gateway and DAU communicate with each other through wireless technologies such as 801.11s, WiMAX, RF mesh, 4G, 3G, and LTE. DAU can take a role of NAN gateway in transferring the collected data to Meter Data Management System (MDMS). MDMS represents a control center that gathers meter data, process the collected data of power consumption, store a copy of these data, and generate a report about the

power generation, and manage the place of power distribution and transmission ((Niyato, 2012); (Cacciapuoti, 2016)). NAN is depicted in Figure 6.2 (b).

- **Wide Area Network (WAN):** The WAN connects the remote systems together in the grid. Examples of these systems are MDMS, Synchronous Optical Network (SONET), and Advanced Metering Infrastructure (AMI). These systems collect the data from smart meters. The Wide-Area Measurement System (WAMS) in WAN manages the process of both aggregating data and transmitting data for power load measurement and for control purposes. The WAN supports a backhaul connection among customer premises, distributed subsystems, public utility, and the power generators. The backhaul can support various technologies (e.g., broadband wireless network or cellular access network) to send the smart meters' data from a NAN to the DAU and then from DAU to MDMS at control center. The WAN has a powerful gateway, which supports WiMAX, 4G, 3G, and satellite to collect the required data ((Niyato, 2012); (Jiang, 2016)).

Table 6.1: Summary of the smart grid characteristics based on hierarchical communications infrastructure ((Shandilya, 2016); (Yu, 2011))

Cognitive area networks	Wide Area Network (WAN)	Neighborhood Area Network (NAN)	Home Area Network (HAN)
Network topology	Centralized	Centralized	Centralized/decentralized
Spectrum band	Licensed band	Licensed band	Unlicensed band
Favorable network protocol	WiMax, 3GPP, RF Mesh, and satellite	801.11s, RF mesh WiMax, 3G, 4G and LTE	IEEE 802.15.4
Network users	Spectrum broker, NGWs	HGWs, NGW	Smart sensors/meters/actuators, HGW
Featured strategy	Optimal spectrum leasing	Hybrid dynamic spectrum access	Cross-layer spectrum sharing
Application	Demand Resource and load management	Advanced metering infrastructure, demand resource, and load management	Advanced metering infrastructure, demand resource, etc
Key techniques	Join spectrum management	Spectrum handoff, guard	Power coordination, access control

channel

The described hierarchical communications infrastructures (HAN, NAN, and WAN) are used with smart grid to increase the network stability and performance. However, smart grid faces a challenge of increasing the number of smart appliances and smart meters (Shandilya, 2016). This leads to the increase of network end-to-end delay especially in the crowded cities. For this reason, in this chapter, we are going to optimize the network delay of WSN by minimizing the delay to increase the network QoS. This infrastructure of smart grid can be summarized in Figure 6.2 (a) (Yu, 2011). The characteristics of the smart grid are outlined in Table (6.1) ((Yu, 2011); (Ahmad, 2016)). Based on Table 6.1, HAN is the only part of the smart grid that uses short range communication protocols such as ZigBee/ IEEE 802.15.4. The characteristics of IEEE 802.15.4 can be summarized in the following subsections.

6.3.1 IEEE 802.15.4 Protocol

IEEE 802.15.4/ZigBee standard is a modern protocol for wireless communication and its characteristics make it suitable for smart grid network. Examples of these characteristics are low power consumption, cheap price, good data rate and low complexity. Furthermore, this protocol can support Carrier Sense Multiple Access (CSMA) to manage the communication between nodes without any collision. Various frequency bands are supported by this protocol (Ahmad, 2016). These bands have different data rate, frequency ranges, and number of channels ((Nobre, 2015); (Dinh, 2016)) as summarized in Table 6.2 (Yassein, 2016). In this work, the frequency band of 2.4 GHz is chosen because it can operate on a higher data rate up to 250 kbps, supports 16 channels for transition and permitted in Asia ((Dinh, 2016); (Yassein, 2016)).

IEEE 802.15.4/ZigBee supports various network topologies such as cluster tree, mesh, and star topologies (Wang, 2016). Table 6.3 summarizes the data frame structure of IEEE 802.15.4/ZigBee. This data frame consists of four parts, including, acknowledgment frame, MAC command frame, beacon frame and data frame. From Table 6.3, IEEE 802.15.4/ZigBee can support a MAC packet size up to 127 bytes. Based on that, the packet payload size that is supported by IEEE ZigBee/802.15.4 can reach up to 114 bytes (Hamdan. 2015).

Table 6.2: A set of radio frequency bands along with their characteristics that supported by IEEE 802.15.4/ZigBee standard (Yassein, 2016)

Frequency bands	**Area**	**Data rate (kbps)**	**Frequency range (MHz)**	**Number(s) of channel**
915 MHz	Australia, America	40	902-928	10 channels
2.4 GHz	Asia, Worldwide	250	2405-2480	16 channels
868 MHz	Europe	20	868.3	1 channel

Table 6.3: Data frame structure of The IEEE 802.15.4 (Hamdan, 2015)

<table>
<tr><td colspan="3"></td><td>2 bytes</td><td>1 byte</td><td>0-20 bytes</td><td>Variable</td><td>2 bytes</td></tr>
<tr><td colspan="3" rowspan="2">MAC sublayer</td><td>Frame control</td><td>Sequence number</td><td>Address fields</td><td>Data payload</td><td>Frame check sequence</td></tr>
<tr><td colspan="3">MAC Header</td><td>MAC Service Data Unit</td><td>MAC Footer</td></tr>
<tr><td>PHY layer</td><td>Sync Header</td><td>PHY Header</td><td colspan="5">PHY Service Data Unit (PSDU)</td></tr>
<tr><td></td><td>5 bytes</td><td>1 byte</td><td colspan="5">≤127 bytes</td></tr>
</table>

6.4 Methodology and Experimental Setup

This section focuses on the HAN part of the smart grid. HAN consists of various sensor nodes and operates using IEEE 802.15.4/ZigBee protocol. These sensors are embedded in various home appliances operated on MicaZ platform (Martinez-Sandoval, 2014). The characteristics of MicaZ platform are outlined in Table 6.4 (Datasheet. 2006).

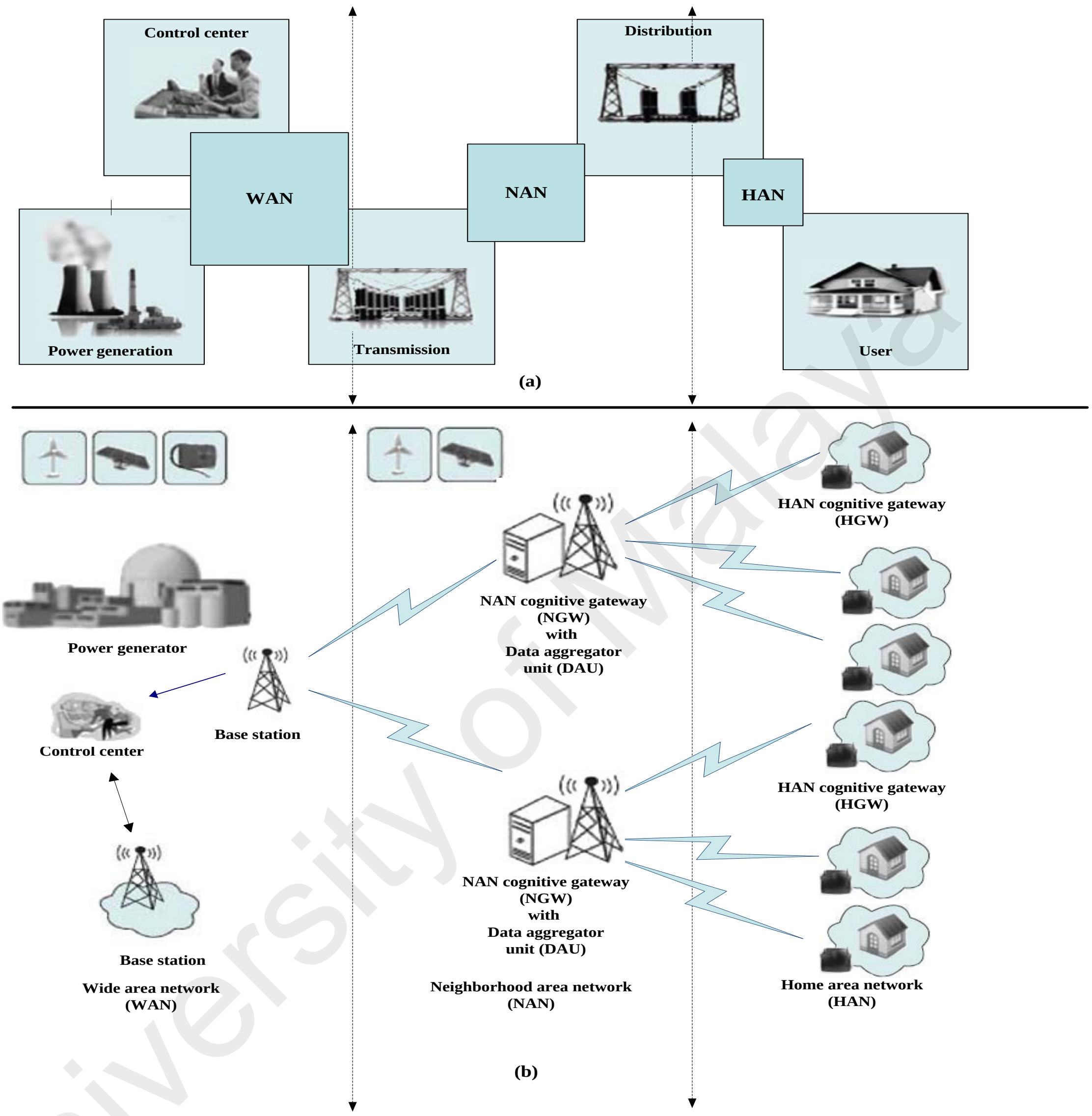


Figure 6.2: Hierarchical communications infrastructure of smart grid. The figure is obtained from (Yu, 2011)

MicaZ platform is suitable for smart grid because it operates on ISM frequency band (license-free band), consumes low power through its operation, and can cover up to 30 meters of buildings or homes areas. However, MicaZ is operated on batteries, thus, has a limited energy resource. Therefore, usage of these devices should be managed properly to extend the battery power and the sensor lifetime. In addition, the number of smart meters affects the network, which if the number of smart meters in HANs increases, the network delay will be increased especially in the crowded cities. This leads to the increase of dropped packets. Therefore, retransmitting the dropped packets will consume more time and power. Based on these limitations, in this chapter, four algorithms such as MOSFP, NSGA-II, OMOPSO, and SPEA2 are used to minimize end-to-end latency and end-to-end delay. The same algorithms are also used to maximize network throughput and network energy efficiency.

Table 6.4: Characteristics of MicaZ platform (Datasheet. 2006)

Features	**Value**	**Remarks**
Frequency band	2.4 GHz band	License free band (ISM band)
Data rate	250 kbps	-
EEPROM	4K bytes	-
Operating system	TinyOS	Open-source
Battery	2X AA batteries	Attached pack
Energy consumption in startup mode	8 mA	-
Energy consumption in communication mode	19.7 mA	-
User Interface	3 LEDs	Red, green and yellow
Range	75 m to 100 m	1/2 wave dipole antenna

In this chapter, the smart home in HAN consists of four sensors are proposed. All of them are embedded in four appliances such as smart controller of air conditioner and light, smart refrigerator, smart TV, and smart washing machine. These sensors communicate with a smart home gateway using the star topology. This gateway is embedded and integrated with the smart meter in one device. This network can be summarized in Figure 6.3. The values of energy consumption for these sensors in start-up mode and in communication mode are equal to 8 mA and 19.7 mA respectively (Datasheet, 2006). The sampling rate of 802.15.4/ZigBee can be varied from 0 to 250 Hz (Bhuiyan, 2017). This can satisfy the requirements of new electricity grid (the smart grid). Normally, these sensors have a BER value, which equal to 0.0004 (Saadon, 2013). By knowing these values and the other values such as the IEEE 802.15.4 physical and MAC headers, the objective functions (equations from 6.1 to 6.12) can be measured. This research is focused on maximizing both packet throughput and network energy efficiency and also minimizing both the end-to-end latency and network end-to-end delay by changing the parameter of packet payload size. The parameter of packet payload size is very important to measure as it plays a significant role in determining the optimal value of the previous mentioned objective functions. As an instance, if the parameter of packet payload size increases, the energy efficiency will be decreased and also the network end-to-end delay will be increased.

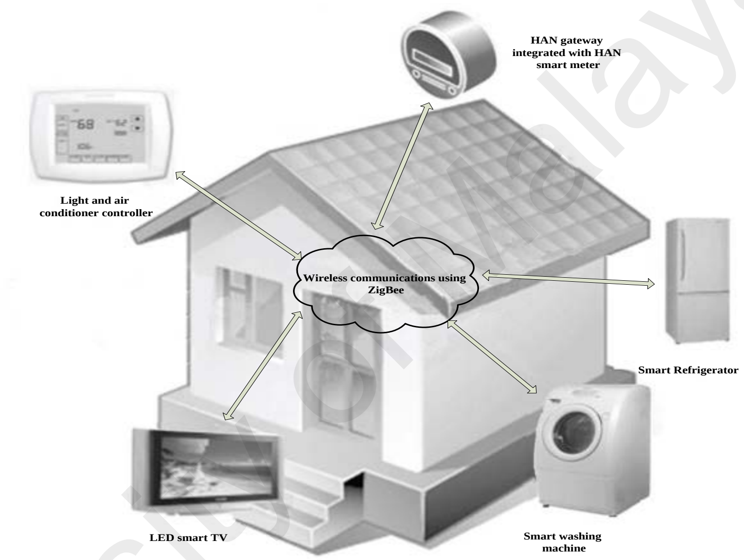


Figure 6.3: The proposed network of Home Area Network (HAN)

For the experiment, Java version of JMetal 4.5 tool is compiled in NetBeans IDE 8.0.2. The experiment is conducted on Intel dual-core CPU-T3200, 3 GB RAM and Windows 7 operation system. The parameters for the optimization algorithms used in this chapter are summarized in Table 6.4. These settings and parameters are assigned as in ((Hamdan, 2015); (Hamdan, 2017)). For modeling part, the settings and parameters are summarized in Table 6.5. The procedure of minimizing the network end-to-end latency and network end-to-end delay as well as maximizing the network energy efficiency and the packet throughput are summarized for each algorithm in Figures 6.4 to 6.5.

The procedure of OMOPSO begins by initializing the packet payload size that changes in the range of 0 to 114 bytes depends on the IEEE 802.15.4/ZigBee data frame as summarized in Figure 6.4. The OMOPSO archives the leaders and perform crowding operation on the elected leaders. If the number of leaders is greater than the determined size, the algorithm keeps the best leaders and eliminates the others. This process involves the execution of the velocity update rule on each particle in the population. Hence, the algorithm performs the mutation operation. Furthermore, the OMOPSO evaluates the objective functions (equations from 6.1 to 6.12) by using each member in the population to maximize both network throughput and energy efficiency and to minimize both end-to-end latency and network end-to-end delay. Now, the algorithm compares the new fitness value of each individual with its old fitness value. The new fitness of the individual will be kept if it is better than old one. Then, OMOPSO updates the leaders of the new population followed by archiving and crowding operators on the leaders. Finally, the number of iterations will be checked. If the maximum number generations (the value of 250 generations as in Table 6.5) is reached, the procedure will terminate, else, the procedure will repeat the past steps.

Figure 6.5 illustrates the procedure of NSGA-II algorithm. NSGA-II begins by initializing the packet payload size. The upper limit and lower limit of this parameter are 114 bytes and 0 byte respectively based on the IEEE 802.15.4/ZigBee data frame. Depends on the first generation of this parameter, the procedure evaluates the objective functions (equations from 6.1 to 6.12), which maximizes both energy efficiency and network throughput, and also minimizes both the end-to-end latency and network end-to-end delay. Moreover, the procedure ranks the population based on values of non-dominated solutions. Then, it performs a set of operations namely, selection, crossover, and mutation operations to generate child population (new population). Based on the results of the prior steps, the procedure uses the child population to evaluate the same equations (objective functions). Therefore, the procedure combines the old population with the child population. Later, the NSGA-II ranks the combined population from the best to the worst results. At the end of the procedure, the number of iterations will be checked, which if the number of iterations is reached the maximum generations (the value of 250 generations as in Table 6.5), the procedure will terminate, else, the procedure will repeat the past steps.

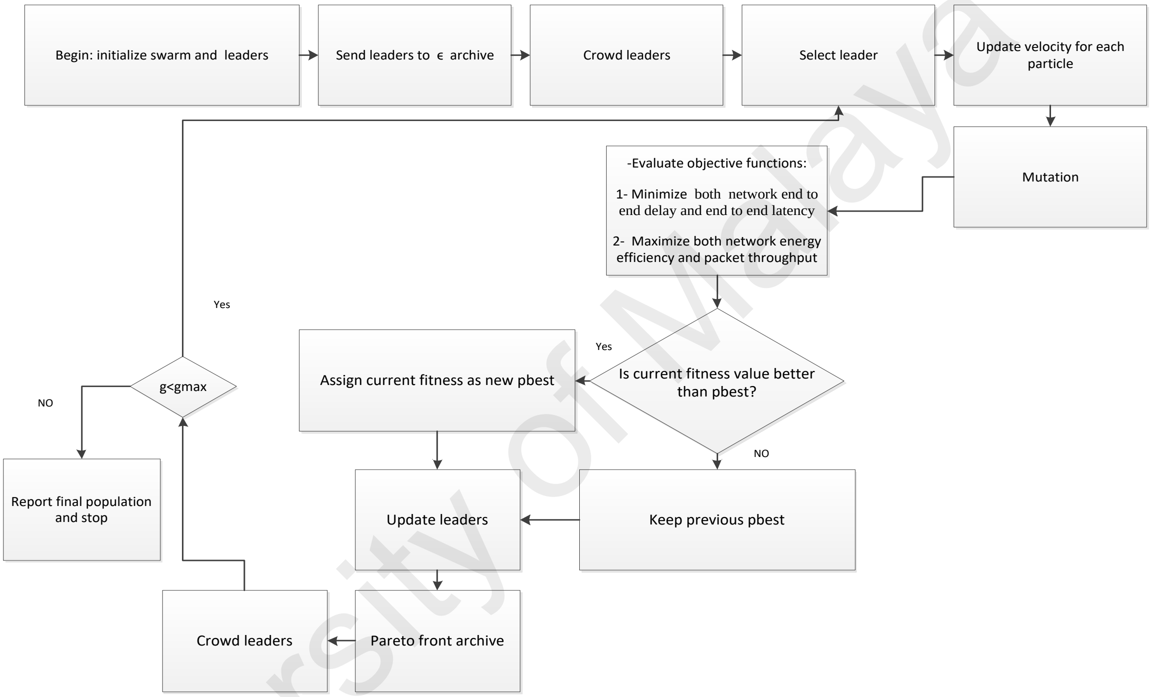


Figure 6.4: Procedure of OMOPSO algorithm in maximizing both energy efficiency and network throughput, and also in minimizing both the end-to-end latency and network end-to-end delay of HAN

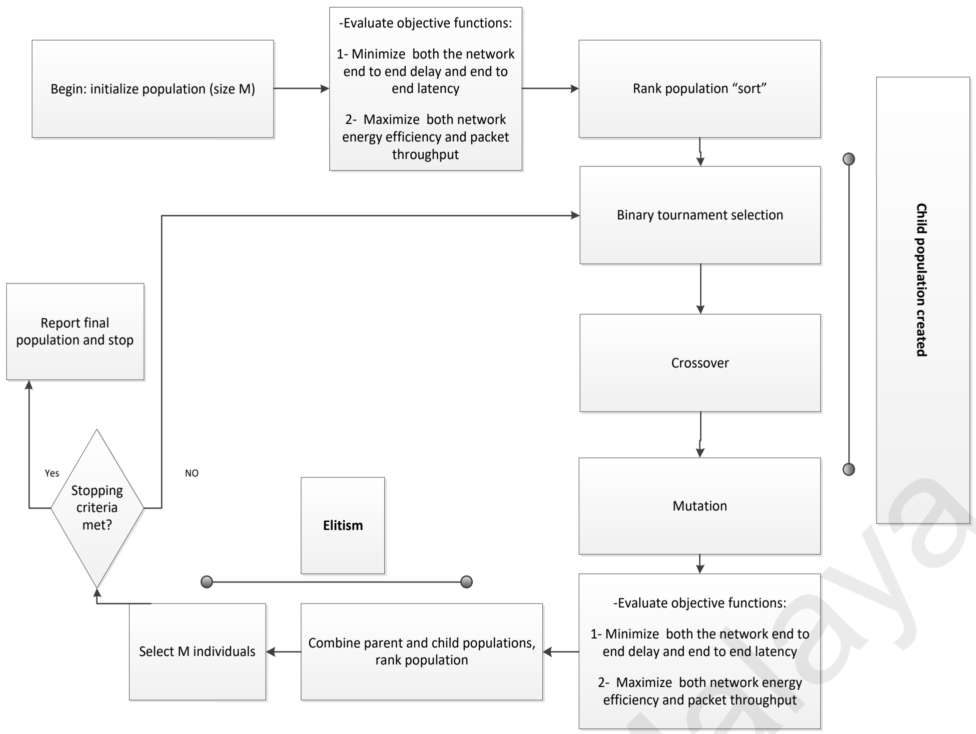


Figure 6.5: Procedure of NSGA-II algorithm in maximizing both energy efficiency and network throughput, and also in minimizing both the end-to-end latency and network end-to-end delay of HAN

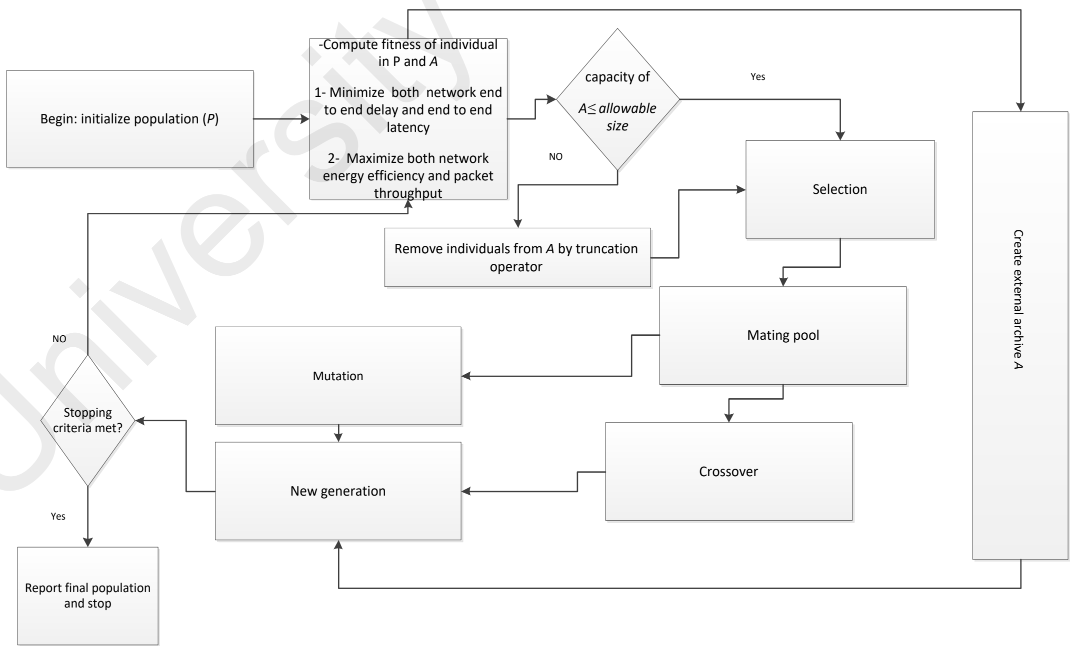


Figure 6.6: Procedure of SPEA2 algorithm in maximizing both energy efficiency and network throughput, and also in minimizing both the end-to-end latency and network end-to-end delay of HAN

Procedure of SPEA2 as in Figure 6.6 can be summarized as follows. SPEA2 begins by initializing the packet payload size within 0 to 114 bytes depending on the

IEEE 802.15.4 data frame. The first population is used to maximize both the energy efficiency and network throughput and also minimize both the end-to-end latency and network end-to-end delay. Then, SPEA2 performs the selection operation on the generated fitness values. After the selection iteration, the SPEA2 generates the mating pool. This pool represents the population that both the crossover and mutation operations are applied to them in order to produce a new population. At the end of the procedure, the number of iterations will be checked, which if the number of iterations is reached the maximum generations (the value of 250 generations as in Table 6.5), the procedure will terminate, else, the procedure will repeat the past steps.

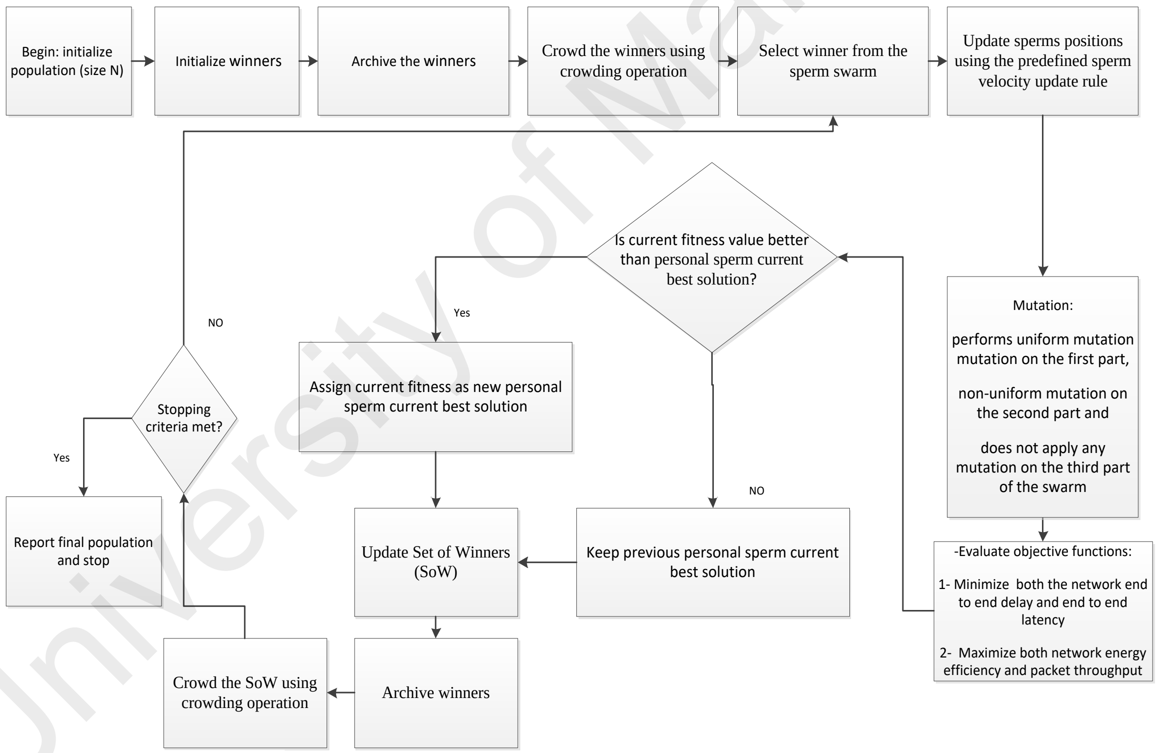


Figure 6.7: Procedure of MOSFP algorithm in maximizing both energy efficiency and network throughput, and also in minimizing both the end-to-end latency and network end-to-end delay of HAN

Similar to SPEA2, NSGA-II, and OMOPSO, MOSFP algorithm begins by initializing the parameter of packet payload size (see Figure (6.7)). The upper limit and lower limit of this parameter are 114 bytes and 0 byte respectively based on the IEEE 802.15.4/ZigBee data frame (see Table 6.3). Then, MOSFP archives the required number of winners and crowds those winners based on crowding operation. At this stage, the algorithm checks the size of the winners. If the size of the winners is greater than the defined maximum size of winners, the algorithm eliminates the worst winners and keeps the best winners. This process involves applying the velocity update rule on each sperm in the population. Furthermore, MOSFP performs different types of mutation operations (e.g. unifrom and non-uniform mutations) on the population to prepare it for evaluation stage. In the evaluation stage, the procedure uses the population to maximize both the packet throughput and energy efficiency and to minimize both the end-to-end latency and the end-to-end delay of the network. MOSFP changes the fitness value of each sperm in the population just if the new fitness value of the sperm is better than the old one. Hence, the procedure updates the set of winners. Then, MOSFP performs both of archiving and crowding operators on the winners. Finally, the number of iterations will be checked. If the number of iterations is reached the maximum generations (the value of 250 generations as in Table 6.5), the procedure will terminate, else, the procedure will repeat the past steps.

Table 6.5: Parameters of the algorithms

Parameters	MOSFP	OMOPSO	NSGA-II	SPEA2
Population size	20	20	20	20
Archive size	(winner) 20	20	(Elite) 20	20
Mating pool size	-	-	-	20
Maximum generation	250	250	250	250
Crossover probability	-	-	0.9	0.9
Mutation probability	1/d where d is the variable code size			

Table 6.6: Simulation parameters

No.	Parameter	Values
0	Time of interframe space (T_{ifs})	192 μ*s*
1	Transceiver's transmitting to receiving turnaround time (T_{TA})	192 μ*s*
2	The duration of one backoff slot (*Tboslot*)	320 μ*s*
3	Use of ACKs	N0
4	PHY header (*LPHY*)	6 bytes

5	MAC header (*LMHR*)	11 bytes
6	MAC footer (*LMFR*)	2 bytes
7	The default minimum value of backoff exponent (*macMinBE*)	3
8	The default maximum value of backoff exponent (*aMaxBE*)	5
9	Number of sensors (n)	5
10	Transceiver's raw data rate (R_{data})	250 kbps
11	The energy consumption in startup mode (E_s)	8 mA
12	Energy consumption through the communication (E_c)	19.7 mA
13	Sampling rate	250 Hz
14	Bit Error Rate (BER)	0.0004

6.5 Experimentation and Results

The experimental results are analyzed in two ways. First, the outcomes from each method for the four objective functions based on ten-time runs are analyzed using Tukey's test (one-way ANOVA). In this test, the mean difference between the methods is significant if the (p-value) is smaller than 0.05.

Second, Pareto front sets from the four algorithms are analyzed for the four objective functions. The Pareto front is used to illustrate the trade-offs between a set of objective functions (optimization functions), which helps to know the optimal value of packet payload size that manages trade-offs between the proposed objective functions.

6.5.1 Comparisons Between the Four Algorithms Using Statistical Analysis

Table (6.7) summarized the objective functions namely, end-to-end latency, end-to-end delay, packet throughput, and energy efficiency for ten time runs for each method. The statistical analysis using Tukey's test (one–way ANOVA) outlined in Table (6.8) shows that our method (MOSFP) significantly outperforms SPEA2 in which, MOSFP substantially decreases the end-to-end latency (-2.718, p=0.001) and end-to-end delay (-0.265, p=<0.001), and increases the packet throughput (0.394, p=<0.001) and energy efficiency (0.116, p=0.001) compared to SPEA2. These mean differences also show that MOSFP outperforms SPEA2 by 41%, 24%, 99%, 41% in term of end-to-end latency, end-to-end delay, packet throughput, and energy efficiency respectively. However, no significant mean variance is observed between MOSFP and the other algorithms i.e. NSGA-II and OMOPSO for all the objective functions. This indicates that MOSFP outperforms NSGA-II and OMOPSO with a small mean variance between them in the range of 3% to 9%.

This subsection highlights another important aspect of the analysis i.e. the consistency of the method to perform between runs. An algorithm with a small standard deviation of the objective function will be considered as a more stable algorithm. From the experiments, SPEA2 has resulted in a more consistent performance for three objective functions between ten time runs among the algorithms. The standard deviations of SPEA2 are approximately 7%, 23%, and 3% much smaller compared to others for end-to-end delay, packet throughput, energy efficiency. In end-to-end latency, MOSFP has shown a more consistent efficiency and performance when its standard deviations are 3%, 4%, and 8% much smaller than NSGA-II, OMOPSO, and SPEA2.

Table 6.7: Comparison between SPEA2, MOSFP, OMOPSO and NSGA-II for four objective functions. The highlighted background with bold font represents the best average for the respective objective function

Objective Functions	Algorithms	Mean	Std. Dev.	Std. Error	95% Confidence Interval for Mean		Min	Max
					Lower Bound	Upper Bound		
Energy Efficiency	SPEA2	0.285	0.181	0.013	0.260	0.310	0.096	0.602
	MOSFP	**0.401**	0.185	0.013	0.375	0.427	0.096	0.602
	OMOPSO	0.392	0.188	0.013	0.366	0.418	0.096	0.602
	NSGA-II	0.387	0.186	0.013	0.361	0.413	0.096	0.602
Packet Throughput	SPEA2	0.398	0.828	0.059	0.282	0.513	0.010	3.161
	MOSFP	**0.791**	1.074	0.076	0.642	0.941	0.010	3.154
	OMOPSO	0.776	1.079	0.076	0.626	0.926	0.010	3.154
	NSGA-II	0.725	1.054	0.075	0.578	0.872	0.010	3.158
End-to-End Delay	SPEA2	1.105	0.422	0.030	1.047	1.164	0.240	1.493
	MOSFP	**0.840**	0.454	0.032	0.777	0.903	0.240	1.492
	OMOPSO	0.858	0.459	0.032	0.794	0.922	0.240	1.492
	NSGA-II	0.874	0.453	0.032	0.811	0.937	0.240	1.493
End-to-End Latency	SPEA2	6.597	4.927	0.348	5.910	7.284	0.279	15.579
	MOSFP	**3.880**	4.539	0.321	3.247	4.513	0.279	15.533
	OMOPSO	4.137	4.741	0.335	3.476	4.798	0.279	15.535
	NSGA-II	4.185	4.673	0.330	3.534	4.837	0.279	15.601

Table 6.8: Analysis of one-way ANOVA (Tukey's test) between SPEA2, MOSFP, OMOPSO and NSGA-II for four objective functions

Objective Functions	Algorithm (I)	Algorithm (J)	Mean Difference (I-J)	Std. Error	*p*-value	95% Confidence Interval	
						Lower Bound	Upper Bound
Energy Efficiency	MOSFP	SPEA2	0.116	0.018	<0.001*	0.068	0.163
		OMOPSO	0.009	0.018	0.965	-0.039	0.056
		NSGA-II	0.013	0.018	0.888	-0.034	0.061
Packet Throughput	MOSFP	SPEA2	0.394	0.101	<0.001*	0.133	0.655
		OMOPSO	0.015	0.101	0.999	-0.246	0.276
		NSGA-II	0.066	0.101	0.916	-0.195	0.327
End-to-End Delay	MOSFP	SPEA2	-0.265	0.045	<0.001*	-0.380	-0.150
		OMOPSO	-0.018	0.045	0.977	-0.133	0.097
		NSGA-II	-0.034	0.045	0.872	-0.149	0.081
End-to-End Latency	MOSFP	SPEA2	-2.718	0.472	<0.001*	-3.933	-1.502
		OMOPSO	-0.257	0.472	0.948	-1.473	0.958
		NSGA-II	-0.306	0.472	0.917	-1.521	0.910

*The mean difference is significant at the 0.05 level.

Overall, our algorithm (MOSFP) obtained the best average of all the objective functions while OMOPSO in the second followed by NSGA-II and SPEA2 respectively. In term of performance consistency, results of SPEA2 have shown a more consistent efficiency and performance in end-to-end latency.

6.5.2 Analysis of Pareto-Optimal Set of the Four Algorithms

As introduced in Chapter 1, the MOOPs are a set of conflict objective functions that consist of maximization and minimization objective functions (Dreżewski, 2017). The concept of Pareto optimality is emerged in 1906 by Vilfredo Pareto as an idea to manage the trade-offs between these objective functions (Engelbrecht, 2006). This concept mainly based on the Pareto front set that is used to balance the objective functions. Two concepts are defined based on Pareto front:

a) The Marginal concept of optimality: this concept aims to define the optimal value of a set of conflicting objective functions based on the intersection point. The intersection point between a set of maximization and minimization objective functions is considered as the optimal value (Bortolotti, 1999). An example of the intersection between two objective functions is illustrated in Figure (6.8) (Massiani, 2013).

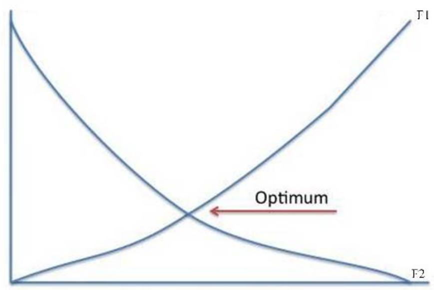


Figure 6.8: The optimum value based on the intersection between two objective functions (Massiani, 2013)

b) The knee point: this point is on the curve of Pareto front and the most preferred solution. This point can be defined by determining the greatest reflex angle that bends of the front from its lift side to its right side or vice-versa. Figure (6.9) illustrates the knee point concept (Deb, 2011). Point B is the knee point, which makes the greatest reflex angle between the A point on the left side of the Pareto front curve and the C point on the right side of the Pareto front curve.

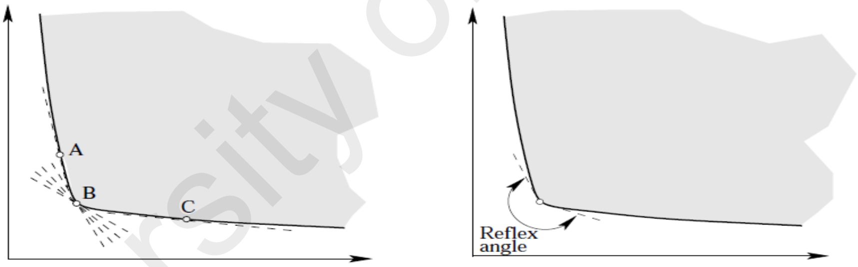


Figure 6.9: The knee point concept (Deb, 2011)

Samples of optimization of the four objective functions are depicted in Figure (6.10) to (6.13). This sample represents the results of minimizing both end-to-end latency and end-to-end delay and also maximizing packet throughput and energy efficiency using four algorithms. These algorithms are MOSFP, SPEA2, NSGA-II, and OMOPSO. From the results, the value of end-to-end delay decreases slightly and the value of end-to-end latency decreases sharply until the value of packet payload size reaches 45 bytes. Then, both of them stabilize under 2 when the packet payload size is beyond 45 bytes. The energy efficiency increases slightly until the value of packet payload size reaches 45 bytes and stabilizes above the 0.5 when the packet payload size increases more than 45 bytes. The packet throughput increases slightly until the value of packet payload size reaches 45 bytes and increases dramatically until the value of packet payload size reaches 114 bytes. In these figures, the optimum points are illustrated by different colors (i.e. yellow, blue, red and green), which are the intersection points of all the objective functions. These points are created when the packet payload size equal to 45 bytes.

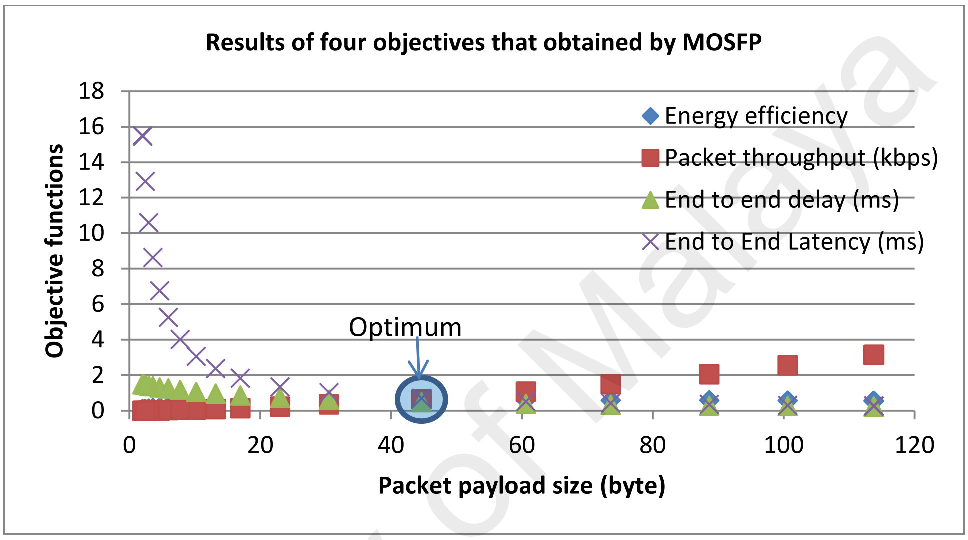


Figure 6.10: Maximizing both energy efficiency and network throughput and also minimizing both end-to-end delay and end-to-end latency based packet payload size that is achieved by MOSFP algorithms

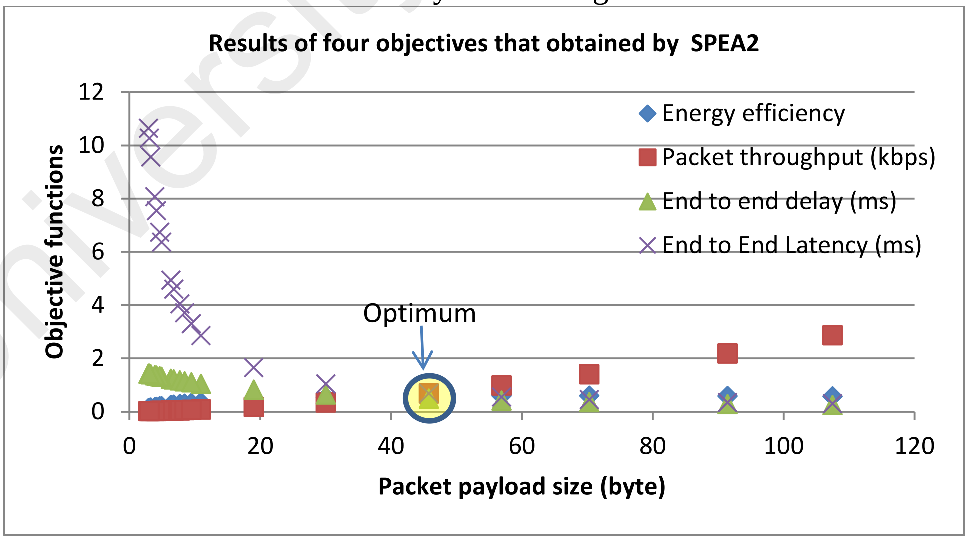


Figure 6.11: Maximizing both energy efficiency and network throughput and also minimizing both end-to-end delay and end-to-end latency based packet payload size that is achieved by SPEA2 algorithms

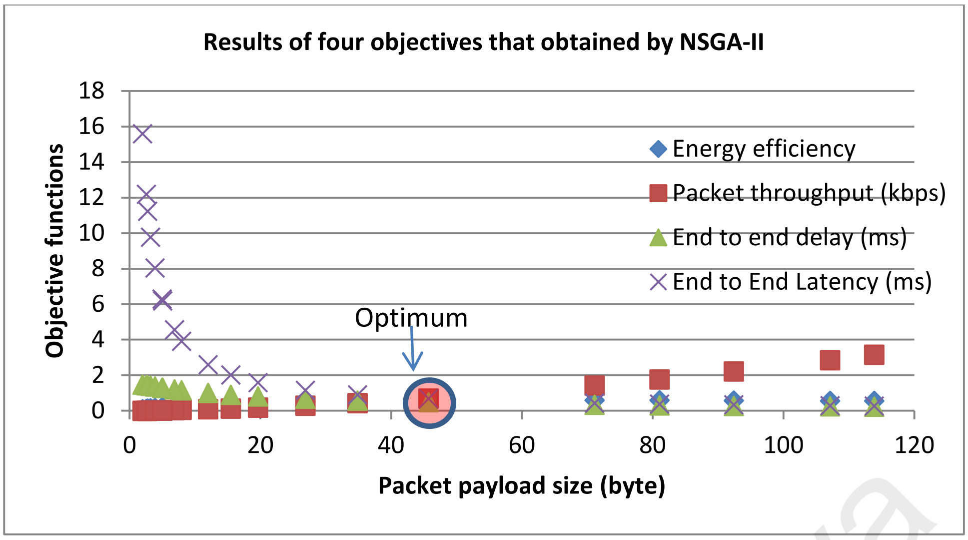


Figure 6.12: Maximizing both energy efficiency and network throughput and also minimizing both end-to-end delay and end-to-end latency based packet payload size that is achieved by NSGA-II algorithms

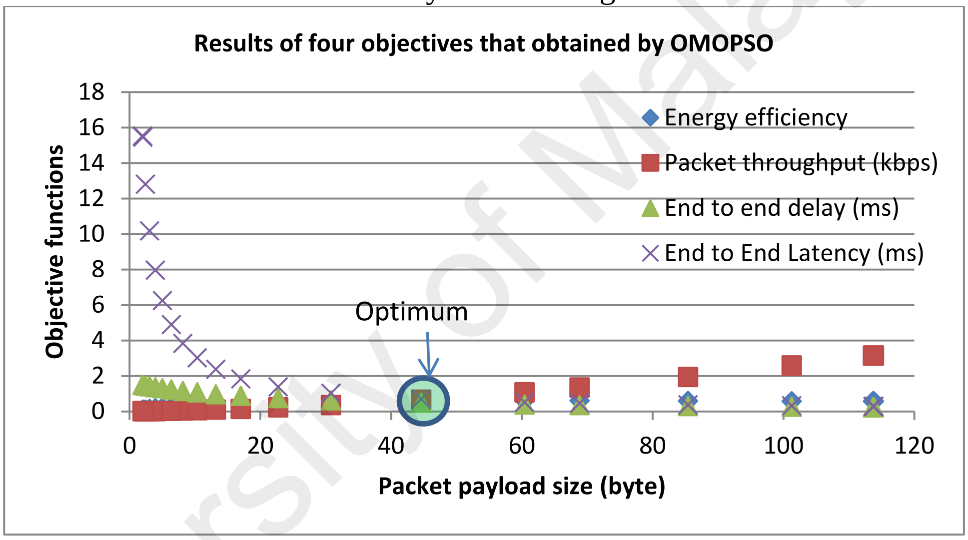


Figure 6.13: Maximizing both energy efficiency and network throughput and also minimizing both end-to-end delay and end-to-end latency based packet payload size that is achieved by OMOPSO algorithms

The Pareto fronts obtained from all the MOOAs at the end of 250 generations are depicted in Figure (6.14). The end-to-end delay, end-to-end latency, network throughput and energy efficiency are denoted by f_1, f_2, f_3, and f_4 respectively. Both OMOPSO and MOSFP generate 19 non-dominated solutions related to Pareto front while both SPEA2 and NSGA-II generate 18 non-dominated solutions for the same objective functions. This demonstrates that MOSFP has a good performance, which obtained 19 values compared to NSGA-II and SPEA2.

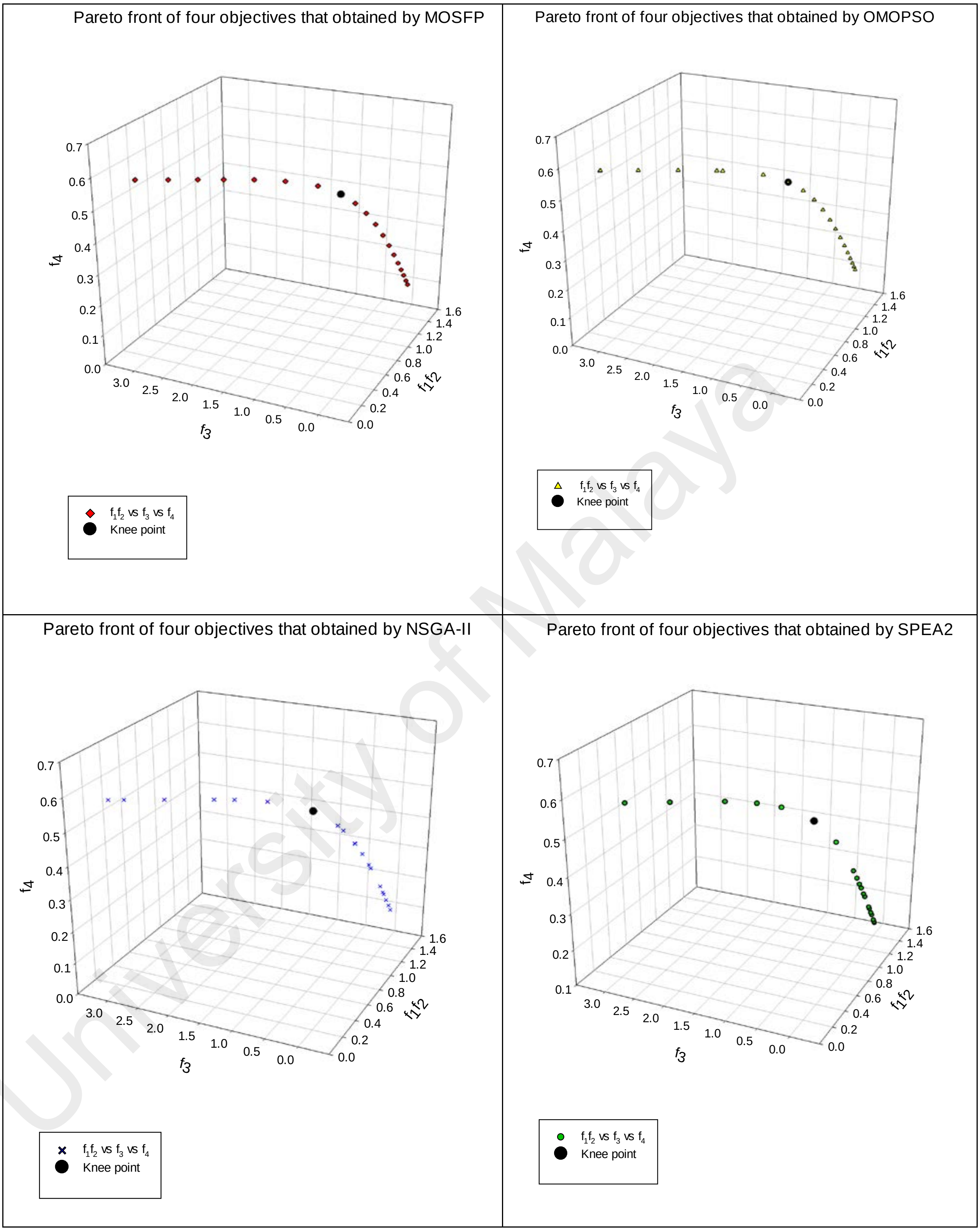


Figure 6.14: Pareto optimal front for end-to-end delay and end-to-end latency (f_1 and f_2) Vs. network throughput (f_3) Vs. energy efficiency (f_4) based on the solutions of all the algorithms

Additionally, MOSFP algorithm has the best distribution and spread of solutions related to the true Pareto front. This followed by the solution of OMOPSO, NSGA-II, and SPEA2 respectively as presented in Figure (6.14). Based on Figure (6.14), the solution spread of both NSGA-II and SPEA2 is weak, which both of them shrink the solutions in one area of the Pareto Front. As illustrated in Figure (6.14), when the network throughput increases, it required more packet payload size. This leads to more latency and delay. The black point on the Pareto front curve represents the knee point. This point represents the optimal point that manages the trade-offs between the objective functions. As explained previously, the optimal point is created when the packet payload size equal to 45 bytes.

6.6 Chapter Discussion

Network modelling is a multi-step procedure to simulate the network challenges and problems in a form of mathematical formulas (objective functions). The network planner and researchers use these models to predict the QoS of the network before the implementation phase. By determining the optimal values of these models, the efficiency and stability of communication link between sender and receiver will be guaranteed.

From the prior works in the literature review, most researchers focused on studying issues related to wireless networks in different topology sizes. Some of them proposed new optimization algorithms and test them based on various network models, while the others study different techniques to increase the network lifetime such as selection technique and task allocation technique. The previous works studied coverage and its impact on the network lifetime. They noted that network lifetime increases when the probability of node coverage increased, also if the task execution time of network nodes decreases, the network lifetime will be increased. However, optimizing network end-to-end delay and end-to-end latency are not studied in previous works. Optimizing these metrics are very important especially for critical network applications such as smart grid network, electrocardiogram and heart pulse monitoring network, and disaster monitoring network. These applications consider the network delay at the top of priorities. Network delay is affected by parameters such as packet size, which consists of the packet header, packet payload size, and packet footer.

Complex computational algorithms have been used to solve and find the optimal value of different kind of real-life problems. However, some of these algorithms such as OMOPSO, NSGA-II, and SPEA2 found it challenging to solve different kinds of objective functions. Examples of these functions are complex objective functions such as, Zitzler-Deb-Thiele 3 (ZDT3) and functions that contain more than two objective functions, such as Walking-Fish-Group 5 and 8 (WFG5 and WFG8). MOSFP has a higher efficiency and the ability to provide an optimal solution for these types of functions as presented in Chapter 5. This is because MOSFP has a higher convergence and spread of the results than OMOPSO, NSGA-II, and SPEA2 while solving these problems. Smart grid is chosen as a case study to demonstrate the ability of our algorithm in solving real-life problems. Also, the smart grid has problems that can be represented in objective functions similar as objective functions that MOSFP has a higher efficiency and the ability to provide an optimal solution for them. Examples of these objective functions are end-to-end delay and end-to-end latency, which the latency is affected by the results of end-to-end delay.

Our proposed algorithm (denoted by MOSFP**)** along with three well-known optimization algorithms such as OMOPSO, NSGA-II, and SPEA2 have been used to optimize a set of network models related to smart grid problems. These models are network end-to-end delay, end-to-end latency, network throughput, and energy efficiency. Packet payload size plays a significant role in determining the results of these models. Packet payload size affects QoS of WSN especially in a dense network. If the network delay increases, the probability of dropped packets will be increased. Hence, retransmitting the dropped packets will consume more energy and time.

The statistical analysis of Tukey's test (one-way ANOVA) between the algorithms is conducted. The statistical analysis demonstrates that our algorithm (MOSFP) significantly outperformed SPEA2 algorithm in optimizing the objective functions while no significant mean difference is noted between MOSFP and both NSGA-II and OMOPSO. However, MOSFP outperformed SPEA2, NSGA-II, and OMOPSO by 51%, 6%, and 3% respectively. In addition, MOSFP obtained the best average value of energy efficiency objective function compared to other algorithms. Energy efficiency is very important to increase the lifetime of the network. In the test of analyzing the Pareto front, the results showed that MOSFP outperformed other algorithms. MOSFP has a good distribution, approximation, and spread of the true Pareto front of the proposed objective functions. This is very clear with the Pareto front that is generated by MOSFP (see Figure (6.14)), which obtained on good results with a good spread and distribution of the solutions rather than, SPEA2 and NSGA-II. The results obtained from four optimization algorithms and the knee points of all the Pareto front samples show that if the packet payload size increase, the network delay, and latency will be increased and both energy efficiency and network throughput will be decreased. Our findings show that 45 bytes are the optimal value of packet payload size that satisfies the trade-offs between all the objective functions. This is based on the knee point and the intersection point of all the objective functions.

6.7 Chapter Summary

1. This chapter study problems and challenges in smart grid especially in HAN.
2. HANs contains sensors embedded in home appliances. These sensors operate via short-range communications using batteries.
3. The misuse of these sensors will lead to rapid death of sensor nodes and reduce the lifetime of the network.

4. Theoretical analysis has been used in this chapter to mitigate the problem, which three well-known algorithms along with our algorithm (MOSFP) have been used to optimize four objective functions related to QoS of any WSN. These models are end-to-end latency, end-to-end delay, energy efficiency, and network throughput.
5. The parameter of packet payload size is used to minimize both end-to-end latency, end-to-end delay, and to maximize both energy efficiency, and network throughput.
6. The statistical analysis using Tukey's test (one-way ANOVA) summarizes that our method (MOSFP) has a significant mean difference with SPEA2. MOSFP outperformed SPEA2 in optimizing the objective functions while no significant mean difference is observed between MOSFP and both OMOPSO and NSGA-II. However, the overall performance of MOSFP outperformed SPEA2, NSGA-II, and OMOPSO by 51%, 6% and 3% respectively.
7. In the test of analyzing the Pareto front, the results showed that MOSFP outperformed other algorithms, which obtained on 19 non-dominated solutions with a good spread and distribution of these solutions rather than, OMOPSO, SPEA2, and NSGA-II.
8. Overall, the intersection point and the knee point of all the Pareto-optimal sets for all the algorithms showed that the optimal value of packet payload size that manages the trade-offs between objective functions is equal to 45 bytes.

CHAPTER 7: CONCLUSIONS

We conclude this thesis by revisiting the objectives outlined in Chapter 1. Then, the work done to achieve the objectives are summarized. The contributions, limitations and future work for this work are also highlighted. This chapter is organized into three sections. Section 7.1 revisits of the objectives of this work. Section 7.2 highlights the contribution of this work and finally Section 7.3 provides future direction for this work.

7.1 Revisiting the Research objectives

Problems of optimization algorithms have been addressed and investigated in this thesis. Five research objectives were outlined in Section (1.4). We revisit these objectives and highlight how the workflow of the research met the objectives.

The first objective was to review the most used single objective optimization algorithms and their multi objective optimization versions in IEEE Xplore and ISI Web of Science databases. A thematic taxonomy along with limitations of prior single objective optimization algorithms such as Particle Swarm Optimization (PSO), Accelerated Particle Swarm Optimization (APSO), Genetic Algorithm (GA), and Parallel Genetic Algorithm (PGA) have been devised to achieve the objective of proposing a new Single Objective Optimization Algorithm (SOOA). Sperm Swarm Optimization (SSO) algorithm has been proposed to achieve the second objective. SSO addresses the convergence issue in previous SOOAs. To test its capability, the proposed SSO was applied to optimize a set of benchmark functions such as Sphere, Rosenbrock, Rastrigin, 2^n Minima, EGGCrate, and Sum Squares functions to achieve the third objective. The results obtained from SSO were compared with the results of PSO, APSO, GA, and PGA. Two metrics were proposed for further comparison namely, quality of results and convergence metrics. The quality of results of SOOA can be determined by comparing between the obtained result of SOOA and the well-known optimal result of an optimization problem (Bianchi, 2009). Convergence is the ability of the algorithm to converge toward optimal result with a determined number of generations ((Ping, 2013), (Pant, 2008)). The results indicate that SSO able to be used as alternative methods. The proposed SSO algorithm outperformed PSO, APSO, GA, and PGA in providing the best average in solving Sphere, Rosenbrock, 2^n Minima, and Sum Squares functions.

The fourth objective was to extend the SSO algorithm to Multi-Objective Optimization Algorithm (MOOA). The proposed algorithm, named as Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP) has been tested on WFG and ZDT benchmark test suites. Its results were compared with the results of three well-known optimization algorithms namely, Non-dominated Sorting Genetic Algorithm (NSGA-II), Optimized Multi-Objective Particle Swarm Optimization (OMOPSO), and Strength Pareto Evolutionary Algorithm 2 (SPEA2). To achieve this, three standard metrics were utilized, i.e. Epsilon, Inverted Generational Distance and Spread. For each algorithm, the maximum generation for each benchmark function is set to 5000. The test for each objective is repeated 100 times to ensure the quality of the results. The proposed MOSFP outperformed both SPEA2 and NSGA-II algorithms in solving all benchmark function suites. MOSFP outperformed OMOPSO in solving the Walking-Fish-Group 5 (WFG5) problems and achieved better solution sets than OMOPSO for the true Pareto front of both Walking-Fish-Group 8 (WFG8) and Zitzler-Deb-Thiele 3 (ZDT3). The high-quality performance and efficiency of MOSFP were reflected on the metrics of IGD and $\in$ of WFG5, and $\in$ of both WFG8 and ZDT3. This indicates that the MOSFP has a better convergence than OMOPSO to discover the search space domain.

The final objective was to optimize a set of WSN network problems. A set of mathematical objective functions have been defined to represent WSN problems. The proposed algorithm was used to minimize both network end-to-end delay and end-to-end latency and also to maximize both network throughput, and energy efficiency. Quantitative and qualitative tests were conducted and the results of the proposed algorithm were compared with the results of other three well-known algorithms such as NSGA-II, SPEA2, and OMOPSO. Overall, in the quantitative test, MOSFP obtained the best average value of optimizing the aforementioned objective functions compared to the other three algorithms. Furthermore, the conducted Tukey's test (one-way ANOVA) signified that our MOSFP has a significant mean difference with SPEA2, which outperformed SPEA2 in optimizing the proposed objective functions while no significant mean difference is observed between MOSFP and both OMOPSO and NSGA-II. However, the overall performance of MOSFP outperformed SPEA2, NSGA-II, and OMOPSO by 51%, 6% and 3% respectively. In addition, the qualitative test of analyzing the Pareto front showed that MOSFP outperformed the other algorithms, which obtained on 19 non-dominated solutions with a good spread and distribution of these solutions rather than OMOPSO, SPEA2, and NSGA-II.

7.2 Contribution of this Work

The contributions of this research to the body of knowledge are summarized in the following points:

- **Sperm Swarm Optimization (SSO) Algorithm:** SSO is a biological nature-inspired algorithm proposed to solve single objective optimization problems. The algorithm is based on the fertilization procedure of female reproductive system. The SSO replicates the sperm movement of going forward in groups from a low temperature zone called Cervix toward the high temperature zone called Fallopian tubes. A comparison between SSO and other well-known algorithms is summarized in Table 7.1.

- **Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP):** MOSFP is proposed as an extended version of SSO to solve the Multi-Objective Optimization Problems (MOOPs). MOSFP operates based on dominance, crowding factor, archive method, and mutation operations to provide an optimal solution for MOOPs. The comparison between MOSFP and other well-known algorithms is summarized in Table 7.1.

Table 7.1: Comparisons between metaheuristic methods

Comparison Criteria	GA, NSGA-II, and SPEA2	PSO and OMOPSO	Proposed SSO and MOSFP
Type of procedure	Discrete procedures	Continuous procedures	Continuous procedures
Type of a metaphor	Darwinian's theory of evolution applied to biology, which simulates the construction of chromosome and its evolution.	Social interaction, which simulates the movement of birds flock while searching for food.	Natural fertilization procedure, which simulates the motility of sperm swarm through the fertilization procedure.
Solutions need ranking and selection	Solutions will be ranked through the evaluations. Selection operator will filter out the population. *Roulette wheel selection* is an example of selection operator in GA.	Solutions will not be ranked through the evaluations. There is no selection operation.	Solutions will not be ranked through the evaluations. There is no selection operation.
Use crossover operation	Use different types of crossover operations such as Simulated Binary Crossover (SBX)	Do not use crossover operations	Do not use crossover operations
Use mutation operation	Use different types of mutation such as polynomial mutation.	OMOPSO uses different types of mutations such as uniform mutation and non-uniform mutation.	MOSFP divides the swarm into three equal parts, after that, performs uniform mutation on the first part and non-uniform mutation on the second part, and also it does not apply any mutation on the third part of the swarm.
Influence of population size or swarm size on solution time	Exponential	Linear	Linear
Population affected by best solution	Deal with each individual independently.	Use the solution of swarm leader (best solution) to add it for other individual solutions.	Use the best solution (the value of winner) as a reference value for other members in the swarm to adjust their velocities
Average fitness value cannot get worse	Average fitness will not be worse because the individual will be ranked from the best to the worse.	Average fitness will not be worse because the velocity of the leader of the swarm (best solution) will be added to all other velocities in	Average fitness will not be worse because all members in the swarm will use the velocity of a

	The best individuals will be reserved for next step while the worst will be eliminated.	the swarm.	winner (optimal solution) as a reference value.
Convergence	Less than PSO, OMOPSO, SSO, and MOSFP.	More than GA, NSGA-II, and SPEA2.	More than GA, PSO, NSGA-II, OMOPSO, and SPEA2.
Ability to find good solution and approximation related to the Pareto front	NSGA-II finds good solution and approximation related to the Pareto front more than SPEA2.	OMOPSO finds good solution and approximation related to the Pareto front more than SPEA2 and NSGA-II	MOSFP finds good solution and approximation related to the Pareto front more than OMOPSO, SPEA2 and NSGA-II

- **Optimize WSN problems:** Problems in WSN have been mathematically modeled to evaluate the QoS of WSN. These models are network end-to-end delay, end-to-end latency, network throughput, and energy efficiency. The models were optimized using our proposed MOSFP algorithm with other three well-known algorithms such as NSGA-II, SPEA2, and OMOPSO. The experimental results indicate that the optimal packet payload size that manages and balances the trade-offs between these objective functions is equal to 45 bytes. Based on that, the real implementation of sensor nodes in smart grid can be configured to packetize this amount of data packets to achieve optimal network QoS of power consumption, network delay and network latency.

7.3 Future Work

A huge amount of efforts and time go into the stage of study. However, a single Ph.D. is honestly never enough to cover all the issues and aspects of any type of research topics. In the following, we highlight the possible future work and directions to extend this research.

- First, the objective functions employed in this work may have their limitations. Other variables that exist during real implementation may affect the outcome of the studies. Therefore, the value of the payload size resulting from this experiment should be tested in the real environment in the future to ensure the reliability of the proposed algorithm.

- Second, the convergence of the algorithm is one of the main aspects that attracts researchers to improve the existing optimization algorithms or to propose a new algorithm based on new idea as we done in this work. Therefore, in future, we will explore the hybridization between our algorithms (SSO) with another optimization algorithm such as (GA). Also, we will use it to optimize problems related to data aggregation ((Mahdi, 2016 a); (Mahdi, 2016b)) in WSN. Hopefully, this will further increase the algorithm convergence.
- Third, we will extend the scope of this research by parallelizing the MOSFP algorithm with another algorithm such as NSGA-II to address the issue of synchronization and a synchronization optimization process.
- Finally, MOSFP will be tested in other important applications such as industrial applications. These applications face problems and require solutions in less efforts at short time. The future works can be summarized in the following figure.

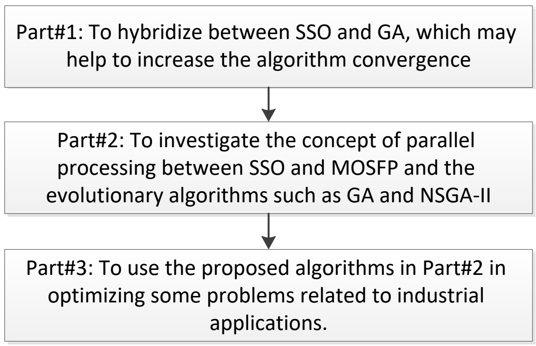


Figure 7.1: Objectives of future works

REFERENCES

Alami, J., & Imrani, A. E. (2008). Dielectric composite multimodal optimization using a multipopulation cultural algorithm. *Intelligent Data Analysis, 12*(4): 359-378.

Alami, J., El Imrani, A., & Bouroumi, A. (2007). A multipopulation cultural algorithm using fuzzy clustering. *Applied Soft Computing*, *7*(2): 506-519.

Ab Aziz, N. A., & Ibrahim, Z. (2012). Asynchronous particle swarm optimization for swarm robotics. *Procedia Engineering*, *41*, 951-957.

AbuNaser, A., Doush, I. A., Mansour, N., & Alshattnawi, S. (2015). Underwater Image Enhancement Using Particle Swarm Optimization. *Journal of Intelligent Systems*, *24*(1): 99-115.

AlQahtani, S. A. (2017). Modeling and performance analysis of unlicensed bands MAC strategy in multi-channel LTE-A networks with M2M/H2H coexistence. *Wireless Networks*, *23*(4): 1-14.

Andréasson, N., Evgrafov, A., & Patriksson, M. (2005). *An Introduction to Optimization: Foundations and Fundamental Algorithms*. Chalmers University of Technology Press.

Acampora, G., Ishibuchi, H. & Vitiello, A. (2014). A comparison of multi-objective evolutionary algorithms for the ontology meta-matching problem. In *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 6 -11 July 2014, pp. 413-420.

Akyildiz, I. F., Melodia, T., & Chowdury, K. R. (2007). Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, *14*(6): 32-39.

Abu-Sharkh, O. M., AlQaralleh, E., & Hasan, O. M. (2017). Adaptive device-to-device communication using Wi-Fi Direct in smart cities. *Wireless Networks*, *23*(7): 2197-2213.

Aguirre, E., Lopez-Iturri, P., Azpilicueta, L., Astrain, J. J., Villadangos, J., Santesteban, D., & Falcone, F. (2016). Implementation and analysis of a wireless sensor network-based pet location monitoring system for domestic scenarios. *Sensors*, *16*(9): 1384.

Ahmad, A., & Hassan, N.U. (Eds) (2016). *Smart grid as a solution for renewable and efficient energy, advances in environmental engineering and green technologies*. IGI global: Hershey, PA, USA.

Al-Anbagi, I., Erol-Kantarci, M., & Mouftah, H. T. (2015). Delay critical smart grid applications and adaptive QoS provisioning. *IEEE Access*, *3*: 1367-1378.

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, *5*(1): 1-35.

Benameur, L., Alami, J., & El Imrani, A. (2009). A hybrid discrete particle swarm algorithm for solving the fixed-spectrum frequency assignment problem. *International Journal of Computational Science and Engineering*, *5*(1): 68-73.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, *8*(2), 239-287.

Bai, Q. (2010). Analysis of particle swarm optimization algorithm. *Computer and information science, 3*(1): 180-190.

Bandyopadhyay, S., & Bhattacharya, R. (2013). On Some Aspects of Nature-Based Algorithms to Solve Multi-Objective Problems. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, Springer Berlin Heidelberg, pp. 477-524.

Bara'a, A. A., Khalil, E. A., & Cosar, A. (2015). Multi-objective evolutionary routing protocol for efficient coverage in mobile sensor networks. *Soft Computing*, *19*(10): 2983-2995.

Bortolotti, B.; Fiorentini, G. (Eds). (1999). *Organized interests and self-regulation: an economic approach*. New York, USA: Oxford University Press.

Baluja, S. (1994). *Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning* (No. CMU-CS-94-163). Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science.

Bahat, A., Caplan, S. R., & Eisenbach, M. (2012). Thermotaxis of human sperm cells in extraordinarily shallow temperature gradients over a wide range. *PLoS One*, *7*(7): e41915.

Bahat, A., & Eisenbach, M. (2006). Sperm thermotaxis. *Molecular and cellular endocrinology*, *252*(1): 115-119.

Borges, S. F., Silva, J. G., & Teixeira, P. C. (2011). Survival and biofilm formation of Listeria monocytogenes in simulated vaginal fluid: influence of pH and strain origin. *FEMS Immunology & Medical Microbiology*, *62*(3): 315-320.

Barone, L., While, L., & Hingston, P. (2002, July). Designing crushers with a multi-objective evolutionary algorithm. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc, pp. 995-1002.

Buche, D., Stoll, P., Dornberger, R., & Koumoutsakos, P. (2002). Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *32*(4), 460-473.

Bhuiyan, M. Z. A., Wu, J., Wang, G., Wang, T., Hassan, M. M. (2017). E-sampling: event-Sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, *12*(1), 1-29.

Beyer, H. G. (2013). *The theory of evolution strategies*. *Springer, New York, NY, Springer Science & Business Media*.

Bezerra, L. C., López-Ibáñez, M., & Stützle, T. (2017, March). An empirical assessment of the properties of inverted generational distance on multi-and many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Cham, pp. 31-45.

Brown, J., & Khan, J. Y. (2012, November). Performance comparison of LTE FDD and TDD based smart grid communications networks for uplink biased traffic. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on* (pp. 276-281). IEEE.

Coello, C, Lamont, G, & Van-Veldhuizen, D. (2013). *Evolutionary algorithms for solving multiobjective problems*, 2nd ed.; Lamont-Veldhuizen, GB.;; New York, NY, USA: Springer, pp. 1-800.

Capitanescu, F., Marvuglia, A., Benetto, E., Ahmadi, A., & Tiruta-Barna, L. (2017). Linear programming-based directed local search for expensive multi-objective optimization problems: Application to drinking water production plants. *European Journal of Operational Research*, *262*(1): 322-334.

Coello Coello, C. A. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, pp. 1051-1056.

Chang, C. Y., & Li, M. H. (2014). A placement mechanism for relay stations in 802.16 j WiMAX networks. Wireless networks, 20(2): 227-243.

Chen, D., Liu, Z., Wang, L., Dou, M., Chen, J., & Li, H. (2013). Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. *Mobile Networks and Applications, 18*(5): 651-663.

Chaudhuri, K., & Dasgupta, D. (2010, January). Multi-objective Evolutionary Algorithms to Solve Coverage and Lifetime Optimization Problem in Wireless Sensor Networks. In *SEMCCO*, pp. 514-522.

Chaudhary, D. K., & Dua, R. L. (2012). Application of multi objective particle swarm optimization to maximize coverage and lifetime of wireless sensor network. *Int. J. Comput. Eng. Res*, *2*(5), 1628-1633.

Christian Nicole. 2013: https://christiannicole72838.wordpress.com/2013/06/26/from-the-figure-for-women-to-understand-their-own-sexual-organs-gender/ (accessed 21 September 2017).

Chamaani, S., Mirtaheri, S. A., Teshnehlab, M., & Shooredeli, M. A. (2007, December). Modified multi-objective particle swarm optimization for electromagnetic absorber design. In *IEEE Applied Electromagnetics, 2007. APACE 2007. Asia-Pacific Conference on*, pp. 1-5.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, *8*(3): 256-279.

Cacciapuoti, A. S., Caleffi, M., Paura, L. (2016). On the probabilistic deployment of smart grid networks in TV white space. *Sensors*, *16*(5): 671.

Carlson, B. M. (2012). *Human Embryology and Developmental Biology E-Book: with Student Consult Online Access*. Elsevier Health Sciences.

Chase, N., Rademacher, M., Goodman, E., Averill, R., & Sidhu, R. (2009). A benchmark study of multi-objective optimization methods. *BMK-3021, Rev, 6*.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1): 29-41.

De León-Aldaco, S. E., Calleja, H., & Alquicira, J. A. (2015). Metaheuristic optimization methods applied to power converters: A review. *IEEE Transactions on Power Electronics*, *30*(12):6791-6803.

Deb, K., & Deb, D. (2014). Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, *4*(1): 1-28.

Devidas, A. R., Ramesh, M. V., & Rangan, V. P. (2016). High performance communication architecture for smart distribution power grid in developing nations. *Wireless Networks*, *22*(3): 1-18.

Deb, K. & Gupta, S. (2011). Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, *43*(11): 1175-1204.

Duan, H., Zhao, W., Wang, G., & Feng, X. (2012). Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO. *Mathematical Problems in Engineering*, *2012*(2012):1-22.

Da Silva Maximiano, M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2012). Multiobjective metaheuristics for frequency assignment problem in mobile networks with large-scale real-world instances. *Engineering Computations*, *29*(2): 144-172.

Dash, P., Saikia, L. C., & Sinha, N. (2015). Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller. *International Journal of Electrical Power & Energy Systems*, *68*, 364-372.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4): 28-39.

Das Neves, J., & Bahia, M. F. (2006). Gels as vaginal drug delivery systems. *International Journal of Pharmaceutics*, *318*(1): 1-14.

Deep, K. (2007, September). A new hybrid self organizing migrating genetic algorithm for function optimization. In *IEEE Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 2796-2803.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2): 182-197.

Dai, C., Wang, Y., & Hu, L. (2016). An improved {\alpha}-dominance strategy for many-objective optimization problems. *Soft Computing*, *20*(3): 1105-1111.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002, May). Scalable multi-objective optimization test problems. In *IEEE Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, pp. 825-830.

Dreżewski, R., & Doroz, K. (2017). An sgent-based co-evolutionary multi-objective algorithm for portfolio optimization. *Symmetry*, *9*(9): 168.

Doudou, M., Djenouri, D., Barcelo-Ordinas, J. M., & Badache, N. (2016). Delay-efficient MAC protocol with traffic differentiation and run-time parameter adaptation for energy-constrained wireless sensor networks. *Wireless networks*, *22*(2):467-490.

Dinh, N., & Lim, S. (2016). Performance evaluations for IEEE 802.15. 4-based IoT smart home solution. *International Journal of Engineering and Technology Innovation*, *6*(4): 274-283.

Datasheet, M. (2006). *MICAz, Wireless Measurement System*. San Jose, California, USA: Crossbow technology inc.

El Imrani, A., Bouroumi, A., El Abidine, H. Z., Limouri, M., & Essaıd, A. (2000). A fuzzy clustering-based niching approach to multimodal function optimization. *Cognitive Systems Research*, *1*(2):119-133.

El-Hamrawy, S., El-Din Fawzy, H., Al-Tobgy, M. (2016). Optimization methods, evolutionary computation optimum design For close range photogrammetry network using particle swarm optimization technique. *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 13(1): 17-23.

Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. *Advances in Engineering Software*, *37*(2): 106-111.

Edmunds, M. W., & Mayhew, M. S. (2013). *Pharmacology for the Primary Care Provider-E-Book*. Elsevier Health Sciences.

Everson, R. M., Fieldsend, J. E., & Singh, S. (2002). Full elite sets for multi-objective optimisation. *In Adaptive Computing in Design and Manufacture V*, Springer, London, pp. 343-354).

Engrand, P. (1998). *A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management*. Electricite de France.

Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons.

Fadel, E., Gungor, V. C., Nassef, L., Akkari, N., Malik, M. A., Almasri, S., & Akyildiz, I. F. (2015). A survey on wireless sensor networks for smart grid. *Computer Communications*, *71*(1): 22-33.

Forrest, S., & Mitchell, M. (1993). What makes a problem hard for a genetic algorithm? Some anomalous results and their explanation. *Machine Learning*, *13*(3): 285-319.

Fogel, D. B. (1997). Evolutionary algorithms in theory and practice. *Oxford University Press*, *Oxford*.

Fan, H. Y., Liu, J., & Lampinen, J. (2010). Some improvement to the mutation donor of differential evolution. *Engineering Computations*, *27*(2): 225-242.

Facemama, (2017): https://www.facemama.com/adn-y-genetica/la-infertilidad-masculina.html (Accessed 19 August 2018).

Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. P. (2011). Smart grid technologies: communication technologies and standards. *IEEE transactions on Industrial informatics*, *7*(4): 529-539.

Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, *18*(2): 327-340.

Goldberg D E, Deb K, & Clark J H (1992). Genetic Algorithms, Noise, and the Sizing of Populations. *Complex Systems*, 6(1): 333-362.

Goldberg, D. E., Deb, K. & Clark, J. H. (1993). Accounting for noise in the sizing of populations. In *L.D Whitley, editor, Foundations of Genetic Algorithms* 2, Morgan Kaufmann, San Francisco, pp 127–140.

Goldberg, D. E., (1989). Genetic algorithms in search, optimization and machine learning. Addison-Wesley (Reading, Mass).

Gharari, R., Poursalehi, N., Abbasi, M., & Aghaie, M. (2016). Implementation of Strength Pareto Evolutionary Algorithm II in the Multiobjective Burnable Poison Placement Optimization of KWU Pressurized Water Reactor. *Nuclear Engineering and Technology*, *48*(5): 1126-1139.

Gutiérrez, J., Villa-Medina, J. F., Nieto-Garibay, A., & Porta-Gándara, M. Á. (2014). Automated irrigation system using a wireless sensor network and GPRS module. *IEEE transactions on instrumentation and measurement*, *63*(1), 166-176.

Guo, W., Healy, W. M., & Zhou, M. (2012). Impacts of 2.4-GHz ISM band interference on IEEE 802.15. 4 wireless sensor network reliability in buildings. *IEEE Transactions on Instrumentation and Measurement*, *61*(9): 2533-2544.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (Eds.). (2013a). *Metaheuristic applications in structures and infrastructures*. Newnes.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, *Addison-Wesley, New York, NY*, 1998.

Gandomi, A. H., Talatahari, S., Yang, X. S., & Deb, S. (2013b). Design optimization of truss structures using cuckoo search algorithm. *The Structural Design of Tall and Special Buildings*, *22*(17): 1330-1349.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013c). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, *29*(1): 17-35.

Gandomi, A. H., & Alavi, A. H. (2011). Multi-stage genetic programming: a new strategy to nonlinear system modeling. *Information Sciences*, *181*(23), 5227-5239.

Gandomi, A. H., Yang, X. S., Alavi, A. H., & Talatahari, S. (2013d). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, *22*(6), 1239-1255.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Deb, S. (2012). Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. *Computers & Mathematics with Applications*, *63*(1): 191-200.

Garcia, J. M. C., Martin, J. R., Gonzales, A. H., & Blanco, P. F. (2004, June). Hybrid heuristic and mathematical programming in oil pipelines networks. In *IEEE Evolutionary Computation, 2004. CEC2004. Congress on*, pp. 1479-1486.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, *76*(2): 60-68.

Gholizadeh, S., & Barzegar, A. (2013). Shape optimization of structures for frequency constraints by sequential harmony search algorithm. *Engineering Optimization*, *45*(6): 627-646.

Guzek, M., Diaz, C. O., Pecero, J. E., Bouvry, P., & Zomaya, A. Y. (2012, December). Impact of voltage levels number for energy-aware bi-objective dag scheduling for multi-processors systems. In *International Conference on Advances in Information Technology*, Springer, Berlin, Heidelberg, pp. 70-80.

Guzek, M., Pecero, J. E., Dorronsoro, B., & Bouvry, P. (2014). Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems. *Applied Soft Computing*, *24*(2), 432-446.

Ganhão, F., Vieira, J., Bernardo, L., & Dinis, R. (2013). Type II Hybrid-ARQ for DS-CDMA: a discrete time markov chain wireless MAC model. In *Internet of*

Things, Smart Spaces, and Next Generation Networking, 1st ed.; Balandin, S.; Andreev, S.; Koucheryavy, Y. (Eds.), Heidelberg, Germany, pp. 176-187.

Hachicha, N., Jarboui, B., & Siarry, P. (2011). A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics. *Information Sciences*, *181*(1): 79-91.

Holland, J. H. (1992). Genetic algorithms. Scientific american, *267*(1): 66-72.

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*(5): 477-506.

Hassanat, A., Alkafaween, E., Al-Nawaiseh, N. A., Abbadi, M. A., Alkasassbeh, M., & Alhasanat, M. B. (2016). Enhancing genetic algorithms using multi mutations. *arXiv preprint arXiv:1602.08313*.

Hassan, R., Cohanim, B., De Weck, O., & Venter, G. (2005, April). A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, p. e21.

Held, G. (2007). *Inter-and intra-vehicle communications*. CRC Press.

Hu, J., Song, J., Zhang, M., & Kang, X. (2008). Topology optimization for urban traffic sensor network. *Tsinghua Science & Technology*, *13*(2), 229-236.

Hamdan, M., Yassein, M.B., & Shehadeh, H.A. (2017). Multi-objective Optimization Modeling for the Impacts of 2.4-GHz ISM band Interference on IEEE 802.15.4 Health Sensors. In *Information Innovation Technology in Smart Cities*, 1st ed.; L. Ismail, L. Zhang, (Eds.), Springer: Singapore, Volume 1, pp.317-330.

Hughes, E. J. (2004, June). Swarm guidance using a multi-objective co-evolutionary on-line evolutionary algorithm. In *IEEE Evolutionary Computation, 2004. CEC2004. Congress on*, pp. 2357-2363.

Health. 2011: http://health-of-people.blogspot.com/2011/03/interestingly-statistics-of-physical.html (Accessed 28 July 2017).

Hamdy, M., Nguyen, A.-T., & Hensen, J. L. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, *121*(2016): 57-71.

Huband, S., Barone, L., While, R. L., & Hingston, P. (2005, March). A Scalable Multi-objective Test Problem Toolkit. In *EMO*, pp. 280-295.

Hamdan, M., Yassein, M. B., & Shehadeh, H. A. (2015). Multi-objective optimization modeling of interference in home health care sensors. In *Innovations in Information Technology (IIT), 2015 11th International Conference on*, Dubai, UAE, 1-3 November 2015, pp. 219-224.

Iturri, P. L., Nazábal, J. A., Azpilicueta, L., Rodriguez, P., Beruete, M., Fernández-Valdivielso, C., & Falcone, F. (2012). Impact of high power interference sources in planning and deployment of wireless sensor networks and devices in the 2.4 GHz frequency band in heterogeneous environments. *Sensors*, *12*(11): 15689-15708.

Ibdah, Y., & Ding, Y. (2017). Path loss models for low-height mobiles in forest and urban. *Wireless Personal Communications, 92*(2):455-465.

JMetal tool. Available online: http://jmetal.sourceforge.net/ (accessed on 2 December 2017).

Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M., & Ismail, M. (2017). Energy-efficient wireless sensor networks for precision agriculture: a review. *Sensors*, *17*(3): 1781-1790.

Jia, J., Chen, J., Chang, G., & Tan, Z. (2009). Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. *Computers & Mathematics with Applications, 57*(11): 1756-1766.

Jia, J., Chen, J., Chang, G., Wen, Y., & Song, J. (2009). Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius. *Computers & Mathematics with Applications, 57*(11), 1767-1775.

Jiang, S., & Cai, Z. (2009). A novel hybrid particle swarm optimization for multi-objective problems. *Artificial Intelligence and Computational Intelligence, 5*(1):28-37.

Jiang, S., Ong, Y. S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics*, *44*(12): 2391-2404.

Joshi, H. I., Pandya, V. J. (2016). Smart power scheduling to reduce peak demand and cost of energy in smart grid. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, *9*(11): 1330-1338.

Jiang, J., Qian, Y. (2016). Distributed communication architecture for smart grid applications. *IEEE Communications Magazine*, *54*(12), 60-67.

Josiński, H., Kostrzewa, D., Michalczuk, A., & Świtoński, A. (2014). The expanded invasive weed optimization metaheuristic for solving continuous and discrete optimization problems. *The Scientific World Journal*, *2014*(2014):1-14.

Kennedy, J & Eberhart, R. (1995). Particle swarm optimization. *In Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 27 November–1 December, pp. 1942–1948.

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. Journal of statistical physics, 34(5): 975-986.

Kennedy, J., Eberhart, R. & Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers, US.

Kennedy, J. (2004). *Swarm intelligence*. In A. Zomaya (Ed.), Handbook of Innovative Computational Paradigms: Biological and Adaptive Computing, 187-220. New York: Springer-Verlag.

Kachitvichyanukul, V. (2012). Comparison of three evolutionary algorithms: GA, PSO, and DE, *Industrial Engineering and Management Systems*, *11*(3): 215-223.

Kumar, K. D. A. (1995). Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems. *Complex Systems*, *9*(1): 431-454.

Kukunuru, N., Thella, B. R., & Davuluri, R. L. (2010). Sensor deployment using particle swarm optimization. *International Journal of Engineering Science and Technology*, *2*(10): 5395-5401.

Kaveh, A., & Talatahari, S. (2011). Hybrid charged system search and particle swarm optimization for engineering design problems. *Engineering Computations*, *28*(4), 423-440.

Khoei, A. R., Keshavarz, S., & Khaloo, A. R. (2010). The genetic algorithm approach for shape optimization of powder compaction processes considering contact friction and cap plasticity models. *Engineering Computations*, *27*(3): 322-353.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, *39*(3): 459-471.

Kaveh, A., & Talatahari, S. (2009). Size optimization of space trusses using Big Bang–Big Crunch algorithm. *Computers & structures*, *87*(17): 1129-1140.

Kaveh, A., & Talatahari, S. (2010). Optimal design of Schwedler and ribbed domes via hybrid Big Bang–Big Crunch algorithm. *Journal of Constructional Steel Research*, *66*(3): 412-419.

Krzeszowski, T., & Wiktorowicz, K. (2016, September). Evaluation of selected fuzzy particle swarm optimization algorithms. In *IEEE Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*, pp. 571-575.

Kabalci, Y. A. (2016). Survey on smart metering and smart grid communication. *Renewable and Sustainable Energy Reviews*, *57*(1): 302-318.

Keyhani, A. (Eds). (2016). *Design of smart power grid renewable energy systems*. John Wiley & Sons: New Jersey, USA.

Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary computation, *8*(2): 149-172.

Li, M., Du, W., & Nian, F. (2014). An adaptive particle swarm optimization algorithm based on directed weighted complex network. Mathematical Problems in Engineering, *2014* (2014):1-7.

Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C. F., & Wang, S. (2017). A review of industrial wireless networks in the context of industry 4.0. *Wireless Networks, 23*(1): 23-41.

Liu, K. (2016). Wireless sensor network target coverage algorithm based on energy saving. *Acta Technica, 61*(3): 49–62.

Langdon, W. B., & Poli, R. (2013). *Foundations of genetic programming*. Springer Science & Business Media.

Lopez-Aguilera, E., Garcia-Villegas, E., & Casademont, J. (2017) Evaluation of IEEE 802.11 coexistence in WLAN deployments. *Wireless Networks*, 23(5): 1-18.

Lagaros, N. D., Plevris, V., & Papadrakakis, M. (2010). Neurocomputing strategies for solving reliability-robust design optimization problems. *Engineering Computations, 27*(7): 819-840.

Li, X., & Yin, M. (2013). An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure. *Advances in Engineering Software, 55*, 10-31.

Li, X., Zhang, J., & Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, *24*(7-8): 1867-1877.

Li, X., & Yin, M. (2012). Application of differential evolution algorithm on self-potential data. *PloS one*, *7*(12): e51199.

Lindroth, P., Patriksson, M., & Strömberg, A. B. (2010). Approximating the Pareto optimal set using a reduced set of objective functions. *European Journal of Operational Research*, *207*(3): 1519-1534.

Lei, J., Fu, G., Yang, L., & Fu, D. M. (2007). Multi-objective optimization design of the YAGI-UDA antenna with an X-shape driven dipole. *Journal of Electromagnetic Waves and Applications*, *21*(7): 963-972.

Lee, Y. H., Cahill, B. J., Porter, S. J., & Marvin, A. C. (2004). A novel evolutionary learning technique for multi-objective array antenna optimization. *Progress In Electromagnetics Research*, *48*(2): 125-144.

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation—GECCO,* Springer Berlin/Heidelberg, pp. 198-198.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, *10*(3): 263-282.

Liang, X., & Balasingham, I. (2007). Performance analysis of the IEEE 802.15. 4 based ECG monitoring network. In *Proceedings of the 7th IASTED International Conferences on Wireless and Optical Communications*, Montreal, Quebec, Canada, May 30 – June 1, pp. 99-104.

Muhlenbein, H. (1992). Evolution in time and space-the parallel genetic algorithm. *In Foundations of genetic algorithms*, *1*(1): 1-22.

Monroy, R., Arroyo-Figueroa, G., Sucar, L. E., & Sossa, H. (Eds.). (2004). MICAI 2004: Advances in Artificial Intelligence. *Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, Springer Proceedings*, pp.1-5.

Meetei, K. T. (2014). A Survey: Swarm Intelligence vs. Genetic Algorithm. *International Journal of Science and Research*, *3*(2):231-235.

Massiani, J. & Picco, G. (2013). The opportunity cost of public funds: Concepts and issues. *Public Budgeting & Finance*, *33*(3): 96-114.

Marinakis, Y., Marinaki, M., & Matsatsinis, N. (2010). A bumble bees mating optimization algorithm for global unconstrained optimization problems. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp.305-318.

Martinez-Sandoval, R., Garcia-Sanchez, A. J., Garcia-Sanchez, F., Garcia-Haro, J., & Flynn, D. A. (2014). Comprehensive WSN-based approach to efficiently manage a smart grid. *Sensors*, *14*(10), 18748-18783.

Mahdi, O. A., Abdul Wahab, A. W., Idris, M. Y. I., Abu Znaid, A., Al-Mayouf, Y. R. B., & Khan, S. (2016a). WDARS: a weighted data aggregation routing strategy with minimum link cost in event-driven WSNs. *Journal of Sensors*, *2016*.

Mahdi, O. A., Abdul Wahab, A. W., Idris, I., Yamani, M., Khan, S., Al-Mayouf, Y. R. B., & Guizani, N. (2016b). A comparison study on node clustering techniques used in target tracking WSNs for efficient data aggregation. *Wireless Communications and Mobile Computing*, *16*(16), 2663-2676.

Naeem, M., Pareek, U., & Lee, D. C. (2012). Swarm intelligence for sensor selection problems. *IEEE Sensors Journal*, *12*(8): 2577-2585.

Nalepa, J., & Kawulok, M. (2014, July). A memetic algorithm to select training data for support vector machines. In *ACM Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 573-580.

Niyato, D., &Wang, P. (2012). Cooperative transmission for meter data collection in smart grid. *IEEE Communications magazine*, *50*(4): 90-97.

Nobre, M., Silva, I., & Guedes, L. A. (2015). Routing and scheduling algorithms for WirelessHARTNetworks: a survey. *Sensors*, *15*(5), 9703-9740.

Patnaik, L. M., & Mandavilli, S. (1996). Adaptation in genetic algorithms. *Genetic algorithms for pattern recognition*, pp.45-64.

Patil, D. D., & Dangewar, B. D. (2014). Multi-Objective particle swarm optimization (MOPSO) based on Pareto dominance approach. *International Journal of Computer Applications*, *107*(4), 13-16.

Paulinas, M. & Ušinskas, A. (2007). A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control, 36*(3): 278-284.

Panigrahi, B. K., Suganthan, P. N., Das, S., & Dash, S. S. (Eds.). (2013). Swarm, Evolutionary, and Memetic Computing: 4th International Conference, SEMCCO 2013, Chennai, India, December 19-21, 2013, Proceedings (Vol. 8297). Springer.

Pinto, M., Gámez, N., Fuentes, L., Amor, M., Horcas, J. M., & Ayala, I. (2015). Dynamic reconfiguration of security policies in wireless sensor networks. *Sensors*, *15*(3): 5251-5280.

Ping Tian, D. (2013). A review of convergence analysis of particle swarm optimization. *International Journal of Grid and Distributed Computing*, *6*(6): 117-128.

Pant, M., Thangaraj, R., & Abraham, A. (2008, September). Particle swarm optimization using adaptive mutation. In *IEEE Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on*, pp. 519-523.

Riquelme, N., Von Lücken, C., & Baran, B. (2015, October). Performance metrics in multi-objective optimization. In *IEEE Computing Conference (CLEI), 2015 Latin American*, pp. 1-11.

Rbouh, I., & El Imrani, A. A. (2014, April). Hurricane Search algorithm a new model for function optimization. In *IEEE Information and Communication Systems (ICICS), 2014 5th International Conference on*, pp. 1-5.

Russell, Stuart J., Norvig, Peter (2003). Artificial intelligence: a modern approach (2nd ed.), *Upper Saddle River, New Jersey: Prentice Hall*, 111–114.

Riko, S., & Andreja, R. (2013). *Intelligent Control Techniques in Mechatronics-Genetic Algorithm*. *Retrieved on*, *3*.

Rane, V. A. (2013). Particle Swarm Optimization (PSO) Algorithm: Parameters Effect And Analysis. *International Journal of Innovative Research and Development, 2*(7): 8-16.

Rout, D. K., & Das, S. (2017). Multiple narrowband interference mitigation using hybrid Hermite pulses for body surface to external communications in UWB body area networks. *Wireless Networks, 23*(2): 387-402.

Rodrigues, J. J., Caldeira, J., & Vaidya, B. (2009). A novel intra-body sensor for vaginal temperature monitoring. *Sensors*, *9*(4): 2797-2808.

Ray, T., & Liew, K. M. (2002). A swarm metaphor for multiobjective design optimization. *Engineering optimization*, *34*(2): 141-153.

Sankarasubramaniam, Y., Akyildiz, I. F., & McLaughlin, S. W. (2003). Energy efficiency based packet size optimization in wireless sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on* (pp. 1-8). IEEE.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3): 221-248.

Shi, Y., & Eberhart, R. C. (1998, March). Parameter selection in particle swarm optimization. In *International conference on evolutionary programming* (pp. 591-600). Springer, Berlin, Heidelberg.

Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved August 17, 2018, *from http://www.sfu.ca/~ssurjano.*

Sörensen, K. (2013). Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, *22*(1): 3-18.

Sakawa, M. (2013). *Fuzzy sets and interactive multiobjective optimization*. Springer Science & Business Media.

Saka, M. P., Hasançebi, O., & Geem, Z. W. (2016). Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation*, *28*(1), 88-97.

Savic, D. (2002). Single-objective vs. multiobjective optimisation for integrated decision support. *Penn State University Press*, *6*(2), 1-7.

Sierra, M. R., & Coello, C. C. (2005, March). Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance. In *Evolutionary multi-criterion optimization*, Berlin, Germany: Springer, pp. 505-519.

Shi, Y. (Ed.). (2012). *Recent Algorithms and Applications in Swarm Intelligence Research*. IGI Global, USA.

Singh, R., Singh, J., & Singh, R. (2016). Security challenges in wireless sensor networks. *International Journal of Computer Science and Information Technology & Security (IJCSITS), 6*(3): 1-6.

Saadon, E. I. S., Abdullah, J., Ismail, N. (2013). Evaluating the IEEE 802.15. 4a UWB physical layer for WSN applications. In IEEE IEEE Symposium on Wireless Technology and Applications (ISWTA), Kuching, Malaysia, 22-25 September, pp. 68-73.

Sagar, A. K., & Lobiyal, D. K. (2014). Coverage and Lifetime Maximization of Wireless Sensor Network with Multi-Objective Evolutionary Algorithm. *International Journal of Scientific & Engineering Research*, *5*(6): 1194-1203.

Sengupta, S., Das, S., Nasir, M., Vasilakos, A. V., & Pedrycz, W. (2012). An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6):1093-1102.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4): 341-359.

Singh, D., Agrawal, S., & Deep, K. (2016). C-SOMAQI: self organizing migrating algorithm with quadratic interpolation crossover operator for constrained global optimization. In *Self-Organizing Migrating Algorithm*, Springer International Publishing, pp. 147-165.

Shirakawa, S., & Nagao, T. (2014, December). Local landscape patterns for fitness landscape analysis. In *Asia-Pacific Conference on Simulated Evolution and Learning*, Springer, Cham, pp. 467-478.

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *IEEE Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, pp. 1945-1950.

Shaw, K. J., Nortcliffe, A. L., Thompson, M., Love, J., Fleming, P. J., & Fonseca, C. M. (1999). Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In *IEEE volutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, pp. 37-45.

Saravanan, R., & Sachithanandam, M. (2001). Genetic algorithm (GA) for multivariable surface grinding process optimisation using a multi-objective function model. *The International Journal of Advanced Manufacturing Technology*, *17*(5): 330-338.

Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, *2017*, 1-25.

Shandilya, S., Thakur, T., Nagar, A.K. (Eds) (2016). *Handbook of research on emerging technologies for electrical power planning, analysis and optimization*. Engineering Science Reference (an Imprint of IGI Global): Hershey, PA, USA.

Talatahari, S., Azar, B. F., Sheikholeslami, R., & Gandomi, A. H. (2012a). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, *17*(3): 1312-1319.

Talatahari, S., Sheikholeslami, R., Shadfaran, M., & Pourbaba, M. (2012b). Optimum design of gravity retaining walls using charged system search algorithm. *Mathematical Problems in Engineering*, *2012*(2012):1-10.

Tung, C. K., Fiore, A. G., Ardon, F., Suarez, S. S., & Wu, M. (2015). Collective dynamics of sperm in viscoelastic fluid. In *APS Meeting Abstracts*, *1*(1), 10-30.

Theophila, B. (2008). *Study of hardware and software optimizations of SPEA2 on hybrid FPGAs*. Rochester Institute of Technology.

Teng, Z., Zhang, M., Jiang, T., Xu, J., & Zhang, Q. (2016). A placement mechanism for relay stations ZigBee routing optimization strategy. *Wireless Networks*, *22*(6): 1869-1875.

Valdez, F., Melin, P., & Castillo, O. (2009). Fuzzy logic for combining particle swarm optimization and genetic algorithms: preliminary results. *MICAI 2009: Advances in Artificial Intelligence*, pp.444-453.

Venkatarayalu, N. V., & Ray, T. (2003, December). Single and multi-objective design of Yagi-Uda antennas using computational intelligence. In *IEEE Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, pp. 1237-1242.

Wang, G. G., Hossein Gandomi, A., Yang, X. S., & Hossein Alavi, A. (2014). A novel improved accelerated particle swarm optimization algorithm for global numerical optimization. *Engineering Computations*, *31*(7): 1198-1220.

Watanabe, K, & Hashem, MMA. (2004). *Evolutionary computations: new algorithms and their applications to evolutionary robots*, 1st ed.; Springer-Verlag Berlin Heidelberg, Springer.

Wang, G., Guo, L., Duan, H., Wang, H., Liu, L., & Shao, M. (2012a). A hybrid metaheuristic DE/CS algorithm for UCAV three-dimension path planning. *The Scientific World Journal*, *2012*(2012):1-11.

Wang, G., Guo, L., Duan, H., Liu, L., & Wang, H. (2012b). A bat algorithm with mutation for UCAV path planning. *The Scientific World Journal*, *2012*.

Wang, G., & Guo, L. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics*, *2013*(2013): 1-22.

Watanabe, S., Hiroyasu, T., & Miki, M. (2001). Parallel evolutionary multi-criterion optimization for mobile telecommunication networks optimization. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN*, pp.167-172.

Wang, J., Ren, X., Chen, F. J., Chen, Y., & Xu, G. (2016). On MAC optimization for large-scale wireless sensor network. *Wireless Networks*, *22*(6), 1877-1889.

Xiong, G., Chen, C., Kishore, S., & Yener, A. (2011, January). Smart (in-home) power scheduling for demand response on the smart grid. In *Innovative smart grid technologies (ISGT), 2011 IEEE PES* (pp. 1-7). IEEE.

Yassein, M. B., Hamdan, M. M., & Shehadeh, H. A. (2016). Performance evaluation of health monitoring network for elderly patient in home. *Asian J Math Comput Res*, *9*(2): 108-118.

Yassein, M. B., Hamdan, M., Shehadeh, H. A., & Mrayan, L. (2017). A Novel approach for health monitoring system using wireless sensor network. *International*

Journal on Communications Antenna and Propagation (IRECAP), 7(4): 271-285.

Yan, X., Du, H., Ye, Q., & Song, G. (2016). Minimum-delay data aggregation schedule in duty-cycled sensor networks. *In International Conference on Wireless Algorithms, Systems, and Applications, Springer International Publishing,* Bozeman, Montana, USA, pp. 305-317.

Yue, X., Guo, Z., Yin, Y., & Liu, X. (2016). Many-objective E-dominance dynamical evolutionary algorithm based on adaptive grid. *Soft Computing*, *3*(4):1-10.

Yang, S., Ong, Y. S., & Jin, Y. (Eds.). (2007). *Evolutionary computation in dynamic and uncertain environments*. Springer Science & Business Media.

Yu, R., Zhang, Y., Gjessing, S., Yuen, C., Xie, S., & Guizani, M. (2011). Cognitive radio based hierarchical communications infrastructure for smart grid. *IEEE network* 2011, *25*(5): 6-14.

Yang, J., Zhang, H., Ling, Y., Pan, C., & Sun, W. (2014). Task allocation for wireless sensor network using modified binary particle swarm optimization. *IEEE Sensors Journal*, *14*(3): 882-892.

Yang, X. S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. (Eds.). (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. Newnes.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing. 2009. NaBIC 2009. World Congress on* (pp. 210-214). IEEE.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74.

Zingg, D. W., Nemec, M., & Pulliam, T. H. (2008). A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique, 17*(1-2): 103-126.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *In Proceedings of the EUROGEN'2001 Eurogen, Evolutionary Method for Design: Optimization and Control for Industrial Problem,* Barcelona, Spain, 19–21 September 2001; pp. 1–21.

Zhao, X., Gao, X. S., & Hu, Z. C. (2007). Evolutionary programming based on non-uniform mutation. *Applied Mathematics and Computation, 192*(1): 1-11.

Zhang, W., & Liu, Y. (2004, June). Reactive power optimization based on PSO in a practical power system. In *IEEE Power Engineering Society General Meeting, 2004. IEEE*, pp. 239-243.

Zitzler, E., Laumanns, M., & Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization. *Metaheuristics for multiobjective optimisation*, 3-37.

Zheng, X., & Liu, H. (2010). A scalable coevolutionary multi-objective particle swarm optimizer. *International Journal of Computational Intelligence Systems*, *3*(5): 590-600.

Zhou, J., Wang, C., & Zhu, J. (2016). Multi-objective optimization of a spring diaphragm clutch on an automobile based on the non-dominated dorting genetic slgorithm (NSGA-II). *Mathematical and Computational Applications, 21*(4): 47.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

Shehadeh, H.A., Ahmedy, I., & Idris, M.Y.I. (2018). Sperm Swarm Optimization Algorithm for Optimizing Wireless Sensor Network Challenges, in ACM *International Conference on Communications and Broadband Networking (ICCBN 2018)*, Singapore, 24-26 February 2018.

Shehadeh, H.A., Idris, M.Y.I., & Ahmedy, I. (2017). Multi-Objective Optimization Algorithm Based on Sperm Fertilization Procedure (MOSFP). *Symmetry*, *9*(10), 241. (Q2, Impact factor: 1.457).

Shehadeh, H.A., Idris, M.Y.I. Ahmedy, I. Ramli, R., & Noor N.M. (2018). MOSFP Method for Solving Wireless Sensor Networks Optimization Problems in Smart Grid Applications, *Energies*, *11*(1), 97. (Q2, Impact factor: 2.262).

Shehadeh, H. A. Ahmedy, I. & Idris, M.Y.I. (2018). Empirical Study of Sperm Swarm Optimization Algorithms, *in book: Volume 869 of the Advances in Intelligent Systems and Computing series Edition: Volume 869 Chapter: K. Arai et al. (Eds.): IntelliSys 2018, AISC 869*, 2019, Publisher: Springer International Publishing Switzerland.

- Submitted and Accepted Articles in Other Research Area:

Ahmedy, I., Shehadeh, H. A., & Idris, M.Y.I. (2017). An Estimation of QoS for Classified Based Approach and Nonclassified Based Approach of Wireless Agriculture Monitoring Network Using a Network Model. *Wireless Communications and Mobile Computing*, *2017*. (Q3, Impact factor: 1.9)

(Submitted) Shehadeh, H. A. Ahmedy, I. & Idris, M.Y.I. (2019). Multi-Objective Optimization Modeling of Near Ground VHF/UHF Radio Communication Network. 2019 *International Conference on Mathematics, Science and Technology Teaching and Learning (ICMSTTL 2019)*, Sydney, Australia.