RINGED SEAL SEARCH FOR GLOBAL OPTIMIZATION VIA A SENSITIVE SEARCH MODEL

YOUNES SAADI

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

2018

RINGED SEAL SEARCH FOR GLOBAL OPTIMIZATION VIA A SENSITIVE SEARCH MODEL

YOUNES SAADI

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

FACULTY OF COMPUTER AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Younes Saadi**

Matric No: **WHA130024**

Name of Degree: **DOCTOR OF PHILOSOPHY**

Title of ~~Project/Research Report/Dissertation/~~Thesis ("this Work"): **RINGED SEAL SEARCH FOR GLOBAL OPTIMIZATION VIA A SENSITIVE SEARCH MODEL**

Field of Study: **SOCIAL NETWORK**

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:

RINGED SEAL SEARCH FOR GLOBAL OPTIMIZATION VIA A SENSITIVE SEARCH MODEL

ABSTRACT

This thesis proposes a nature-inspired metaheuristic algorithm for global optimization. The proposed algorithm, which is called Ringed Seal Search (RSS), is inspired from the movement of the animal ringed seal. The proposed algorithm is characterized by a search model namely the sensitive search model, where the exploitation-exploration is adaptively balanced. The quality of the algorithm is comprehensively evaluated on various standard benchmark test functions using variety of quality metrics and using three baseline algorithms for comparison. The time consumption analysis shows that RSS consumes less time compared to its homologs. This result is compatible with the convergence analysis. The solution quality analysis demonstrates that the convergence speed of RSS obtained better solution quality, which can be interpreted as a mature search. The diversity evaluation shows that the proposed algorithm achieved an optimal diversity values in most of the benchmark test functions. The experimental results show that the proposed algorithm in this thesis improves the global optimization quality in uni-objective and multi-objective environments while the exploitation and exploration are adaptively balanced. Finally, the proposed algorithm is applied on a data clustering case study using seven benchmark datasets to validate and check its ability to solve real optimization problems. The obtained results show that the proposed algorithm can be used for data clustering.

Keywords: Metaheuristics, Global Optimization, Exploitation, Exploration.

PENCARIAN BERKELILING ANJING LAUT UNTUK OPTIMIZATION GLOBAL MELALUI MODEL PENCARIAN SENSITIF

ABSTRAK

Tesis ini mencadangkan algoritma metaheuristik yang diilhamkan oleh alam untuk pengoptimuman global. Algoritma yang dicadangkan, yang disebut Ringed Seal Search (RSS), diilhamkan dari pergerakan meterai cincin haiwan. Algoritma yang dicadangkan dicirikan oleh model carian iaitu model carian yang sensitif, di mana eksploitasi eksplorasi disesuaikan secara seimbang. Kualiti algoritma dinilai secara komprehensif ke atas pelbagai piawai kenanda aras bagituangsi ujian pelbagai metrik kualiti dan menggunakan tiga algoritma asas untuk perbandingan. Analisis penggunaan masa menunjukkan bahawa RSS menggunakan kurang masa berbanding homolognya. Hasil ini bersepadacan dengan analisis konvergensi. Analisis kualiti penyelesaian menunjukkan bahawa kelajuan konvergensi penumpuan RSS memperoleh kualiti penyelesaian yang lebih baik, yang boleh ditafsirkan sebagai carian yang matang. Penilaian kepelbagaian menunjukkan bahawa algoritma yang dicadangkan mencapai nilai kepelbagaian yang optimum dalam kebanyakan fungsi ujian penanda aras. Hasil eksperimen menunjukkan bahawa algoritma yang dicadangkan dalam tesis ini meningkatkan kualiti pengoptimuman global dalam persekitaran yang uni-objektif dan multi-objektif manakala eksploitasi dan eksplorasi adalah seimbang. secara adaptif Akhir sekali, dengan menggunakan algoritma yang dicadangkan pada kajian kes kelompok data yang menggunakan tujuh penanda aras data set untuk mengesahcan dan menyemak keupayaannya untuk menyelesaikan masalah pengoptimuman sebenar. Hasil yang diperoleh menunjukkan bahawa algoritma yang dicadangkan boleh digunakan untuk pengelompokan data.

Kata kunci: Metaheuristik, Pengoptimuman Global, Eksploitasi, Eksplorasi.

ACKNOWLEDGEMENTS

All my thanks goes to all people involved in this process of growth as a being that thinks.

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Tutut Herawan and Dr. Vimala Balakrishnan who gave me the opportunity to be their student. I am deeply grateful to my supervisors for the continuous support of my PhD study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me a lot in all the time of research and writing of this thesis. Their efforts really mean a lot to me. Their extensive discussions around my work and interesting exploration in this field have been very helpful for this thesis.

I would like express my personal appreciation to Dr. Suraya Binti Hamid for her guidance and support.

I dedicate this dissertation to the memory of my father, who would have been happy to see me achieving this stage. I also owe my loving thanks to my family especially my beloved mother. They have lost a lot due to my research abroad. Without their encouragements and understanding it would have been impossible for me to finish this work. My special gratitude is due to my brothers and sisters for their loving support.

Lastly, I would like to thank all my friends, all postgraduate students, staff in Faculty Computer Science and Information Technology and Postgraduate Center for their support, cooperation and contribution along the way. Thank you all very much.

The financial support of the High Impact Research Center is hereby gratefully acknowledged.

Younes, January 2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

P	:	A probability parameter
GA	:	Genetic Algorithm
PSO	:	Particle Swarm Optimization
CS	:	Cuckoo Search
Levy	:	Levy walk
Brownian	:	Brownian walk
2D	:	Two dimension
AIS	:	Artificial Immune System
FA	:	Firefly Algorithm
ACO	:	Ant Colony Optimization
RSS	:	Ringed Seal Search
EC	:	Evolutionary Computation
SI	:	Swarm Intelligence
x	:	Variable
v	:	Velocity
t	:	Time
c_{i}	:	Acceleration coefficient
r	:	Random vector
x(A)	:	Upper bound
x(B)	:	Lower bound
G_{i}	:	Global Optimum
P_i	:	Best particle
P_α	:	Probability factor
n	:	Host nests

U	:	Distribution value
T	:	Candidate solution
λ	:	Occurrence of the event
n	:	Number of events
μ	:	Mean
σ	:	Standard Deviation
β	:	Index
$L(s)$	:	Levy distribution
s	:	Random step
$Step_N$	:	The next state
ω	:	Step size
Ω	:	Search space
ρ	:	State of the space
Δx	:	State of the search space
α	:	Step size
k	:	Standard deviation
d	:	Dimension
$Ndots$	:	Number of particles
L^{best}	:	Best lair
L_i	:	Lair
f_*	:	Global optimal solution
AB	:	Average Best
MB	:	Median Best
SD	:	Standard Deviation
UCI	:	University of California Irvine dataset

Var	:	Variance
SQ	:	Solution Quality
DB	:	Davies–Bouldin index
DI	:	Dunn Index
RI	:	Rand Index
TP	:	Total Number of True Positive
TN	:	Total Number of True Negative
FP	:	Total Number of Positive Values
FN	:	Total Number of False Negative values

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1. Background

Global optimization consists of searching and finding an optimal solution in a specific search space. As humans, everyday several types of optimization are faced such as finding the shortest route for a city or setting of the most optimal location of furnishments in a school. The capability to find a better solution grows when the search for many possible solutions is increased. This mechanism is called exploration and it leads to cover the whole search possibilities. In contrast, the capability to search for possible solutions near the existing solution is called exploitation (Damaševičius & Woźniak, 2017; Mafarja & Mirjalili, 2017). A good search requires a balance between the exploitation and the exploration of the search (Mirjalili, 2016; Yang, Deb, & Fong, 2014). The number of possible solutions for an optimization problem increases when the dimension of the problem (number of routes, number of furnishments) is augmented. Solving optimization problems with high dimension is time consuming and it requires using of mathematical approaches to tackle the huge number of possibilities (Mirjalili, Jangir, & Saremi, 2017; Rao & Waghmare, 2017).

Metaheuristic approaches are proposed to solve global optimization problems even in high dimension within a reasonable time (Yazdani & Jolai, 2016). These approaches start by using of candidate solutions, where a heuristic search is applied to find new better solution than the existing solution. The process of search for new solutions only requires information on how to measure the fitness of a candidate solution. Genetic Algorithm (GA) is considered as one of the most popular approaches (Srinivas & Patnaik, 1994). It uses operators inspired by natural genetic variation and natural selection (Eberhart & Kennedy, 1995; Knysh & Kureichik, 2010; Srinivas & Patnaik, 1994; Yang & Deb, 2014). Particles Swarm Optimization (PSO) was inspired by the fish and bird swarm intelligence (Eberhart & Kennedy, 1995); on the other hand, Firefly Algorithm (FA) was inspired by the flashing pattern of tropical fireflies (Alba & Dorronsoro, 2005; Bianchi, Dorigo, Gambardella, & Gutjahr, 2009; Blum, Puchinger, Raidl, & Roli, 2011; Blum & Roli, 2003; Bonabeau, Dorigo, & Theraulaz, 1999). The Cuckoo Search (CS) was inspired by the brood intelligent behavior of some cuckoo species. Its strategy consists of laying its eggs in other cuckoos' nests (Yang, 2010b).

Metaheuristics are characterized by several exploitation-exploration parameters, which have impacts on the search performance (Sorensen, Sevaux, & Glover, 2017). For most approaches, there are default exploitation-exploration parameters, which are tuned up and generally used. However, a parameter tuning might work well for a particular problem but not so well on another (S. Deb, Fong, & Tian, 2015; Neumüller, Wagner, Kronberger, & Affenzeller, 2012). It is shown that predefining parameters is primordial for every new problem instance (Birattari & Kacprzyk, 2009; Smit & Eiben, 2009). However, the process of finding the suitable parameters is difficult and it is related to the nature of the optimization problem as well (Črepinšek, Liu, & Mernik, 2013; Fagan & van Vuuren, 2013). Although metaheuristics exist since long time, but an adaptive balance between exploitation and exploration which is able to work well on most optimization problems has not been introduced yet in the literature. According to Yang (2010b), tuning an exploitation-exploration balance for any given metaheuristic is a complex problem itself. Therefore, in order to avoid tuning the balance, a metaheuristic approach requires having the exploitation-exploration to be adaptively balanced. The key elements of this research challenge are as follows.

i. **Screening the nature of the problem** to find which characteristics influence the exploitation-exploration balance.

ii. **Screening the parameters of exploitation and exploration** to find which parameters influence the performance of metaheuristics.

iii. **Determining** the relationship between the nature of the problem, exploitation-exploration parameters and the performance of metaheuristics.

iv. **Adaptively balance** exploitation-exploration of a metaheuristic for a given optimization problem.

v. **Assessing performance** of the metaheuristic algorithm to variations where exploitation-exploration is adaptively balanced when applied and tested to different benchmark optimization problems.

The main barriers to tackle this research challenge are as follows.

- **Problem nature space.** The large number of metaheuristics, make determining the problem characteristics difficult (Gogna & Tayal, 2013; Piotrowski, Napiorkowski, Napiorkowski, & Rowinski, 2017).
- **Exploitation-Exploration parameters space.** There is huge number of parameters to be considered (Črepinšek et al., 2013; Karafotias, Hoogendoorn, & Eiben, 2015; Yang et al., 2014).
- **Performance metrics.** Performance should be measured in terms of convergence speed, time consumption, diversity, maturity and solution quality (Fagan & van Vuuren, 2013; Jamil & Yang, 2013).
- **A case study scenario.** Applying an algorithm with an exploitation-exploration adaptively balanced on a real application scenario.

These barriers and challenges must be a part of any new proposed metaheuristic approach in this research. Without tackling the problems related to these barriers and challenges, the proposed metaheuristic is not applicable to all optimization problems as tuning the exploitation-exploration balance cannot be addressed easily. So the main question here is how these challenges can be dealt?

An examination of the research journals shows that the majority of metaheuristics are nature-inspired and their design is based on nature phenomenon, where the largest category of nature-inspired algorithms is based on animal behavior. It is known that balancing the search consists of finding the adequate ratio between exploitation and exploration that can lead to the optimal solution efficiently (Črepinšek et al., 2013; Saeed Saremi & Sejnowski, 2016). However, there is no specific rule for how to balance exploration and exploitation (Yang et al., 2014). In this context, huge number of metaheuristic approaches based on animal behavior have been introduced such as Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), Ant Colony Optimization (ACO) (Dorigo, Birattari, & Stützle, 2006), Grey Wolf Optimizer (GWO) (Mirjalili, Mirjalili, & Lewis, 2014), Cuckoo Search (CS) (Yang & Deb, 2009), Lion Optimization Algorithm (LOA) (Yazdani & Jolai, 2016) and Elephant Search Algorithm (ESA) (S. Deb et al., 2015). Yang et al. (2014) emphasized this by saying, "The key components in metaheuristic algorithms for global optimization are local intensive exploitation and global diverse exploration, and their interaction can significantly affect the efficiency of a metaheuristic algorithm". Many researchers believe that metaheuristics based on animal behavior are effective, because their search mechanisms inspired from animal movement have balance mechanisms for exploitation and exploration (Fister Jr, Yang, Fister, Brest, & Fister, 2013). However, they are known to be hard to tune up due to their stochastic nature (Eiben & Schippers, 1998).

While these limitations in the literature are justified, it is clear that there is no alternative solution proposed in the literature to tune up the balance between exploitation and exploration. In this thesis, a new metaheuristic algorithm is presented where exploitation-exploration is adaptively balanced. This idea combines field of study and animal movement models to develop the suitable movement that can adaptively balance exploitation and exploration.

1.2. Problem Statement

Researchers perceive metaheuristic algorithms as a great source for solving global optimization problems, finding global optimum a key issue but very difficult one to solve because there is no specific rule to search for global optimal solutions (BoussaïD, Lepagnot, & Siarry, 2013; Juan, Faulin, Grasman, Rabe, & Figueira, 2015; Yang et al., 2014; Yazdani & Jolai, 2016). Searching for solutions must be balanced between exploitation and exploration (Cuevas, Echavarría, & Ramírez-Ortegón, 2014; Shahrzad Saremi, Mirjalili, & Lewis, 2017; Yang et al., 2014). Therefore, a metaheuristic algorithm might not provide optimal solutions if tuning the exploitation-exploration balance is not applied (Ben Ghalia, 2008; Črepinšek et al., 2013; Cuevas et al., 2014). The exploitation-exploration tuning can provide good results for a particular problem but cannot be applied on another as argued by Črepinšek et al. (2013) and (Islam, Li, & Mei, 2017) as tuning exploitation-exploration is highly sensitive to the nature of the optimization problem; thus, require lot of time and difficult to handle. Very few researchers have been found to incorporate nature-inspired approaches for the purpose of balancing exploitation-exploration tuning while considering both animal search techniques and tuning exploitation-exploration balance. Animal search techniques require complete knowledge on modeling animal movement (Bastille-Rousseau et al., 2017). It was stated that modeling an animal movement is a very difficult task because it is not easy to understand the mechanisms that drive the movement (Bartumeus, Da Luz, Viswanathan, & Catalan, 2005). The stochastic models are used to model animal movements, where the impact of exploitation-exploration is not easy to quantify and approximate (Benhamou, 2007) . A weak exploitation-exploration balance is expected to occur, however, evaluating the impact of exploitation-exploration is extremely difficult as stated in (Črepinšek et al. (2013); Karafotias et al., 2015), and even if a particular stochastic model provides a balanced search, the impact on another optimization problem could differ (Črepinšek et al., 2013). Thus, there is a need for new techniques for modeling animal movement to be proposed to handle exploitation-exploration balance.

Despite the significance of stochastic search models, very few studies have been found in the domain of metaheuristics that involved random search models. The predominant models used in this domain for modeling animal movement are: Levy walk (Benhamou, 2007), Ballistic walk and Brownian walk (Nolting, 2013; Nurzaman et al., 2011), where these stochastic techniques assume a random search featured by a power-law distribution, which make them appropriate for modeling animal movement (Nolting, 2013). However, they are inappropriate for tuning exploitation-exploration balance due to the unique pattern of the search. The research gap of this research is based on the fact that most of the studies used trial and error based on predefined parameters for tuning the exploitation-exploration balance, whereas the techniques are time consuming, very complex and may result in premature convergence and loss of diversity (Benhamou, 2007; Črepinšek et al., 2013; Karafotias et al., 2015; Nolting, 2013).

In view of the research gap and the limitations of random search models in handling exploitation-exploration separately, there is a need for modeling the search to balance exploitation-exploration. Moreover, the capabilities of composite search models and random search models are the most appropriate techniques for the projection of modeling animal movement while considering the impact of balancing exploitation-exploration adaptively. The proposed model will be statistically evaluated, and compared with other baseline approaches such as GA, PSO and CS.

1.3. Aim of Research

The aim of this research is to apply composite search to model an animal movement to build a global optimization algorithm that can have an adaptive balance between exploitation and exploration while considering the convergence, solution quality, diversity, time consumption and maturity of the search. In this research, ringed seal movement will be used as an animal movement model.

1.4. Research Objectives

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

i. To develop a composite search model based on ringed seal movement to adaptively balance exploitation and exploration.
ii. To derive a metaheuristic algorithm for solving global optimization based on the composite search model.
iii. To test, validate and compare the effectiveness of the metaheuristic algorithm in terms of time consumption, solution quality, convergence, maturity and diversity of the search.

1.5. Research Questions

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

i. How does a composite search model be developed based on ringed seal movement to adaptively balance exploitation and exploration?
ii. How does a metaheuristic algorithm for solving global optimization is derived based on the composite search model?
iii. What would be the optimum way to test the developed metaheuristic algorithm?

1.6. Research Motivation

This research was triggered by the limitations of random search models on metaheuristics, dependence of exploitation-exploration balance tuning on the search model and the nature of the optimization problem (Benhamou, 2007; Črepinšek et al., 2013; Thomas T. Hills, Todd, Lazer, Redish, & Couzin, 2015; Kazimierski, Abramson, & Kuperman, 2015; Yang et al., 2014). These factors prompted urgent need in order to have a relatively reliable metaheuristic search adaptively balanced that will allow solving optimization problems. Similarly, metaheuristic algorithms can achieve better optimal solutions based on a balanced exploitation-exploration. The performance of metaheuristic applications such as deep learning applications, scheduling applications, routing applications, ranking problems and forecasting applications heavily depend on exploitation-exploration balance (Shahrzad Saremi et al., 2017; Soler-Dominguez, Juan, & Kizys, 2017; Yang & Deb, 2014). Therefore, a reliable metaheuristic algorithm based on a balanced exploitation-exploration can assist these applications to achieve high performance.

1.7. Research Design

The research starts by conducting a literature review of the advances made in the exploitation-exploration tuning approaches, and unveils limitations in the literature. The limitations discovered in the previous works are the research problems of the present study. The problem statement was formulated based on the limitations. The research aims and objectives were derived from the problem definition.

The design of the composite search is inspired from the ringed seal search (RSS) and implemented using a composite of Brownian walk and Levy walk. The Brownian walk is used for exploitation whereas Levy walk for exploration. The preliminary experiments were conducted to check whether the search model fulfills the requirements of exploitation-exploration balance or not. The process of building the metaheuristic algorithm namely RSS were derived from the composite search model.

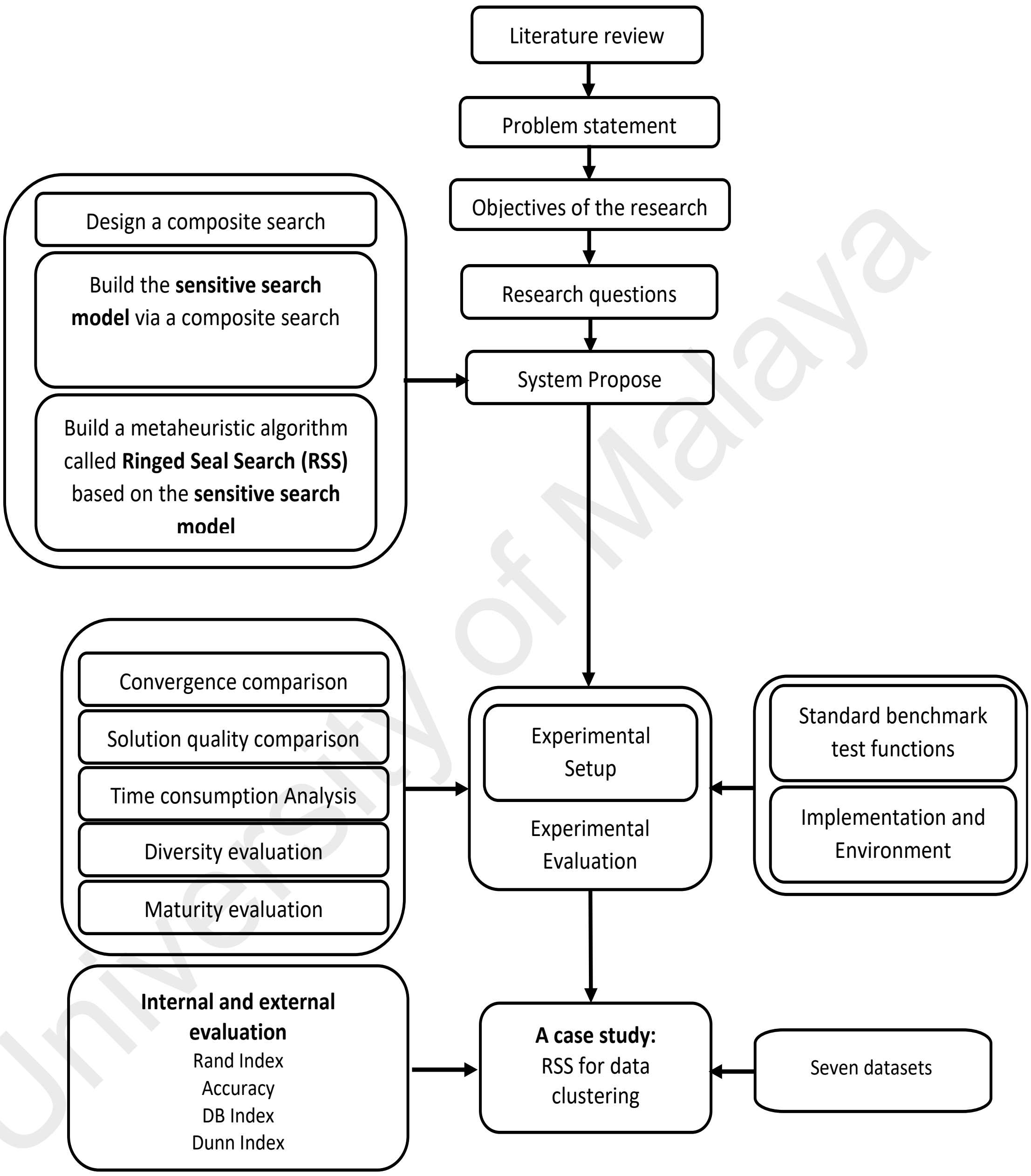


Figure 1.1: Proposed research design

For the purpose of evaluation, a set of fifteen benchmark test functions were used to measure the capability of the RSS to solve global optimization problems. The obtained results were compared to other baseline approaches such GA, PSO and CS. The convergence, diversity, solution quality and time cost were assessed using the outputs of the benchmark test functions to explore the significance among the RSS which is based on adaptive balance and other baseline approaches. The entire process of the research design is presented in Figure 1.1.

1.8. Research Contributions

The contributions of this research are described as follows:

1. A new composite search model called the sensitive search model featured by an adaptive balance between exploitation and exploration.
2. A new metaheuristic algorithm, RSS, for global optimization is derived from the composite search model.
3. The number of parameters to tune the proposed algorithm (RSS) is restricted to one parameter, making RSS to be less sensitive to parameters settings compared to PSO, GA and CS.
4. An extensive evaluation based on five metrics: time consumption, convergence, diversity, solution quality and maturity. Moreover, a validation based on a clustering optimization problem is also introduced.

1.9. Significance of the Study

The research unveiled an alternative approach to exploitation-exploration tuning, which has added to the approaches already discussed in the literature review with significant impact on several domains as follows:

1.9.1. To the Machine Learning Community

Based on the applications of metaheuristics in machine learning and deep learning (De Rosa, Papa, & Yang, 2017; Fong, Deb, & Yang, 2018), the proposed algorithm is expected to provide a significant solution, where solving optimization problems will not require exploitation-exploration tuning and trial and error process. The RSS approach can improve computational efficiency and solution quality over the machine learning approaches such as improving neural network back propagation computation time (Gudise & Venayagamoorthy, 2003), optimizing deep learning parameters (Fong et al., 2018; Young, Rose, Karnowski, Lim, & Patton, 2015) and increase the solution quality of partitional clustering (Nanda & Panda, 2014). The research work contributes to the present effort being made in improving solution quality and computation time in neural networks by proposing a more autonomous adaptive approach that may actually be applied on different problems. Moreover, the research work contributes to the efforts related to parameter selection (Young et al., 2015) where finding the optimal parameters for a particular optimization problem is time consuming and requires lot of testing.

1.9.2. To the Metaheuristic Optimization Approaches

The research can advance the metaheuristic parameter optimization especially automating the balance between exploitation and exploration with results that were statistically validated. The proposed metaheuristic approach with few parameters tuning have the potential to work well on several optimization problems without any need for tuning. The proposed approach might be better applied on finding optimal balance parameters better than the existing tuning approaches.

1.10. Scope of the Research

The focus of this research is to build a metaheuristic algorithm using a composite search model inspired by ringed seal movement and characterized by an adaptive balance between exploitation and exploration. A total of fifteen benchmark test functions were used as benchmark. They are divided into two groups: uni-objective and multi-objective. Uni-objective test functions will be used to test smooth problems, and multi-objective test functions will be used to test problems with many local optima. The measurements adopted for the purpose of evaluating the performance of the proposed method are convergence speed, solution quality, time consumption, diversity and maturity analysis.

1.11. Thesis Outline

Chapter One

This chapter provides the background of the research including problem statement, objectives, aims, scope, significance of the studies, research questions, motivation, and brief explanation about the proposed research design.

Chapter Two

The background concept and theoretical foundations of metaheuristics is presented in the chapter to provide solid preliminary information to the readers, where the applications of search techniques based animal movement in metaheuristics are discussed. A detailed overview of the related work in domain of metaheuristics is presented as well. Moreover, the previous attempts to handle exploitation-exploration balance are discussed. A comparative study between search models is presented to select the suitable models to build the proposed algorithm.

Chapter Three

The focus in this chapter is on establishing the direction of research that takes into account the issues decorticated in Chapter two. The proposed approach which consists of modeling the movement of the ringed seal by using a composite search model is introduced. Diagrammatical and pictorial representation of the proposed composite search model and the derived metaheuristic algorithm are also introduced. Finally, benchmark test functions and the related characteristics are explained.

Chapter Four

The obtained results are discussed. The experimental results are justified for each class which provides a good analysis for the efficiency of the proposed approach. The results consist of comparing performance of the benchmark test functions over the baseline metaheuristic algorithms in terms of convergence, diversity, maturity, solution quality and time consumption.

Chapter Five

In this chapter, a case study is introduced to validate the proposed metaheuristic algorithm and show that it can be used to solve real data clustering problems, where a total of seven benchmark datasets are applied to measure several clustering metrics.

Chapter Six

Chapter Six of the thesis covered conclusions derived from the empirical findings, and further research to be conducted in the future. Contributions and limitations made by the study are highlighted.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

This chapter provides an introduction about metaheuristic algorithms, relevant notions; concepts and fundamental theories including exploitation-exploration and search techniques based on animal movement were presented. Moreover, the chapter reviews the related work and what has been introduced using these techniques and concepts, where the impact of exploitation-exploration balance on providing optimal solutions is explained. The literature is synthesized based on strengths and weaknesses of the search techniques based on animal movement, and best search models were chosen for comparison purposes.

2.2. Metaheuristics for Global Optimization

Every optimization problem is characterized by a set of possible solutions called search space, in which the space is bounded by upper and lower bound (BoussaïD et al., 2013). The solution can be defined as an input for a function, where the function in the domain of global optimization is called the objective function. A solution quality is defined as the output of a given input via an objective function. The global optimal solution of a particular problem is declared found if there is no other input which will provide better output (Yang & Deb, 2014). The best quality can be defined as the minimum or maximum, it depends on the nature of the optimization problem. The difficulty of solving a global optimization problem depends on several factors such as the nature of the problem and dimensionality which describes the number of candidate solutions in the search space. It is known that linear optimization problems are solved by using an expression of several terms, where the objective function is featured by a simple linear function. Moreover, global optimization problems that are defined as nonlinear problems are classified into two categories: uni-objective problems and multi-objective problems, where multi-objective problems are much difficult to solve compared to uni-objective problems (Deb, Sindhya, & Hakanen, 2016).

It is shown that there is no a specific algorithm that can solve global optimization problems exactly in a deterministic way as practice in linear problems (Floudas & Pardalos, 2014; Horst & Tuy, 2013) . To find the solution of a global optimization problem, it is possible to use a backtracking algorithm to compute all possible solutions (Civicioglu, 2013). However, covering the whole search space is time consuming and not feasible. In order to handle this issue, random search techniques based on iterative improvement are proposed (Spall, 2005). The idea consists of improving the search results based on a method called problem-independent heuristic search or metaheuristics, where the solution quality is based on the manipulation of the search. Metaheuristics consist of a set of iterative random steps that formulate at the end a stochastic distribution (Juan et al., 2015). The following figure illustrates the general taxonomy of metaheuristics.

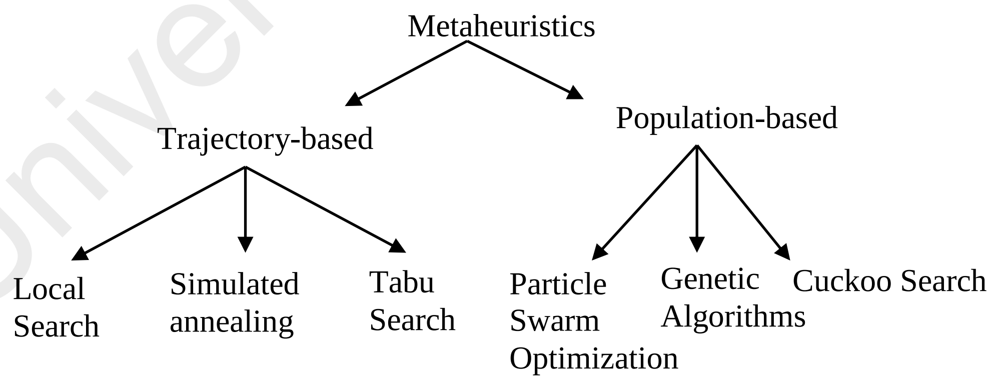


Figure 2.1: Taxonomy of metaheuristics (Affenzeller, Wagner, & Winkler, 2008)

Figure 2.1 shows the taxonomy of metaheuristic approaches in a general context. The following sections will focus only on population-based approaches since they are the most popular metaheuristics (Fong et al., 2018). Population based metaheuristics start with a set of solutions and they apply a search for better set of solutions. The most widely used population-based techniques are related to Evolutionary Computation (EC) and Swarm Intelligence (SI) (Engelbrecht, 2006; Fogel, 2006). EC approaches are inspired from the Darwinian theory and its ability to modify a population of individuals by recombining and mutating them (Cuevas, Osuna, & Oliva, 2017). On the other hand, SI is based on the idea of social interaction between individuals, rather than purely individual cognitive abilities (Blum & Li, 2008). Several methods have been introduced, local methods to find the local optimum and global methods to find the global optimum (Auger, Schoenauer, & Teytaud, 2005). This section introduces the analysis of global optimization approaches, in particular: Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Cuckoo Search (CS) because they are the most popular metaheuristics that have demonstrated their merits (Fong et al., 2018).

2.2.1. Genetic algorithms

A Genetic Algorithm (GA) is a metaheuristic search inspired from natural evolution phenomenon existing in nature since billions of years (Holland, 1992). It is considered as the most popular evolutionary computation method, and is modeled to create new solutions to be used in optimization and search problems. GAs are part of population-based algorithm where optimization problems are solved by generating solutions based on techniques inspired from the natural evolution. Particularly, generating solutions is based on many steps, such as mutation, selection, and crossover. The best organism in term of fitness for the current generation carries on genes to the next generations. The main principle of genetic is included in the role of the operators (crossover and mutation), in which a new type is introduced through the change in the structure of the gene. Reproduction, crossover and mutation can be considered as the fundamental steps in a GA (T. Li, Shao, Zuo, & Huang, 2017).

The GA starts with an initial number of random solutions called population defined in a specific search space namely chromosomes. At each generation, a new set of solutions is created based on the fitness of the selected individuals (David E Goldberg, 2013). This process allows creating a population evolution for the individuals that are better adapted to their ecosystem than the individuals that they were generated from. Individuals are encoded as strings, chromosomes, comprised of some alphabet, so the chromosome values are represented with a unique map in the decision variable domain. Generally, a binary alphabet is the most used representation in GAs. However, there are other types of representations used, such as: integer, ternary etc (Deng, Liu, & Zhou, 2015). As an example, let x_1 and x_2 are two variables of a specific problem, could be represented onto the chromosome structure as shown in Figure 2.2 below:

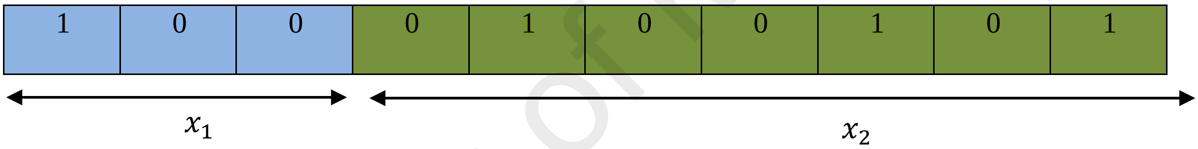


Figure 2.2: Chromosome structure

The variable x_1 is represented with 3 bits and x_2 is represented with 7 bits, this representation reflects the accuracy interval of the individual decision variables. No information about the problem can be obtained from the analysis of the isolated nature of the chromosome string (David E. Goldberg, 1989).

Decoding of the chromosome is a primordial step in order to extract any possible meaning from the representation. As described below, the search process operates on the representation (encoding) of the decision variables, instead of the feasible region and taking into consideration one exception, i.e. the case of using real valued genes. After decoding the chromosome representation into the feasible region, it will be possible to evaluate the fitness of individual members of the population. The evaluation is calculated via an objective function that computes the fitness of the individual in the constraint (Michalewicz, 2013). In nature, this can be defined as the capacity of individuals to survive inside their own territories. Selecting pairs of individuals to be mated in the reproduction stage is based on fitness evaluation calculated by the objective function (David E Goldberg, 2013). The fitness value is used in selection stage to converge towards individuals with high fitness. These individuals with high performance have a high probability to be selected during mating process. On the other hand, individuals with low fitness have a low probability to be selected.

After evaluating the fitness values of the individuals, fit individuals can be selected from the population, after then recombined to produce a new generation. Genetic operators control the features of the chromosomes, taking into consideration that the gene code of some individuals generates high fitness values. The operator exchanges information between pairs of individuals (T. Li et al., 2017). As an example, consider the binary string pair as shown in the following Figure.

$\mathcal{P}_1$	1	1	0	0	0	1	0	0	1
$\mathcal{P}_2$	1	0	0	1	0	0	1	1	1

Figure 2.3: A binary string pair

From Figure 2.3, if a position i is selected uniformly in randomness between 1 and a length $S-1$, the exchange of genetic information is done at this selected point. As a result, a new offspring strings is obtained. Figure 2.4 shows example describes two new offspring are generated during the crossover at a selected point $i=4$.

$New\ \mathcal{O}_1$	1	1	0	0	0	0	1	1	1
$New\ \mathcal{O}_2$	1	0	0	1	0	1	0	0	1

Figure 2.4: New offspring strings

From Figure 2.4, the crossover operation is applied on some strings of the population with a probability P_x. After then, another operator is used for mutation with probability P_m. The mutation process consists of changing the genetic representation of the individual by using specific probabilistic rules. For the binary string, mutation consists of changing the state of a single bit, from 0 to 1, or the inverse from 1 to 0. Particularly, mutation role consists of ensuring that the probability for obtaining a specific subspace of the constraint is never zero. Thus, mutation effect tends to prevent converging to a local optimum instead of the global optimum. At the end, individual strings decoded in order to evaluate the objective function. As a result, fitness values of individuals are computed and then individuals are selected for mating using their fitness. The same process is applied for upcoming generations. Following this process, best individuals are carried on to the next generation, however worse ones die out. The stopping criterion terminates the genetic process once some conditions are satisfied.

```
begin
        t = 0;
        initialize P(t);
        evaluate P(t);
        while (stop creterion)

                t = t + 1;
                select P(t) from P(t-1);
                reproduce pairs in P(t);
                evaluate P(t);
        end while
end
```

Figure 2.5: A Basic genetic Algorithm, introduced by (David E Goldberg, 2013)

Figure 2.5 shows a basic GA introduced by Goldberg (David E Goldberg, 2013). It is used to clarify the morphology of GA and its main steps. The algorithm uses a time dependent variable p, where the population is initialized randomly, at $t=0$ being $P(0)$ .The above description of GA, shows that the algorithm has many potentialities compared with other optimization algorithms:

1) Genetic algorithm is a parallel search based population.
2) Genetic algorithm is only based on the objective function and fitness values.
3) Genetic algorithm is based on stochastic transition rules.

It is very important to highlight that GA has great potentialities in term of providing solutions. However, the user always has the choice to select the final solution. Furthermore, GA process contains many steps, which increases the complexity and the computation time and thus the final cost. In some situations, problem nature cannot accept one individual solution, as in the case of multi-objective optimization problems. In this case, the GA strategy consists of exploring the parallelism potentialities to provide alternative solutions simultaneously.

2.2.2. Particle Swarm Optimization

Particle swarm optimization (PSO) is one of the mostly used techniques for optimization problems introduced by J. Kennedy and Eberhart (1995) . The main idea of this technique is inspired from animal swarm groups; where there is no dominance or leaders for specific elements. It consists of mimicking the behavior of animals that have no leadership, in which these animals use a random method to find food (Ben Ghalia, 2008). Particularly, a herd of animals such as birds, have no guides, they use a random search to locate food, in such a way the swarm always follows members with the nearest position to the food source. The herds get the best situation in parallel via the interaction with other members who already got a situation with better values. This scenario happens in repeated iterations until the best location of food is found. The PSO algorithm consists of swarm particles, each particle is considered as a potential solution.

Swarm particles have an enormous ability to explore the search area. They have two main mechanisms: the capability to memorize best local position and also the ability to get neighbors best position (Pedersen & Chipperfield, 2010). The position of a particle is changing according to the velocity. When generating a new solution $x_i(t+1)$, for say a particle i in the search area at a specific time t, the particle performs a movement calculated using a velocity $v_i(t)$ to the current particle position

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{2.1}$$

where $x_i(0) \sim U(x_{\min}, x_{\max})$ and the velocity $v_i(t)$ is obtained as follows:

$$v_i(t) = v_i(t-1) + c_1 r_1 (\text{localbest}(t) - x_i(t-1)) + c_2 r_2 (\text{globalbest}(t) - x_i(t-1)), \tag{2.2}$$

where c_1 and c_2 are acceleration coefficient, r_1 and r_2 are random vectors. The following example describes PSO with a simple demonstration (Ben Ghalia, 2008). Min $f(x)$, where $x(B) \le x < x(A)$, where $x(A)$ the upper bound and $x(B)$ is the lower bound. The PSO algorithm can be summarized into 3 stages: at the first stage, the size of the particles group is initialized with N. The value of N should not be too small or too big in order to ensure that there are many candidate solutions around the best solution. At the second stage, generate an initial number of population x taking into consideration the upper and lower bounds in order to ensure that the total of the particle is included in the search area (Du & Swamy, 2016b; Pedersen & Chipperfield, 2010). Then, the particle j and velocity at i are specified, $x(i)$ and $v_j(i)$. The initial particles are defined as below:

$$\text{Initial\_Pa rticles} = [x_1(0), x_2(0), \cdots, x_n(0)], \tag{2.3}$$

where $x_j(0)$ is the vector coordinates of the particle with

$$j = [1, 2, \cdots, n]. \tag{2.4}$$

The evaluation of the objective function values can be denoted by the following syntax:

$$f[x_1(0)], f[x_2(0)], \cdots, f[x_n(0)]. \tag{2.5}$$

Initially, at iteration $i = 1$, all the particle velocities are set to zero. At the i^{th} iteration, the particles achieve new coordinates with specific velocity values. After that, the PSO algorithm computes the optimal coordinate of particle j at iteration i and find $P_{best}(j)$. According to the objective, the algorithm specifies the lowest or the highest objective function value. The coordinates of particles are calculated using equation 2.1. After that, calculate the best value among all particles $x_j(i)$ found at iteration i. As a result, G_{best} will be selected as the minimum or maximum among all particles function values at i number of iterations. The calculation of velocity for the particle j at iteration i is done using the formula 2.2. The parameters c_1 and c_2 are two cognitive parameters used to represent the social effect between one particle group members . However, r_1 and r_2 are random numbers belong to the range $[0,1]$. Finally, at the last stage, verify whether the current position (solution) is convergent or not. In case the positions of all particles are toward the same value, it is a convergence case. At the opposite case, where all the particles are not leading to an equal value, another iteration is run, calculate new values for $P_{best}(j)$ and G_{best}. Generally, there are five different approaches for setting a stopping criterion (Du & Swamy, 2016b; Gao, Du, & Yan, 2015; James Kennedy, 1999; J. Kennedy & Eberhart, 1995; Khan, Yang, Wang, & Liu, 2016; Mendes, Kennedy, & Neves, 2004).

The first approach based on maximum number of iterations. The Second based on finding a specific position. The third approach based on evaluating the performance of consecutive iterations and stop once there is no improvement. The fourth approach consists of stopping the algorithm process once the normalized swarm radius is close to zero. The fifth approach consists of a stopping criterion that is used to terminate the algorithm once an objective function descent is approximately equal to zero.

In certain cases, using specific stopping criterion doesn't allow to know whether the particle able to cast on local optima, local minima, global optima or global minima. In the basic PSO, there is a lack of solution (Pedersen & Chipperfield, 2010), as PSO can get the local optima easily. In some particular cases, the new position obtained by the particle is equal to the global best, and thus the particle will not be able to change the position (Du & Swamy, 2016b). In case of the particle is the global best of all the swarm, all the other particles will converge in the same direction of the best particle. Therefore, the swarm moves early to the local optimum. Once the particle's new position is not close to the global best and local best, the velocity will be increased rapidly into a high value. This affects the position of the particle during the next stages. As a result, the particle will achieve a position with high value, which may cause going out of the search area. In analysis, PSO has advantages and disadvantages (Du & Swamy, 2016b; Pedersen & Chipperfield, 2010). Advantages of the basic PSO can be described as follows:

1. PSO is based on the intelligence and it can be applied into both scientific research and engineering use.
2. PSO have no overlapping and mutation calculation.
3. In PSO, the search can be carried out by the speed of the particle.
4. During the development of several generations, only the most optimal particle can transmit information onto the other particles, and the speed of the researching is very fast.

5. The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability.
6. PSO adopts the real number code, and it is decided directly by the solution.
7. The number of the dimension is equal to the constant of the solution.

On the other hands, disadvantages of the basic PSO algorithm can be summarized in three points:

1. The PSO method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.
2. The PSO method cannot work out the problems of scattering and optimization
3. The PSO method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

2.2.3. Cuckoo Search Algorithm

Cuckoo Search (CS) is a population-based nature-inspired algorithm developed by Yang and Deb (2009). It is inspired by the brood intelligent behavior of some cuckoo species. Its main strategy consists of searching for a nest that belongs to another bird species to lays his eggs inside it. The cuckoo strategy also consists of distributing his eggs amongst different numbers of nests. This parasitic behavior is very advantageous to ensure the reproduction of new generation whatever the circumstances. This unique and deceptive breeding behavior pattern is the main skeleton of cuckoo search algorithm for solving optimization problems. CS formulates the eggs in the nest as a set of candidate solutions for specific optimization problem. Each cuckoo egg is a new candidate solution. The main target of this representation is to use the new solutions (new eggs) to replace the current solutions. After a specific number of iterations, a lot of solutions will be replaced, resulting in the best possible solution for the problem. CS algorithm is based on the following rules (Fister, Yang, Fister, & Fister, 2014).

- Each cuckoo bird produces one egg at a time and put it in a selected nest.
- The nests with high quality of eggs will qualify to the next generations.
- The total number of host nests is constant, where each host can detect a fake egg with a probability $p_\alpha \in [0,1]$. The host bird has two choices: throw the egg out of the nest or migrate the current nest and construct a new nest in a new place elsewhere. Based on the rules above, the fundamental skeleton of CS can be outlined in the pseudo code shown in Figure 2.6.

begin

Objective function $f(x), x = (x_1, \ldots, x_d)^T$

Generate initial population of n host nests $x_i \, (i = 1,2\ldots,n)$

while *(t<MaxGeneration) or (stop criterion)*

Get a cuckoo randomly by Levy flights

evaluate its quality/fitness F_i

Choose a nest among n (say, j) randomly

If $F_i > F_j$

Replace j *by the new solution;*

end

A fraction p_α *of worse nests are abandoned and new ones are built;*

Keep the best solutions (or nests with quality solutions);

Rank the solutions and find the current best

end while

Postprocess results and visualization

end

Figure 2.6: Cuckoo search algorithm proposed by Yang et Deb (Yang & Deb, 2009)

From Figure 2.6, the CS algorithm starts with an initial number of n host nests generated randomly inside a specific search space. Then, the CS algorithm dedicates one egg randomly for the host nests using a Levy flight. If the fitness F_i (the quality) of cuckoo egg is better than the fitness of the egg of the host bird F_j, the cuckoo egg will survive and it will not be detected by the host bird. In this case the fitness of the egg will replace the fitness of the whole nest, thus the notion of producing new better solution is applied (Yang & Deb, 2014). If the cuckoo egg fitness F_i is less than host bird egg fitness F_j, the cuckoo egg will be detected and destroyed by the host bird. The CS algorithm ranks the nests and selects the best one. A fraction of worse nests p_α is neglected and new nests are built. If the stop criterion is not satisfied, CS algorithm starts a new iteration by dedicating cuckoo eggs to new nests randomly. When generating a new solution $x^{(t+1)}$ for, say, a cuckoo i, a Levy flight is performed (Yang & Deb, 2009).

$$x^{(t+1)} = x^t + \propto \oplus \text{Levy}(\lambda). \tag{2.6}$$

The search behavior of CS is done via a Levy flight. Basically, the term Levy flight is used to represent a random walk characterized by a specific step size (random number). It is the length between two consecutive portions of direction, which is calculated from a probability distribution with an inverse power-law tail described as below (Edwards et al., 2007; Viswanathan et al., 1999; Viswanathan, Raposo, & Da Luz, 2008).

$$\text{Levy} \sim u = t^{-\lambda}, \tag{2.7}$$

where $1 < \lambda < 3$ and t is the flight length. In fact, the generation of random numbers with Levy flights particularly consists of two main steps: the selection of a random direction and the calculation of step size which is in conformity with Levy distribution. Selecting a random direction is calculated through a uniform distribution. The authors of cuckoo search proposed an approach based on Mantegna's algorithm which is characterized by a symmetric distribution, where positive and negative steps values are considered (Fister et al., 2014; Yang, 2010b). The probability distribution of a random variable, is considered stable if the linear combination of its two identical copies (or U_1 and U_2) conforms to the same distribution. Namely, $aU_1 + bU_2$ and $cU + d$ have the same distribution value, where $a, b > 0, c, d \in R$. According to Mantegna's algorithm, the step length can be calculated as below (Yang, 2010b):

$$s = \frac{u}{|v|^{1/B}}, \tag{2.8}$$

where u and v are obtained from a normal distributions

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2), \tag{2.9}$$

where

$$\sigma_u = \left\{ \frac{\Gamma(1+B)\sin\left(\pi B/2\right)}{\Gamma\left[(1+B)/2\right] B 2^{(B-1)/2}} \right\}^{1/B}, \sigma_v = 1 \tag{2.10}$$

For a quick look, it seems that the local search in CS is relatively weak and that may affect the optimization output. Therefore, the search model used in CS doesn't take into consideration environment interaction. In reality, bird's flight direction is affected by many different obstacles and sometimes too sensitive to nature factors, such as predators, wind and etc. Neglect of external factors may affect the search direction and thus the finding new solutions. For CS, from a processing time point of view, neglect of external factors could be justified by the reason that changing the search direction due to unexpected fact could take time to get a new solution from the search space, and thus affect the processing time for the algorithm. However, that doesn’t reflect the real scenario in nature, where changing the direction caused by noise or threat is a part of the random movement itself.

2.3. Global Optimization

Global optimization consists of finding the point x that minimizes or maximizes a function $f : X \rightarrow T$ called the objective function, where X denotes a specific search space and T denotes a set of candidate solutions. A local optimum (local optimization) of a function is a point where the function value is smaller than or equal to the value at nearby points, but possibly bigger than at a distant point (Birattari, 2009; Törn *et al*, 2008). A global optimum (global optimization) is a point where the function value is smaller than or equal to the value at all other feasible points (Törn *et al* , 2008; Liberti, 2008). The following figure illustrates a function f defined over a two-dimensional search space.

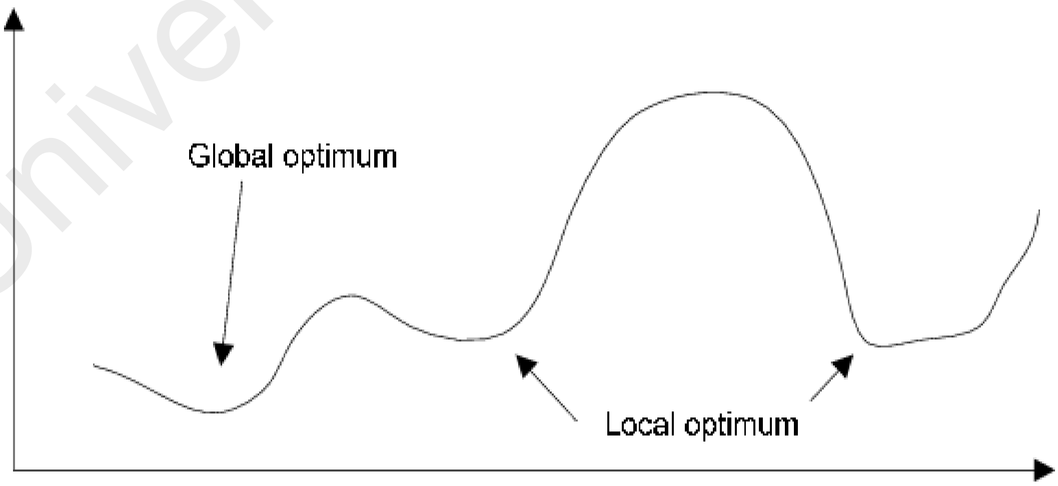


Figure 2.7: A search space illustrates global and local optimum (Slak, Tavčar, & Duhovnik, 2014)

From Figure 2.7, it is easy to distinguish between local and global as outlined there. A local optimum is an optimum of one of its local areas, where the local maximum to overcome is denoted as the highest point that the search progress can achieve within a local area. In contrast, a global optimum is an optimum of the whole search space.

One of the solutions used to solve global optimization problems consists using of stochastic methods, which are characterized by the capability to deal with random choices of the next step of search. Moreover, stochastic methods are featured by two separate phases: exploitation and exploration (Bartumeus, Raposo, Viswanathan, & da Luz, 2014) . At each phase a specific pattern of search is performed and a particular number of step sizes are realized with several random directions.

2.4. Exploitation-Exploration

Metaheuristic algorithms have become ubiquitous and vital to solve global optimization problems for numerous applications in science and technology. Every metaheuristic algorithm needs to fulfill the requirements of exploitation and exploration of the search space. Exploration represents the capability of the algorithm to perform an extensive search to cover the whole search space. However, exploitation consists of searching for the optimum solution around the current visited point. Too much of exploitation can result in local optima problem, in contrast too much of exploration can affect a slow convergence and loss of diversity (Yang, 2011). Therefore, a good metaheuristic search requires to be balanced between exploration and exploitation. Balancing the search consists of finding the adequate ratio between exploitation and exploration that can lead to the optimal solution efficiently (Črepinšek et al., 2013). However, there is no specific rule for how to balance exploration and exploitation (Yang et al., 2014).

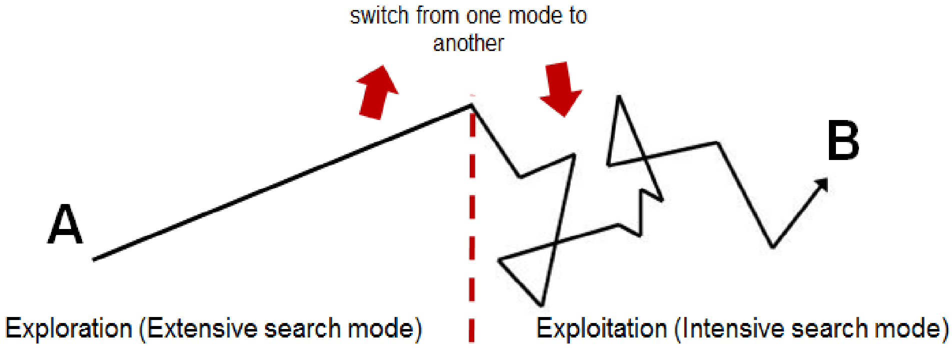


Figure 2.8: An example of exploitation-exploration

Figure 2.8 describes an example of exploitation-exploration search from a location A to a location B, where switching from one mode to another can be predefined or adaptive. In case of predefined, the setting of parameters is applied before starting the search. However, in case of adaptive, the parameters changes according to search situation. In this context, many metaheuristic approaches have been introduced such as Particle Swarm Optimization (PSO) (J. Kennedy & Eberhart, 1995), Ant Colony Optimization (ACO) (Dorigo et al., 2006), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), Cuckoo Search (CS) (Yang & Deb, 2009). Yang et al. (2014) emphasized this by saying, "The key components in metaheuristic algorithms for global optimization are local intensive exploitation and global diverse exploration, and their interaction can significantly affect the efficiency of a metaheuristic algorithm".

Many researchers believe that metaheuristic algorithms are effective, because their search mechanisms inspired from natural phenomena have balance mechanisms for exploitation and exploration (Fister Jr et al., 2013). However, metaheuristic algorithms are known to be hard to analyze. Due to their stochastic nature, it is difficult to build practically a significant theoretical frame on performance; hence they are often analyzed experimentally instead of theoretically (Črepinšek et al., 2013). The most common terms used in the experimental explanations are: exploitation and exploration. These concepts are widely used in the literature to analyze the performance of metaheuristic search. However, there is often misunderstanding upon definitions for exploitation and exploration (Fagan & van Vuuren, 2013). The first attempt presented a theoretical basis for the optimal balance of exploitation and exploration for 2D multimodal objective functions was presented by Yang et al. (2014). Their work focused on the key component of exploitation-exploration and introduced a simple practical estimation for the ratio of search times of exploitation and exploration. Fagan and van Vuuren (2013) highlighted various questions about the definitions that may be given to exploitation and exploration in each context of use. In contrast, the way how metaheuristic approaches are formulating and tuning exploitation and exploration is still not tackled yet.

Several terms of exploitation-exploration were found in the literature on metaheuristics (Auger et al., 2005; Li, Wang, & Liu, 2008; Sasena, Papalambros, & Goovaerts, 2002). These terminologies reflects the characteristics of specific approaches, however it cannot allow differentiating between exploitation-exploration forms. Xu and Zhang (2014) highlighted that there is an unambiguous approach which consists of considering each search within the whole search as an exploration, and any search within a part of the space is considered as exploitation. This approach, in machine learning refers to acquisition and utilization of knowledge (Yen, Yang, Hickey, & Goldstein, 2001). In some literature, exploitation-exploration is represented by intensity of randomness (Pchelkin, 2003). In this context, Chen, Xin, Peng, Dou, and Zhang (2009) discussed exploitation-exploration as two types of processes in acquiring information about undefined problems. The concept of Chen et al. (2009) considers a sampling process as exploitation iff (if only if) the sampling point is independent of the information obtained by previous sampling points. Hence, a sampling process is considered as exploitation iff they are generating a sample of points in dependence with the information obtained previously (Chen et al., 2009). In Lozano and García-Martínez (2010), exploration and exploitation were represented by diversification and intensification respectively, where diversification refers to the ability to cover many different places of the search space and intensification consists of reaching the best quality of the visited points. An optimal search requires having a balanced search between exploitation and exploration. In the following section, the most related work will be discussed.

2.5. Related Work in Exploitation-Exploration balance

The balance between exploration and exploitation has a significant impact on metaheuristic algorithms. The significance of this topic can be found in several domains of science and technology such as data science applications, machine learning , forecasting models (Chiroma et al., 2016), outlier models (Talbi, 2002), scheduling applications (Burnwal & Deb, 2013), ranking models (Wang & Yang, 2009) and other applications with NP-hard problems. Any optimization approach requires a balanced exploitation-exploration, where an optimal balance helps to speed up the computation time, increase the diversity of search, achieve a mature convergence and improve the solution quality. Finding the proper exploitation-exploration balance for a metaheuristic becomes a very complicated task. This section provides a detailed complete overview about balancing exploitation-exploration approaches used in metaheuristic algorithms. More importantly, a global view about metaheuristic algorithms could be achieved by understanding how and what balancing approaches are used. For example, a better understanding could explain why a particular search technique within a specific configuration is better than another.

Moreover, the existing random walk techniques used to model the animal movement are introduced such as Brownian walk, Levy walk in Yang and Deb (2009) and the composite random walk. Furthermore, a comparative study based on the balancing exploitation-exploration between these techniques is presented. An early discussion on search techniques among social insect behavior was carried out by Bonabeau et al. (1999). They were amazed by the characteristics that drive search in social insects. These search characteristics are used later to introduce various swarm intelligence algorithms such as Ant algorithms (Dorigo et al., 2006), Bee algorithms (Pham et al., 2005), Flower Pollination Algorithm (FPA) (Yang, 2012), Bat Algorithm (BA) (Yang, 2010c) and other algorithms which may be quite popular such as Artificial Immune System (AIS) (Farmer, Packard, & Perelson, 1986).

It is remarkable verity that each algorithm introduces different search techniques featured by different parameters, while the role of these parameters is to tune up the search to achieve an optimal solution efficiently. On the other hand, randomness of the search is usually seen to be principal. In this context many probability distribution have been applied (Yang, 2010b). The difference between metaheuristic algorithms can be intricate to understand, not only by novice people within the domain but also by many metaheuristics experts. The same thing is observed for exploitation-exploration approaches. The differences can be seen not only among metaheuristic instances but also between the same types of metaheuristic (Jans & Degraeve, 2007). Basically, it is all about reaching an adequate amount between exploitation and exploration. However, this is itself considered as a complex problem (Yang et al., 2014), and is related to the fact that there is no a particular rule to find the adequate ratio of exploitation and exploration (Fagan & van Vuuren, 2013). Instead of diversifying the search to cover the whole search space, the search can be intensive with small step sizes and that may cause immature convergence. This can be solved by increasing the rate of exploration to surpass local optima values (Črepinšek et al., 2013). An optimal balance can be achieved when the search technique is featured with a mechanism to tune up exploration and exploitation. The tuning can be found explicit such as (J. Kennedy & Eberhart, 1995) or implicit such as (Mirjalili et al., 2014). Smit and Eiben (2009) demonstrated that the Rastrigin function shows good performance in evolutionary algorithms using different tuning: population size = 14 and 448, tournament proportion = 0.8782 and 0.3503, and generational gap = 0.8443 and 0.0215. It is clear that the size of the replaced population is much smaller, which can lead to a significant difference in terms of selection. From this example, the aforementioned differences are quite easier to observe in terms of exploitation-exploration of the search.

Blum and Roli (2003) introduced a new approach of classifications of metaheuristics based on several features such as: nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic vs. static objective function, one vs. various neighborhood structures, and memory usage vs. non-memory usage, where a general view on diversification-intensification was provided. According to Blum and Roli (2003), exploration and exploitation is more related to short-term mechanisms tied based on randomness. However, diversification and intensification are related to medium and long term mechanisms based on memory usage. Another classification approach was proposed by Talbi (2002) where diversification and intensification were not included as the main characteristic for taxonomy. Črepinšek et al. (2013) focused only on the sub category of metaheuristics where they classified evolutionary algorithms into uni and multi process approaches based on the way exploitation-exploration balance is reached. Eiben and Schippers (1998) noticed that in evolutionary approaches the search space is explored by crossover and mutation, while exploitation is performed by selection. Moreover, Eiben and Schippers (1998) concluded that there was no a particular common approach about exploitation-exploration in evolutionary approaches. They highlighted that more research efforts are needed for better understanding the search in metaheuristics approaches. An alternative approach consists of forcing metaheuristics to employ themselves the role for both exploitation and exploration, and that is based on hybridization of the search with search algorithms featured by exploitation and exploration.

In order to achieve an optimal solution, GA must maintain a balance between exploitation and exploration and the related parameters to the search. This balance is determined by two main parameters: crossover and mutation. However, the balance process requires to be predefined which takes time to find the suitable values for the tuning up the balance. Črepinšek et al. (2013) showed that selection operator can be used to maintain exploitation-exploration by changing the selection amount. Maintaining an optimal tuning balance between exploitation and exploration is required for several optimization problems (David E Goldberg, 2013). The difficulty to balance exploitation and exploration in GA is based on the fact that it is very difficult to predict if a newly obtained solution by crossover or mutation operator will fit the exploitation-exploration search space. Moreover, these are not the only parameters to control to balance the search. Population size has an important impact too, where a search space with a large population size is explored better than a search space with a small size (Smit & Eiben, 2009). What is observed from this discussion is that the exploitation-exploration balance is able to be achieved however it requires lot of trial and error to find the optimal parameters to the problem landscape.

Table 2.1 describes the list of the most related work in exploitation-exploration approaches and the corresponding techniques used for tuning balance. It is clear that most of the approaches are based on predefinition of exploitation-exploration parameters. In GA, the exploitation is applied by using the evolution process where the mutation operator drives the search to the regions with best solution. In contrast, exploration is ensured by the crossover operator.

Table 2.1: Most Related work in exploitation-exploration approaches

Reference	Search technique	Exploitation	Exploration	Tuning of balance	Details	Convergence
(Yazdani and Joai, 2016)	Random search	Hunting operator	Migration operator	predefined	Exploitation-exploration is controlled by several prides	It is in relation with the division of prides
(Mirjalili., 2015)	A spiral movement	Moving inside the space between the moth and flame	Moving outside the space between the moth and flame	predefined	Balance depends on the max number of flames	Depends on the adaptive number of flames
(Deb et al., 2015)	Swarm behavior	Local clan	Global clan	predefined	Balance depends on clan and separating parameters	Depends on clan updating parameters
(Fong et al., 2015)	Random walk	Search agents	Escape probability model	predefined	Escape probability controls the extent of exploration	Depends on the rate of the escape
(James et al., 2015)	Random walk	Dimension mask in the random walk step	Rate of vibration	predefined	Degree of exploration is controlled by the algorithm parameters	Depends on the parameters setting
(Findik, 2015)	Evolution process	Mutation	Crossover process	predefined	The distance between crossing points defines the balance	Depends on the crossing operator
(Mirjalili, 2015)	Swarm behavior	Cohesion parameter	Alignment parameter	predefined	alignment and cohesion are tuned to control the balance	Depends on dose of alignment and cohesion
(Yang, 2014)	Ley flight	Local pollination	Global pollination	predefined	The tuning is based on switch probability *p*	Depends on suitable switch probability rate
(Mirjalili, 2014)	Random search via agents	Attack prey model	Search preys model	predefined	Tuning is guaranteed by a specific adaptive values	Depends on the initialized parameters
(Cuevas et al., 2014)	Cycle of the state of the matter	Solid state	Gas state	predefined	The tuning is related to the original parameters of the gas state.	Fast convergence associated with an optimal setting of matter
(Yang and Gandomi, 2011)	Echolocation based search	Loudness	Pulse emission	predefined	The balance is affected by the sensitivity of the search	Convergence affected by sensitivity of pulse emission
(Yang and Deb, 2009)	Levy flight	Step size	Step size	predefined	Random search based on Levy flights	Depends on the setting
(Kennedy & Eberhart, 1995)	Swarm behavior	Particles pbest	Particles gbest	predefined	The particles are oriented via velocity acceleration	Depends the settings
(Dorigo et al., 2006)	Swarm behavior	Pheromone	Heuristic information	predefined	Tuning is controlled by sharing information between ants.	The convergence is very sensitive to antennation.
(Holland, 1975)	Evolution process	Mutation	Crossover process	predefined	The distance between crossing points defines the balance	Depends the settings

Dorigo et al. (2006) introduced the ACO. It is based on ant swarm, where the exploitation is realized by pheromone operator and exploration is realized by heuristic information operator. ACO is featured by several search parameters where the combination of all these parameters should be experimented first in order to find the suitable amount of pheromone to balance exploitation and exploration (Mullen, Monekosso, Barman, & Remagnino, 2009). J. Kennedy and Eberhart (1995) presented the PSO, which is a swarm intelligence based. This approach becomes very popular in several domains of technology. However, it has a huge number of parameters which increased the complexity of tuning parameters.

Exploitation-exploration for PSO is presented by particles local and global. Moreover, achieving the balance between the local and the global particles is challenging in terms of finding the suitable velocity setting. Yang and Deb (2009) presented the CS which is inspired from the brood parasite of cuckoo bird. The search for solution is realized using Levy flight which can be set to have different step size and thus exploit and explore the search space. However, a balanced search requires a tuning of population size, step size and mortality rate in order to fit the performance of CS with the optimization problem. It is shown that Levy flight performs better in exploration, however it is not the optimal search strategy for exploitation (Benhamou, 2007; Gautestad & Mysterud, 2013). Yang et al. (2012) proposed another approach based on bat behaviour. The concept of search is based on the echolocation that is used by bat to find preys. Exploitation is represented by loudness of sonar, while exploration is represented by pulse which is used to set the size of the search. Tuning the balance between exploitation and exploration requires setting several parameters and that needs a lot of time to experiment the suitable setting for each optimization problem. Moreover, a parameter setting might work well for a particular optimization problem but may not work well on another optimization problem.

Cuevas et al. (2017) developed an approach based on states of the matter, where the exploitation-exploration balance is represented by the state of the matter that can be solid, gas or liquid. Several parameters are needed to be tuned up to balance the exploitation-exploration including the number of molecule population, gas operator, liquid operator and solid operator. Mirjalili et al. (2014) introduced a new metaheuristic approach based on wolf behaviour. Exploitation-exploration is modelled by two states: attacking a prey and searching for a prey. Tuning parameters is guaranteed by specific adaptive values, however an initialization is needed. Another approach was presented by Yang (2012) based on Levy flight, where exploitation-exploration is interpreted by local and global search. The factor p is used to switch and balance the search. Finding the right value of this factor is challenging especially for the problems with high dimensionality.

Mirjalili (2016) proposed a new approach inspired from the behaviour of dragon fly insect. Exploitation is modelled using cohesion operator which refers to the tendency of individuals towards the centre of their neighbourhood and exploration is modelled using alignment operator which refers to velocity of matching between individuals. Considering these two operators, the search can achieve an optimal solution. However, setting the initial values of these operators is primordial to scale the search with the problem landscape. Another recent evolution approach was proposed by Findik (2015), where a sort of genetic algorithm incorporated with the theory of swarm where the global optimum is modelled by the leader of bulls herd.

The exploitation and exploration realized by mutation and crossover operators. Setting the search to solve an optimization problem requires tuning up the initial parameters including crossover and mutation operators. James and Li (2015) presented a new metaheuristic approach inspired from the foraging strategy of spiders, by using vibrations on the spider web to locate the position's prey. The authors highlighted that finding the suitable parameters to balance exploitation-exploration might be time-consuming, hence proposed to predefine two parameters: rate of vibration to control exploration and dimension mask value to control the exploitation of the search space.

Fong, Deb, and Yang (2015) proposed a metaheuristic approach inspired from the wolf behaviour. Exploitation of the search space is conducted by the wolves' search agents, which is modelled by a random search to browse better solutions near their current locations. Exploration of the search space is modelled by the parameter of escape probability, which is used to control the diversification of the search. Increasing the value of escape probability means more number of wolves will move to the other dimensions away and that means exploring more regions. Finding the optimal balance of the search needs to set the suitable amount of escape probability and search agents operators. However, that requires trial-and-error process which is a complex and time consuming task. Another metaheuristic approach was proposed by Deb at al. (2015) consisting of hybridization of evolutionary technique and balancing exploitation-exploration. The search is inspired from elephants and their clans behaviour. It is shown that exploitation can be controlled using local clans and global clans. Exploitation operator is represented by female elephants, while exploration operator is represented by male elephants.

Mirjalili (2015) presented a metaheuristic approach inspired from the moth insect and its behaviour. The exploitation of the search space consists of moving inside the space between the moth and the flame. In contrast, exploration consists of moving outside the space between the moth and the flame. Balancing exploitation and exploration is controlled by the number of flames operator. In such approaches, all operators' combinations should be experimented to find the optimal tuning. Yazdani and Jolai (2016) proposed the Lion Optimization Algorithm (LOA) which is inspired from the behaviour of lions and their cooperation mechanism. The exploitation is realized using hunting operator, where a mechanism is modelled to find the solution. Furthermore, the exploration of the search space is controlled using the migration operator to diversify the target preys. Tuning parameters to achieve an optimal balance is a question that every metaheuristic algorithm has to fulfil. Several metaheuristics nature-inspired approach have been proposed last recent years such as: Marriage in honey Bee Optimization algorithm (MBO) (Abbass, 2001), Bacterial Foraging Algorithm (BFA) (Du & Swamy, 2016a), Shuffled Frog Leaping algorithm (SFL) (Eusuff, Lansey, & Pasha, 2006), Cat Swarm Algorithm (CSA) (Chu, Tsai, & Pan, 2006), Firefly algorithm (FA) (Yang, 2010a), Dolphin Partner Optimization (Shiqin, Jianjun, & Guangxing, 2009), Flower pollination algorithm (Yang, 2012), Krill herd (Gandomi & Alavi, 2012), Forest Optimization Algorithm (Ghaemi & Feizi-Derakhshi, 2014). It is clear that in most of the existing metaheuristics, the following approaches have been tried.

i. Trial-and-error methods, which is based on experiment of all the possible combinations to find the optimal output (Birattari & Kacprzyk, 2009; Črepinšek et al., 2013).
ii. Using same setting, but it is only applicable for similar cases.
iii. Following recommended parameters, which are not optimal for all cases (Smit & Eiben, 2009).

From the exploration-exploitation point of view, this means that different problems require different amounts of exploration-exploitation. This can be interpreted by an optimal balance between exploitation and exploration. For example, for uni-objective problems, less exploration is required than in case of multi-objectives problems. Therefore, the goal of any search algorithm is to inherently find a good balance between exploration and exploitation for different problems. It is shown that the amount of exploitation-exploration is changing during the search for solutions. For example, in early stages, a movement with large step size is needed to explore the search space. However, in the later stages a search with small step size is enough to exploit the search area, which is interpreted as diversification and intensification by Yang et al. (2014). It is shown that mathematical modeling of animal movement provides efficient search approaches for metaheuristic algorithms. In the following section, modeling search approaches based on animal movement are discussed.

2.6. Modelling Search Approaches based on Animal Movement

The movement of animals is characterized by three modeling theories: classic patch-use models, random search models and composite search models (Bartumeus & Catalan, 2009; Bartumeus et al., 2005; Nolting, 2013). It is explained by Charnov (1976) how foragers browse the search space to exploit discrete and defined patches. In the classic patch-use models, the focus is how foragers decide when to leave particular patches. However, other details such as how foragers determine patches are ignored. By incorporating stochastic processes, the random search models have been introduced, where targets are denoted by points, and the movement patterns of the foragers required to achieve theses points is described. After then, the hybridization of random models is applied to produce the composite search models, where the movement is divided into intensive and extensive modes. The general hierarchy of these models is described in Figure 2.9.

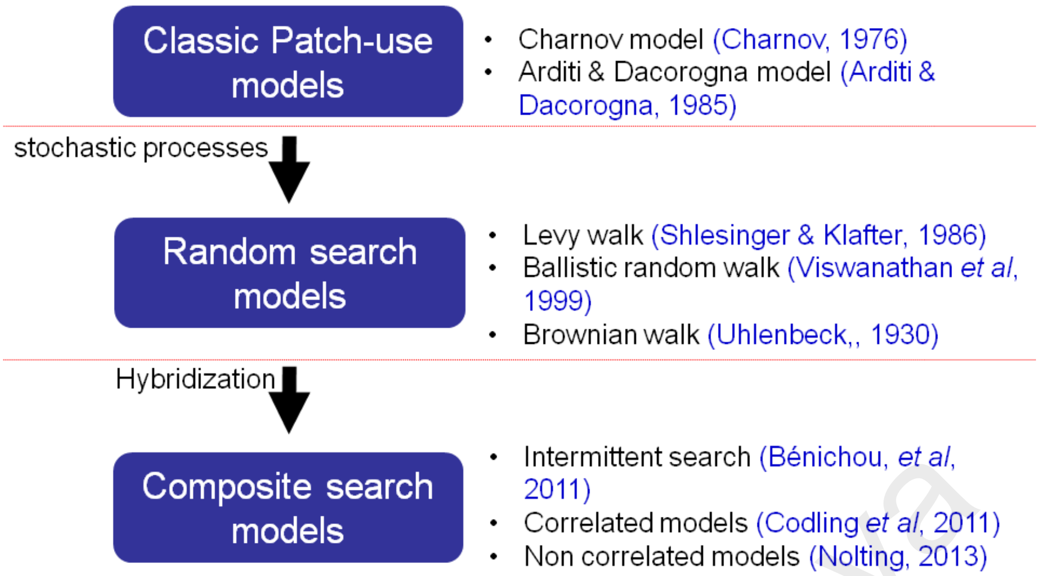


Figure 2.9: Search Approaches Based on Animal movement

2.6.1. Classic Patch-use models

Indentifying the key characteristics of the classic patch-use models requires to understand the Charnov's marginal theorem (Charnov, 1976). The Charnov model is one of the fundamental principles of optimal searching. It consists of infinite number of resource patches that are divided into a particular number of types. The required travel time to move between patches is fixed. Moreover, the number of the patches that will never be revisited and the probability to visit a particular patch are tuned up. Oaten (1977) found that this model represents the movement as a deterministic process, so the behavior of the search is determined before it even begins the search. However, in real world the movement of animals is not a deterministic continuous process, it is a discrete stochastic process featured by finding different targets. The lack of locating of resources intra and inter patches is a limitation that has been observed in patch-use models.

Arditi and Dacorogna (1985) solved these issues by proposing a spatially explicit model where targets can be arbitrary located spatially. However, the way how it locates the search path is unrealistic and not compatible with the NP-hard nature. Although this model is able to determine the true optimal path for a movement, its predictions are too accurate to be a useful null model. When a specific path is predicted through a search space, it is unrealistic to estimate that the experimental observations will exactly match that path, and it is difficult to assume how a particular movement is close to the optimal path (Nolting, 2013).

2.6.2. Random Search Models

Random search models are introduced as an alternative to the patch-use models. They are characterized by a stochastic process, where the targets are represented as points and the search movement is defined within a particular radius. The main element of random search models is based on randomization of variables. In the upcoming subsections, a review of the fundamentals of random walks is presented. These concepts may help to understand how metaheuristics are carried out.

2.6.2.1. Random Variables

A random variable can be defined as a function, where the events are translated to real number. For each random variable, a probability distribution is associated with the function to express its desnity (Yang, 2010b). For example, the number of clicks per day for web page, and the number of requests of e-ticket distributer per hour can be interpreted as Poisson distribution

$$p(n;\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \quad (n = 0, 1, 2, ...), \tag{2.11}$$

here $\lambda > 0$ refers to the expectation of occurrence of the event during a range of time.

The nature of random variable is defined according to its distribution. Gaussian distribution and normal distribution considered as the most widely used distributions (Yang, 2010b). The reason behind their popularity is related to the fact that many natural phenomena have the same distributions

$$p\left(x\,;\mu\,,\sigma^2\right)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\,\sigma^2}\right],\qquad -\infty < x < \infty\,, \tag{2.12}$$

Where μ denotes the mean, σ represents the standard deviation. The normal distribution is represented by $N\left(\mu\,,\sigma^2\right)$. When $\mu = 1$ and $\mu = 0$, the distribution is denoted by $N\,(0\,,1)$ and it is called normal distribution.

In the context metaheuristics, Levy distribution is very well known because of its applicability in modeling animal movement (Benhamou, 2007; Gautestad & Mysterud, 2013; Sims, Humphries, Bradford, & Bruce, 2012), where the distribution is represented by the sum of N identically and independently distribution random variables with Fourier transform denoted as follows

$$F_N(k)=\exp\left[-\,N\,|\,k\,|^{\beta}\right]. \tag{2.13}$$

$$L\left(s\right)=\frac{1}{\pi}\int_0^{\infty}\cos\left(\tau\;s\right)e^{-\alpha\;\tau^{\;\beta}}\;d\tau\;,\qquad (0<\beta<2)\,, \tag{2.14}$$

where $L\left(s\right)$ represents the Levy distribution an β denotes the index. It is shown that for most cases $\alpha = 1$. The integral equation can take two different schemes, in case of $\beta = 1$, the integral representes the Cauchy distribution. In case $\beta = 2$, the integral represents the normal distribution, where Levy walk in this case becomes similar to the Brownian motion (Nolting, 2013; Yang, 2010b).

The mathematical representation of the integral can be expressed as an asymptotic series, where the leading-order approximation of the whole walk length provides a power-law distribution described as follows (Yang, 2010b)

$$L(s) \sim |s|^{-1-\beta}, \quad (2.15)$$

it is very important to highlight that the distribution is described as a heavy-tailed and the variance of the distribution is infinite for $(0 < \beta < 2)$. In the following section random walks based power-law are discussed (Benhamou, 2007; Nolting, 2013; Yang, 2010b).

2.6.2.2. Random Walks

In mathematics, the definition of a random walk is introduced as a formalization of a path that is constructed using a series of successive random steps (Nolting, 2013; Nolting, Hinkelman, Brassil, & Tenhumberg, 2015). The search process is not requiring any prior knowledge of the location of targets. The search movement for finding the optimal target is called search strategy. Let $Step_N$ represents the sum of the consecutive random steps denoted by S_i, where the $Step_N$ create a form of a random walk and it can be described as follows.

$$Step_N = \sum_{i=1}^{N} S_i = S_1 + \ldots + S_N \quad (2.16)$$

where S_i represents a random step generated from a random distribution. The whole random walk can be represented as a recursive model as follows.

$$Step_N = \sum_{i=1}^{N-1} + S_N = Step_{N-1} + S_N \quad (2.17)$$

here the next state of the random walk $Step_N$ is in dependence with the current state $Step_{N-1}$. Particularly, this feature is the main characteristic of Markov Chain approach (Gilks, Richardson, & Spiegelhalter, 1995; Vignat & Plastino, 2006). The step size of S_i can be tuned up or fixed according to the search situation. For example, in case of a rabbit walking on a road, at each step, the movement can only be backward or forward because of the nature of the road and this may create at the end a random walk. In case this rabbit walks on a garden, then the walk can be in any direction, which can be interpreted as a 2D random walk. The mathematical representation of this scenario movement can be described by the following equation

$$Step_{t+1} = Step_t + \omega_t \tag{2.18}$$

where $Step_t$ is the current state at time t, and ω_t is a step variable represents the step size and it is implemented by using a known distribution. In fact, the step size is main factor used to define the nature of the random walk and it very according to the type of distribution. The use of random walk as models of animal movement search has a long record in the literature (Bartumeus et al., 2005; Nolting, 2013; Viswanathan, 2011). There are three models widely used to model the animal movement: Levy walk, Ballistic walk and Brownian walk. An optimal random walk should include the full set of hypothesis that describes an animal movement behavior and how it responds to any environment changes. However, such a random walk is unattainable because of the high complexity of the animal movement (Nolting, 2013).

2.6.2.3. Levy Walk

Levy walk is considered as one of the random walk techniques that are potentially used to model animal movement. It is characterized by a specific step size (random number), where the length between two consecutive portions of direction is calculated from a probability distribution with an inverse power-law tail described as below (Buldyrev et al., 1999; Freeman et al., 2007).

$$L(s) \sim |s|^{-1-\beta}, \tag{2.19}$$

where $(0 < \beta < 2)$. In fact, the generation of random numbers with Levy walk particularly consists of two main steps: the selection of a random direction and the calculation of step size which is in conformity with Levy distribution.

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\dfrac{\gamma}{2\pi}} \exp\left[-\dfrac{\gamma}{2(s-\mu)}\right] \dfrac{1}{(s-\mu)^{\frac{3}{2}}}, & 0 < \mu < s < \infty \\ 0 & otherwise, \end{cases} \tag{2.20}$$

here the parameter $\mu > 0$ is a minimum step size and γ is a parameter denotes scaling. As $s \to \infty$, the generated distribution can be described as

$$L(s, \gamma, \mu) \approx \sqrt{\frac{\gamma}{2\pi}} \frac{1}{s^{3/2}}, \tag{2.21}$$

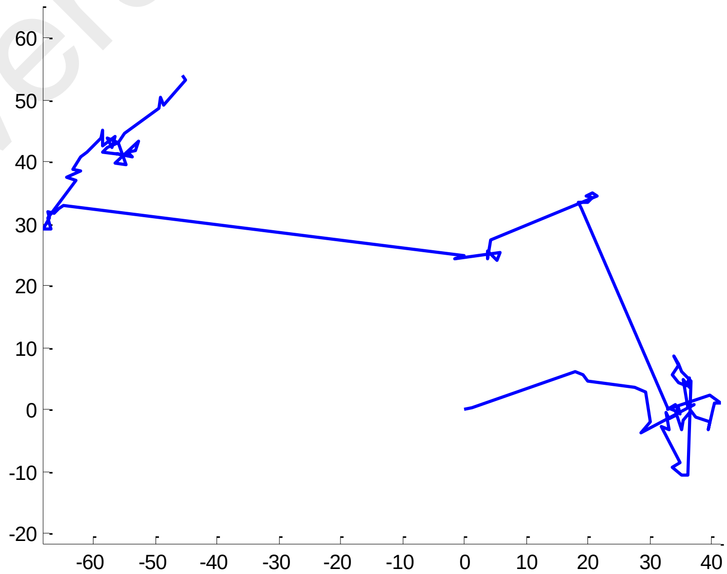


Figure 2.10: Levy walk in 2D, number of steps = 100

From Figure 2.10, an example of Levy walk with 100 steps is illustrated. It is characterized by an anomalous diffusion, where the mean squared displacement increases faster linearly with time. It is shown that Levy distribution is defined in terms of transform (Nolting, 2013; Yang, 2010b).

$$F(k) = \exp[-\alpha \mid k \mid^{\beta}]. \qquad 0 < \beta < 2, \tag{2.22}$$

where the parameter α denotes scaling. For the $\beta = 3$, the distribution can be denoted as follows

$$F(k) = \exp[-\alpha \ k^{2}], \tag{2.23}$$

whose inverse Fourier transform corresponds to a Gaussian distribution. Another special case is $\beta = 1$, and that leads to the following equation

$$F(k) = \exp[-\alpha \ k \ ], \tag{2.24}$$

which corresponds to a Cauchy distribution

$$p(x, \gamma, \mu) = \frac{1}{\pi} \frac{\gamma}{\gamma^{2} + (x-\mu)^{2}}, \tag{2.25}$$

where μ represents the location, while γ is used to scale the distribution. The inverse integral can be described as follows

$$L(s) = \frac{1}{\pi} \int_{0}^{\infty} \cos(k \ s) \, e^{-\alpha |k|^{\beta}} \, dk, \tag{2.26}$$

And it can be generated only in case of s is large.

$$L(s) \to \frac{\alpha\beta\,\Gamma(\beta)\sin\left(\pi\beta/2\right)}{\pi|s|^{1+\beta}}, \qquad s \to \infty \tag{2.27}$$

where $\Gamma(z)$ is the Gamma function

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt. \tag{2.28}$$

when n is an integer, $\Gamma(n) = (n-1)!$. Levy walk is very efficient in modeling animal movement. Benhamou (2007) demonstrated that animals perform Levy walk patterns during search for resources that are distributed in different patches. For this, animals use two modes, intensive mode to concentrate on the search inside the patch (exploitation), and extensive mode to move from one patch to another (exploration). It was shown by Gautestad (2012), Plank and James (2008) that animals route is quite similar to Levy walk.

2.6.2.4. Ballistic Walk

Ballistic walk is the simplest of all random walks models. A movement search using this model moves in a straight line with a randomly selected direction until it finds a target. After exploiting the found target, a searcher randomly chooses a new direction, and again moves in a straight line in order to find another target.

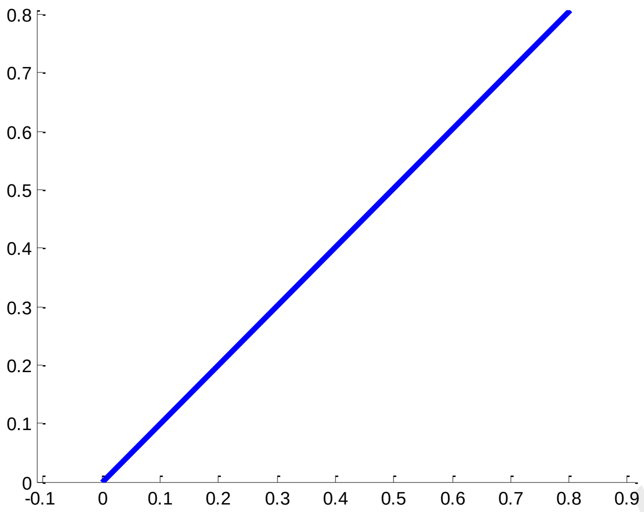


Figure 2.11: Ballistic walk

Figure 2.11 shows that when the step lengths of a random walk are drawn from a probability distribution with finite variance, the random walk converges to diffusive motion at sufficiently long time scales. If a particle moves according to one of these stochastic processes, its displacement from its initial position scales in proportion to $t^{\frac{1}{2}}$ where t denotes the flight length. If a particle's displacement from its initial position scales slower than $t^{\frac{1}{2}}$, its motion is sub-diffusive. If a particle's displacement from its initial condition scale faster than $t^{\frac{1}{2}}$, its motion is super-diffusive. Ballistic motion is an example of super-diffusive motion.

2.6.2.5. Brownian Walk

Brownian motion is a stochastic process that, on a heuristic level, can be thought of as the limit of a simple random walk, as the step sizes approach zero. The resulting trajectories are continuous, but nowhere differentiable. Brownian walk is characterized with a normal diffusion where the mean squared displacement increases linearly.

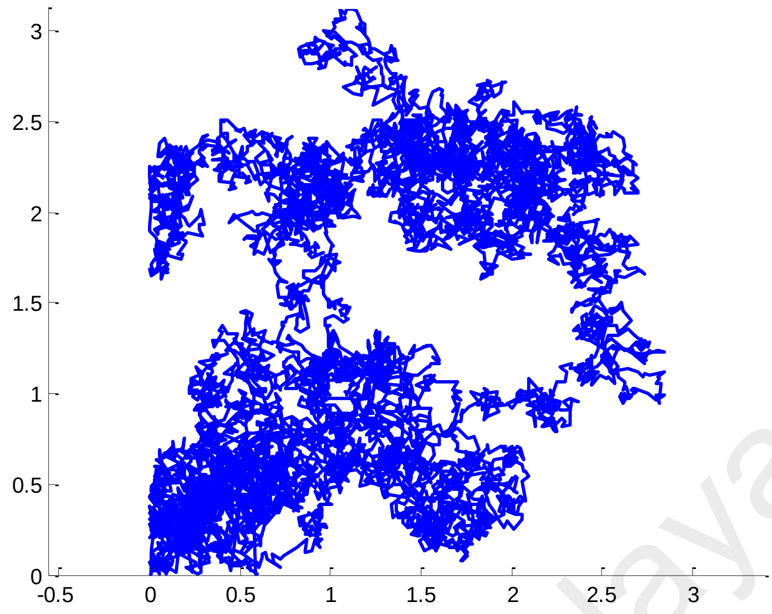


Figure 2.12: Brownian walk in 2D, number of steps = 1000

Figure 2.12, shows an example of Brownian walk with 1000 steps. It is among the most commonly invoked stochastic models of animal movement. It is shown that Brownian are defined as power-law random walks where the probability distribution is denoted as $\rho \sim u = t^{-\lambda}$. λ is used to describe the nature of the probability distribution of Brownian and its value is defined between 1 and 3. For λ > 3, the variance of the probability distribution is finite, and the resulting random walk converges to Brownian motion at sufficiently long times. Hence power-law walks with λ >= 3 are referred to as Brownian walks.

2.6.3. Composite Search Models

In composite search models the search efforts are divided into two modes: intensive and extensive. The intensive mode is employed in the spaces where there are abundant targets. In contrast, the extensive mode is employed in the spaces where targets are rare and poor. It is shown that the targets are close to each other in abundant spaces and far from each other in the target poor spaces (Benhamou, 2007). In intensive search, the animal searches the space using short step size movement with different directions. In extensive search, the animal moves in the search space using long step size between the sparse targets with few interruptions. This combination of search modes is called the composite search (Plank & James, 2008).

The animal movement literature generally refers to composite search as patch-restricted search (Weimerskirch, Pinaud, Pawlowski, & Bost, 2007) or patch-concentrated search (Benhamou, 1992). There are several examples of animal movement that utilize composite search such as honeybees (Tyson, Wilson, & Lane, 2011), birds (Nolet & Mooij, 2002), fish (Thomas T Hills & Adler, 2002) and ringed seal (Kelly et al., 2010). Some models proved that when the targets are abundant, Brownian walk is performed by animals, whereas targets are sparse at the search space Levy walk is performed (Sims et al., 2012). However, there is no researches focused on switching between exploitation and exploration. In Nurzaman et al. (2011) a model of Levy and Brownian is presented, showing how Escherichia-Coli switches from Levy to Brownian mode based on target densities.

Considering the features of exploitation-exploration such as balance, switch mode, spatial explicit representation and stochastic movement, ringed seal movement is selected to build a composite search model. A comparison is presented in Table 2.2 between the search models in order to find the suitable search strategy that can provide a balanced exploitation exploration. Moreover, this comparative study is used to validate and show the significance of choosing specific search models to model the sensitive search model.

Table 2.2: A comparison between search models

Model	Exploration	Exploitation	Balance	Switch mode	Spatial explicit representation	Stochastic movement
Classic patch-use models	✓	✓	-	-	-	-
Random Search Models	✓	✓	-	-	✓	✓
Composite search models	✓	✓	✓	✓	✓	✓

From Table 2.2, it is clear that the only way to have a search model with a balanced exploitation and exploration is to make the choice based on composite search models, where the necessity to establish a balanced exploitation-exploration requires to study the nature of the incorporated random walks in the composite search model. In Table 2.3, a comparison between power-law random walks is presented.

Table 2.3: A comparison between random walks

Model	Exploration	Exploitation	Motion	Details
Brownian walk	-	Support	Sub-diffusive	It is shown that this random walk performs better during intensive search
Levy walk	Support	-	Super-diffusive	Efficient for extensive search and it doesn't require to adjust the scale of its behavior to the search space bounds
Ballistic walk	Support	-	Super-diffusive	It is only recommended for destructive search that is characterized with no change in direction

From Table 2.3, it is clear that the combination of Brownian-Levy shows an efficient way to form a composite model suitable for exploitation and exploration balance.

2.7. Ringed Seal Movement

Optimization is a substantial challenge for organisms, where escaping predators, searching for habitats and foraging defines their behaviour. The mechanism used by organisms to search optimally to get best habitats is developed through hundreds of years in nature. In this study, the focus is on ringed seal movement which is a semi aquatic animal, not only because of its extraordinary ability to stay and dive underwater for a long time, but also because of its behaviour used to resist natural fluctuations. This behaviour is developed since thousands of years, making the seal to be adaptable to unexpected and difficult conditions. As all semi aquatic animals, underwater activities of diving for seals are constrained by the need for surface gas exchange. The seal breeding also requires a suitable environment to guarantee the reproduction of new generations (Le Boeuf et al., 2000).

During autumn and winter in the Canadian arctic, the ice starts freezing over, so the seals create breathing holes and snow covered lairs. Between March and May, ringed seals give birth to pups in snow-covered lairs connected to the ocean. These lairs provide a thermal protection against cold air temperatures and high wind chill, and afford at least some protection from predators such as bears (Hammill & Smith, 1991; Pilfold, Derocher, Stirling, Richardson, & Andriashek, 2012; Williams, Nations, Smith, Moulton, & Perham, 2006). A seal could have a complex of lairs at one specific area (Gjertz & Lydersen, 1986; Kovacs, Lydersen, & Gjertz, 1996; Pilfold, Derocher, Stirling, & Richardson, 2014), which can be used for many functions: breeding and birthing of young pups and resting. Lairs are maintained until the end of the breeding season in spring, approximately six weeks after pupping, or until snow melt causes structural collapse (Pilfold et al., 2012). In nature, two different types of lairs were observed (Lydersen & Gjertz, 1986). Generally, the famous type in both coastal and offshore habitats is haul-out lairs, which is characterized by a single-chambered room and has a round design. Another different type of lairs found is called the birth lair. A birth lair can be characterized by the existence of placental remains, hair and also by extensive tunnels created by pups.

The seal pup strategy consists of searching the best lair to avoid predators. The young pup moves between lairs within her complex of lairs. If a lair is attacked, destroyed or its quality not good, pups are able to change the location between lairs structures (Lydersen & Gjertz, 1986; Pilfold et al., 2014). The search movement of the seal is sensitive to external noise emitted by predators such as polar bears. In case of noises, the pup leaves the proximity far away. However, in normal situation where there is no external noise; the pup keeps browsing the proximity (the multi-chambered lair) searching for best location. Basically, the quality of the habitat depends on the structure of the lairs, therefore during the breeding season male ringed seal emits a strong gasoline smell which may indicate the location of the lairs (Kunnasranta, Hyvärinen, Sipilä, & Medvedev, 2001; Lydersen & Gjertz, 1986). Wounds on both males and females represent another smell index that can mark territories. This makes seals very vulnerable and unsafe and could be targeted by bears. A polar bear can locate seal lairs using the smell index (Hammill & Smith, 1991). Its strategy consists of sniffing the ice surface with self-possession searching for a seal meal; if a smell is detected, the polar bear will run and jump on the snow over the hole to collapse the lair and block the exit. The bear can then catch the mother and the pup together. The ringed seal strategy used to search and choose the best lair can be associated with the objective problem to be used to balance between the exploitation and the exploration of the search.

2.8. Summary

In this chapter, the basic concepts of metaheuristics were discussed including the baseline algorithms used for global optimization such as GA, PSO and CS. A detailed complete overview about balancing exploitation-exploration approaches and the related work are discussed and pointed out that most of the existing metaheuristic approaches are based on predefined techniques to balance exploitation-exploration. A comparative study is conducted between the search models used to model the animal movement and pointed out that composite search models as most suitable among them. Random walks were compared based power-law and it is found that the most suitable random walks for building a composite search model are Levy walk and Brownian walk for the purpose of balancing exploitation and exploration. Finally, ringed seal movement, which is characterized by a composite search pattern is discussed. It is found that the sensitivity of the seal plays a role in switching between intensive search and extensive search, which is applicable in the context of balancing exploitation and exploration. In the following chapter, the proposed theoretical framework of this research is introduced.

CHAPTER 3: RESEARCH METHODOLOGY

3.1. Introduction

The idea to provide computers with artificial intelligence (AI) solutions started from early days of computer age. Although there was not a specific date mentioned in the literature, Alan Turing and John Von Neuman were founders in the domain of computer science stimulated by the hope of achieving the degree of intelligence where complex problems can be solved.

Computer sciences were also absorbed in other technology domains such as psychology and biology, where natural systems were the only way towards building of optimization solutions into computer programs. Furthermore, applications of computers were extended to mimicking biological evolution by using Darwinian concept, modeling animal movement and development of evolutionary computation such as GA and PSO.

The process of modeling animal movement to develop algorithms consists of several steps. In this reaserch, the focus is on modeling the movement of the ringed seal, which is characterized by a composite search pattern that can allow building a global optimization approach with a balanced exploitation-exploration. This chapter explains the research methodology used in the study. The research starts by conducting a literature review of the advances made in the exploitation-exploration tuning approaches, and unveils limitations in the literature. The limitations discovered in the previous works are the research problems of the present study. The problem statement was formulated based on the limitations. The research aims and objectives were derived from the problem definition.

3.2. Proposed Methodology

The proposed research methodology is divided into four main phases as shown in Figure 3.1. Every step of the proposed methodology is described as follows.

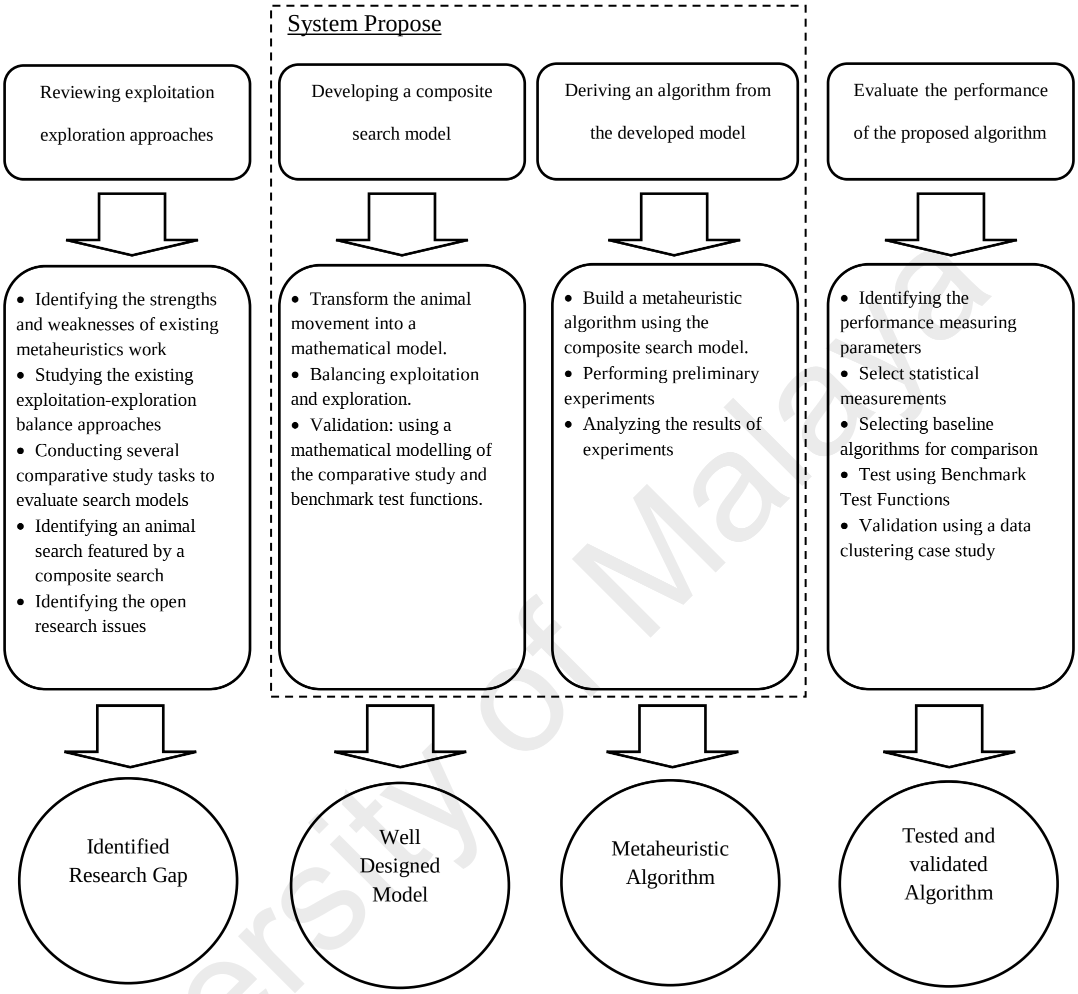


Figure 3.1: Proposed Research methodology

3.2.1. Reviewing Exploitation Exploration Approaches

The existing metaheuristics nature-inspired were reviewed as well as the challenges in balancing exploitation-exploration. The existing metaheuristics nature-inspired algorithms were defined to what extent they can overcome the challenges. Furthermore, the problems of existing exploitation-exploration methods for metaheuristics algorithms were studied. Moreover, a comparative study between search models to define the suitable search model to build an exploitation-exploration search. In order to find the suitable random walk for intensive and extensive search, a comparative study is presented. An overview about the animal search approaches is introduced to identify an animal search featured by a composite search pattern. The analysis of the existing approaches gives a wider perspective of the problems in exploitation-exploration methods in metaheuristics and thus helped to identify the research gap.

3.2.2. System Propose

In order to achieve the objectives, a model which will be called the RSS is proposed. The design of the sensitive search model is inspired from the ringed seal movement and implemented using a composite of Brownian walk and Levy walk. It is shown in the literature review that the Brownian walk performs well for exploitation and Levy walk performs well for exploration. The process of building the metaheuristic algorithm namely the Ringed Seal Search (RSS) were derived from the sensitive search model.

3.2.2.1. Developing the Composite Search Model

Generally, in nature lot of organisms perform random search during foraging and searching for resources such as food and water. Several recent studies showed that many animals perform random search based on statistical procedures (Bartumeus et al., 2014; Dees, 2009; Humphries et al., 2010; Ito, Uehara, Morita, Tainaka, & Yoshimura, 2013; Nurzaman et al., 2011; Sims et al., 2012; G. M. Viswanathan, Da Luz, Raposo, & Stanley, 2011). One of the random walk techniques that gained much interest is the Levy walk, which is characterized by a heavy tailed step length distribution. On the other hand, some new introduced search techniques (Bartumeus, Catalan, Fulco, Lyra, & Viswanathan, 2002; Buldyrev et al., 1999) show that Levy walk performs better for search with sparse targets. In contrast, Brownian walk is more efficient where the step lengths are not heavy tailed. The aim of this section is to describe the search behavior of the seal pup during normal and urgent state. The movement of seal pup is characterized by a high sensitivity to external noise as shown in the figure below

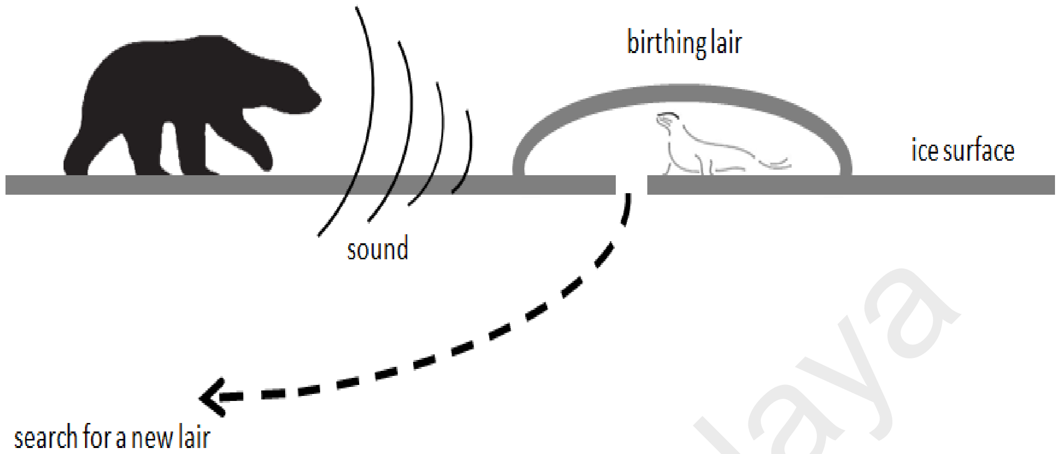


Figure 3.2: Seal's movement when leaving the lair (urgent state)

Figure 3.2 shows a seal pup inside a birthing lair, on the other side a bear in movement on the ice surface. In case of an urgent state, the seal pup strategy consists of two options, keep silent and wait for unknown destiny, or jump inside the sea through the hole to find another lair to escape the predator.

Recent researches showed that some noise-based strategies, namely biological fluctuation has an effect on the life sciences (Yanagida, Ueda, Murata, Esaki, & Ishii, 2007). This strategy also exists in many varieties of bacteria, where its role consists of providing an adaptation to environment changes. Stimulated by this natural phenomenon, several models have been introduced to explain the biological fluctuation (Kashiwagi, Urabe, Kaneko, & Yomo, 2006; Nurzaman et al., 2011; Yanagida et al., 2007). The movement of seal is also characterized by sensitive reaction to external noise. The search of the seal is therefore designed to have two different patterns, normal search (normal state) where there is no noise or urgent search (urgent state) in case of noise.

For the urgent search state, the seal pup leaves its own lair and performs a long step lengths using a Levy walk as shown in Figure 3.2. The purpose of this long step search pattern is to escape the external noise threat emitted by the predator and explore if other lairs could be safer. In terms of global optimization point of view, this could be interpreted as an exploration of the search space. For the normal search state, the seal exploits the local area searching for a better location as shown in Figure 3.3. In contrast to the urgent state, in normal state the seal is not threatened by an external noise and that is an enough reason to keep exploiting the proximity of the current lair.

Figure 3.3: Seal inside a multi-chambered lair during a normal state, designed by Robert Barnes, UNEP/GRID-Arendal (Robert, 2007)

Figure 3.3 shows a seal pup inside a multi-chambered lair. In the absence of external noise, the seal prefers to exploit the local area (the chambers of the lair). This represents a normal search state when the seal pup performs a Brownian walk with a non-heavy tailed step length that can be interpreted as an intensive search at the proximity (exploitation). In nature, one mother seal can have a complex structure of lairs at one place.

3.2.2.2. Levy Walk and Brownian Walk

In mathematics, the definition of a random walk is introduced as a formalization of a path that is constructed using a series of successive random steps. Levy walk is considered as one of the random walk techniques that are potentially used to model animal movement. It is characterized by a specific step size (random number), where the length between two consecutive portions of direction is calculated from a probability distribution with an inverse power-law tail described as below (Buldyrev et al., 1999; Freeman et al., 2007; Viswanathan et al., 2008).

$$\text{Levy} \sim u = t^{-\lambda} \tag{3.1}$$

where $1 < \lambda < 3$ and t is the flight length. In fact, the generation of random numbers with Levy walk particularly consists of two main steps: the selection of a random direction and the calculation of step size which is in conformity with Levy distribution. Selecting a random direction is calculated through a uniform distribution. In case where $\lambda \geq 3$, the distribution will not be in a heavy tail and the total sums of the lengths converge to a Gaussian distribution. Levy walk is characterized by an anomalous diffusion, where the mean squared displacement increases faster linearly with time. However, Brownian walk is characterized with a normal diffusion where the mean squared displacement increases linearly.

In Benhamou (2007), it is shown that animals perform Levy walk patterns during search for resources that are distributed in different patches. For this, animals use two modes, intensive mode to concentrate on the search inside the patch (exploitation), and extensive mode to move from one patch to another (exploration). It was shown that animals route is quite similar to Levy walk (Gautestad, 2012; Plank & James, 2008; Sims et al., 2012). However, some models demonstrated that when prey resources are abundant, Brownian walk is performed by animals whereas when preys are distributed into different patches Levy walk is performed (Sims et al., 2012). In (Nurzaman et al., 2011) a model of Levy and Brownian is presented, showing how Escherichia-Coli switches from Levy to Brownian mode based on target densities. Implicitly, the main question is what mechanism animals use to switch from one mode to another. As explained above, the seal search used to find other lairs (exploration) is in correlation with the presence of the external noise. However, in the opposite case where there is no external noise, the seal stays at the same lair and keeps exploiting the multi-chambered lairs. Based on this approach, the seal search can be divided into two states: normal and urgent. In each state, the individual exhibits a specific walk pattern (Levy or Brownian).

3.2.2.3. The Formal definition of the Sensitive Search Model

In the proposed approach, the movement of the seal pup inside its multi-chambered lair or during search for new lairs can be described as a series of events. Formally, let $(\Omega, \beta, \partial, \rho)$ be a search space that contains β predator and ∂ seal pup. In the interpretation, (Ω, ρ) is the state of the search space. If the current state of the search space ρ is ω where $\omega = 1$ (ω represents the external noise), then ∂ is informed that Ω contains β , which is a predator emitting a noise ω during movement. Given E event in Ω, a state (Ω, ρ) is called urgent state if Ω includes β and ∂ members of the event at the search space that contains the noise ω. Let A be an event where $(\Omega, \beta, \partial, \rho)$ is the search space. If the current state of the search space ρ is ω where $\omega = 0$, then ∂ is not informed that Ω contains β, then (Ω, ρ) is considered as a normal state. In urgent state ∂ performs a Levy walk, however in the opposite case (normal state) ∂ performs a Brownian walk. The movement of the seal pup from one lair to another can be described as Figure 3.4.

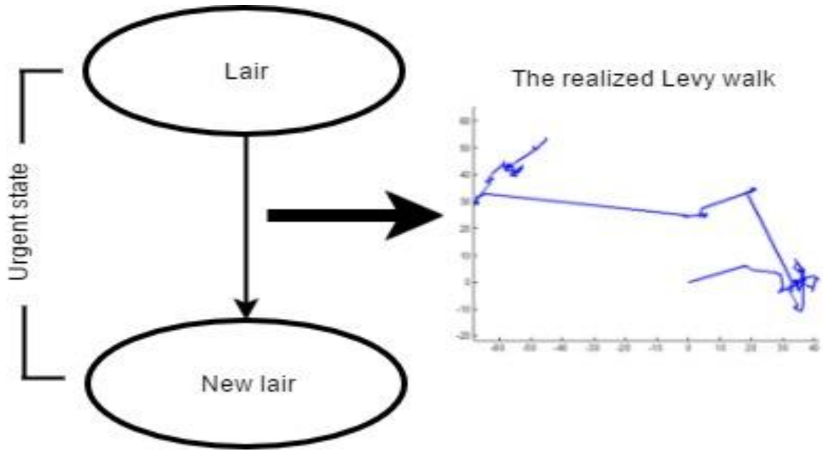


Figure 3.4: Seal Search during Urgent State

In the urgent state, the search pattern is characterized by an extensive search modeled by a Levy walk. Furthermore, in case of normal state the movement of the seal pup from one chamber to another can be described as Figure 3.5.

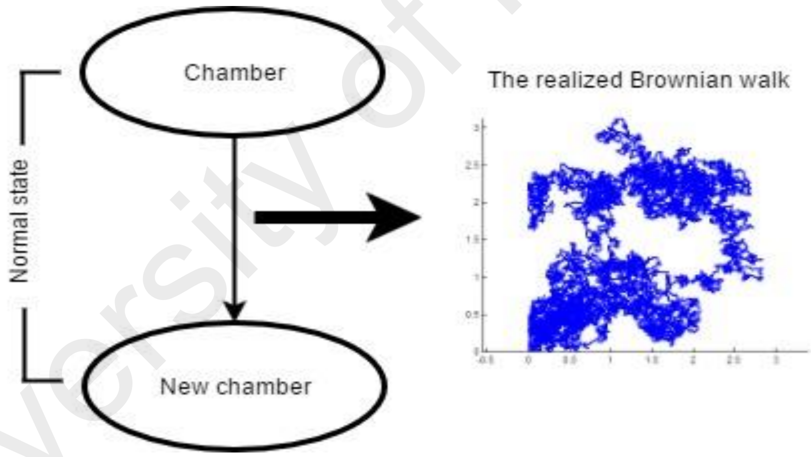


Figure 3.5: Seal Search during Normal State

The sensitive search model is composite search which can be illustrated as shown in the following figure.

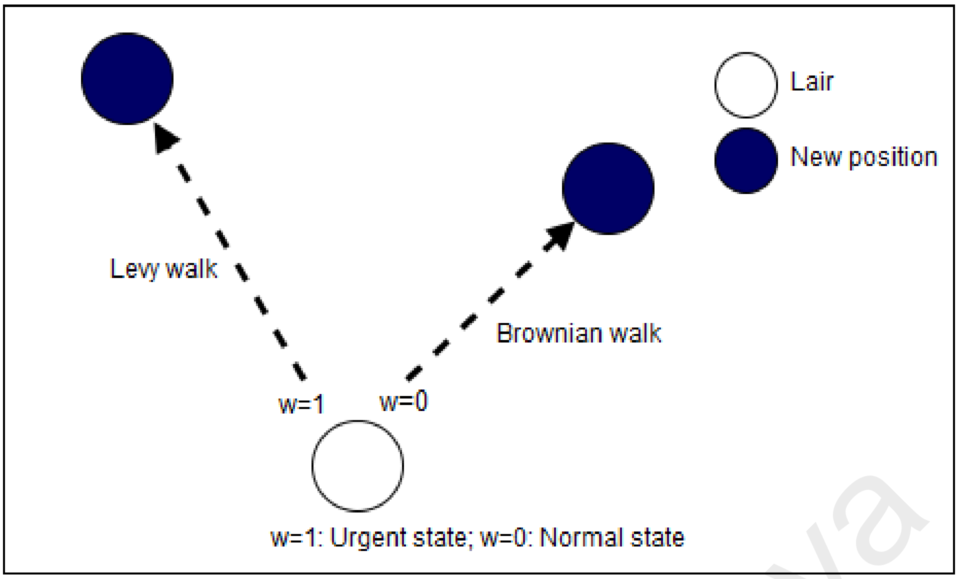


Figure 3.6: The sensitive search model

From Figure 3.6, moving from a lair to a new lair requires a specific search pattern. During the generation of new solutions (new lairs) $x^{(g+1)}$ for, say, a seal i, a new lair is found:

$$x_i^{g+1} = x_i^g + \alpha \oplus \Delta x\,, \tag{3.2}$$

where α is the step size which is related to the search pattern, during normal or urgent state.

$$\Delta x = \begin{cases} Levy(\lambda), & \omega = 1 \\ Brownian(\lambda), & \omega = 0 \end{cases}, \tag{3.3}$$

where ω is considered as a pseudo-random integer from a uniform discrete distribution. In case of Levy walk, the random walk is characterized by a step size calculated from a probability distribution with an inverse power-law tail as shown in equation 3.1. In case of Brownian walk, the search for a new chamber inside the structure of a multi-chambered lair as shown in Figure 3.5, the search is characterized by a step size described as below.

$$S = (k * \mathrm{randn}(d, \mathrm{Ndots})). \tag{3.4}$$

where k is the standard deviation of the normal distribution for diffusion rate coefficient, d is the dimensions of the problem and $Ndots$ represents the number of particles of the Brownian in the search space.

The sensitive search model is developed using the outputs of the comparative study and validated using a set of fifteen benchmark test functions. More details about testing can be found in Section 3.3.1.

3.2.2.4. Deriving the Ringed Seal Search

Ringed Seal Search (RSS) is particularly based on seal pup search for best lairs to escape predators. Everytime a new lair with a good quality is found, the pup will move into it. At the end, the lair (habitat) with the best fitness (quality) will be the term that RSS is going to optimize. The RSS scenario is based on the following representations:

1) Each female seal gives birth to one pup at a time in a specific habitat chosen randomly.
2) The seal pup moves randomly inside its ecosystem to find a good lair to escape predators.
3) The movement of the seal pup can take two states: Normal where the search is intensive using a Brownian walk or Urgent where the search is extensive using Levy walk.
4) If $L^{best,k}$ the best seen lair from the current K of the existing lairs, $L^{best,k}$ is better than $L^{best,k-1}$ the best of the previous iteration in term of fitness value, L^{best} is updated to be $L^{best,k}$, otherwise L^{best} remains with no update.
5) Gradually, worse lairs will be abandoned and seals continue moving to other lairs (or chambers) (convergence to good solutions).

The number of lairs is fixed where the mortality rate of seals is interpreted by the rate of lairs destruction which is equal to 15% (Kunnasranta et al., 2001). The complete algorithm is divided into three main parts. The first part corresponds to the initialization stage, while the remaining two stages represent the search for new solutions (lairs) and abandonment of worse lairs, respectively. All the optimization processes consists of a vector of values $L_i\left(i=1,2,\cdots,n\right)$ representing the initial solution. The overall process of optimization is described in Figure 3.7.

Ringed Seal Search algorithm
Input: Initial number of lairs
Output: Best lair

Begin

1. *Objective function:* $f(l), l=\left(l_1,\cdots,l_d\right)^T$
2. *Generate an initial number of birthing lairs,* $L_i\left(i=1,2,\cdots,n\right)$
3. ***While*** *(stopping criterion)*
4.   ***if*** *noise=false*
5. *Search in the proximity for a new lair by using a Brownian walk;*
6.   ***else***
7. *Expand the search far away for a new lair by using a Levy walk;*
8. ***Endif***
9. *Evaluate the fitness of each new lair and compare with the previous;*
10. ***If*** $L^{best,k} > L^{best,k-1}$
11. *choose the new lair,* $L^{best} = L^{best,k}$ *;*
12. ***else***
13. *go to 4*
14. ***endif***
15. *15% of the used lairs are detected and destroyed by bear, another new set will be selected randomly from nature;*
16. *Rank the solutions;*
17. ***Endwhile***
18. *return the best lair;*

end

Figure 3.7: The Ringed Seal Search algorithm

Figure 3.7 shows the main skeleton of RSS. The algorithm starts with an initial number of birthing lairs n. The pups move in the search space to get new lairs with better quality. For each new found lair, the fitness is evaluated based on the seal's decision to move, if the new lair is better than previous. After ranking the lairs, the RSS selects randomly 15% from the search space and replaces the worse lairs. At the end, according to the stop criterion, the RSS returns the best lair. The main steps of Figure 3.7 are described in details as below:

- ***Generating Initial Lairs***

Solving an optimization problem always starts with initial values. For that it is necessary that these initial values be formed as an array. Here in RSS algorithm the values represent the lair in which the seal pup is living. The lair is defined as below:

$$L_i, i = (1, 2, \cdots, n), \tag{3.5}$$

The lairs are distributed randomly, and each lair l contains many chambers m. For example, for a lair i, it is an array of $[1 \times m]$, representing current existing lair l of a habitat.

$$L = [1 \times m]. \tag{3.6}$$

Such values are randomly and uniformly distributed at the search space between the pre-defined lower bound Lb_j and upper bound Ub_j, as illustrated in the following expression:

$$L_i = Lb + (Ub - Lb).rand(size(Lb)) \tag{3.7}$$

$$i = (1, 2, \cdots, n)$$

where i represents the number of the lair and n indicates the number of the initialized lairs.

- ***Seal's Search for Lairs***

During each iteration, the pup performs a movement and selects a lair randomly (a new solution). The movement can be in two different patterns: in normal state using Brownian (intensive) or in an urgent state using Levy walk (extensive). For each mode there is a specific type of random walk, where the steps are determined in terms of the step length, with a specific probability distribution with the search direction being random. The main operators of search are described as below:

- ***Random Noise***

In order to simulate the random external noise emitted by predators, the proposed algorithm generates a uniformly distributed pseudorandom integer to model the noise ϖ. The noise ϖ takes two values: $\varpi = 0$ and $\varpi = 1$. If $\varpi = 0$, the search space state (Ω, ρ) will be in a normal state, and an intensive search (exploitation) will be performed by the seal pup at the proximity of the multi-chambered lair. By contrast if $\varpi = 1$, the search space (Ω, ρ) state will be in a urgent state, as a result the seal performs an extensive search to find a new lair (exploration).

- ***Normal State***

In the normal state, the random noise value $\varpi = 0$, then the seal ∂ experiment a normal behavior and search. Such state is characterized by a random movement at the proximity of the multi-chambered lair. Therefore, the movement is modeled via a Brownian walk as a non-heavy tailed step length that can be interpreted as an intensive search (exploitation).

- ***Urgent State***

In the urgent state, the seal is threatened by the external noise emitted. As a result, the search space (Ω, ρ) takes an urgent state affected by the external noise emitted

$\omega = 1$. This state is characterized by an extensive walk in the search space. It is modeled via a Levy walk which is known by a heavy tailed step length distribution which is suitable for search in case of sparse targets. The nature of the urgent state case stimulates the seal to move outside the lair, and tries to get another solution to escape the predator's threat.

- ***Best Lairs Updating***

Despite the fact that this updating process is not a part of the Sensitive Search Model, it is used to simply select and store the best so far solution (lair) found. In order to update the best lair L^{best} found so far, the best seen lair from the current K of the existing lairs $L^{best,k}$ is compared to the best lair of the previous iteration $L^{best,k-1}$. If $L^{best,k}$ is better than $L^{best,k-1}$ according to its fitness value, L^{best} is updated to be $L^{best,k}$, otherwise L^{best} remains with no update. Thus L^{best} memorizes the best historical lairs found so far.

- ***Abondoning Worst lairs***

After all the seals have moved to new lairs, certain lairs with high smell index will be detected and destroyed by the bear. The percentage of destruction of lairs (mortality rate of seals) is set to 15% by default, and it can be modified according to the nature of optimization problem. These abandoned lairs will not be suitable to host pups again and will be abandoned definitively. The rest of the lairs will host pups until the pups decide to leave due to one of the reasons below:

a. The snow covered the lair has melted.

b. A predator attacks the lair, so the seal escapes the area.

Another interesting feature for seal pups is that one lair can be used communally by different seal pups (Kunnasranta et al., 2001), something that occurs rarely in nature.

- ***Convergence to optimal lairs***

After certain iterations, all the seal pups move to a new lair (new solutions), which is better than the previous locations. These newfound lairs will provide better protection to the pups to avoid the predator's threat. As a consequence, there will be less killed seal pups by predators, which can ensure the reproduction of new generations. The fast convergence to the optimal locations (lairs) ends the RSS algorithm quickly.

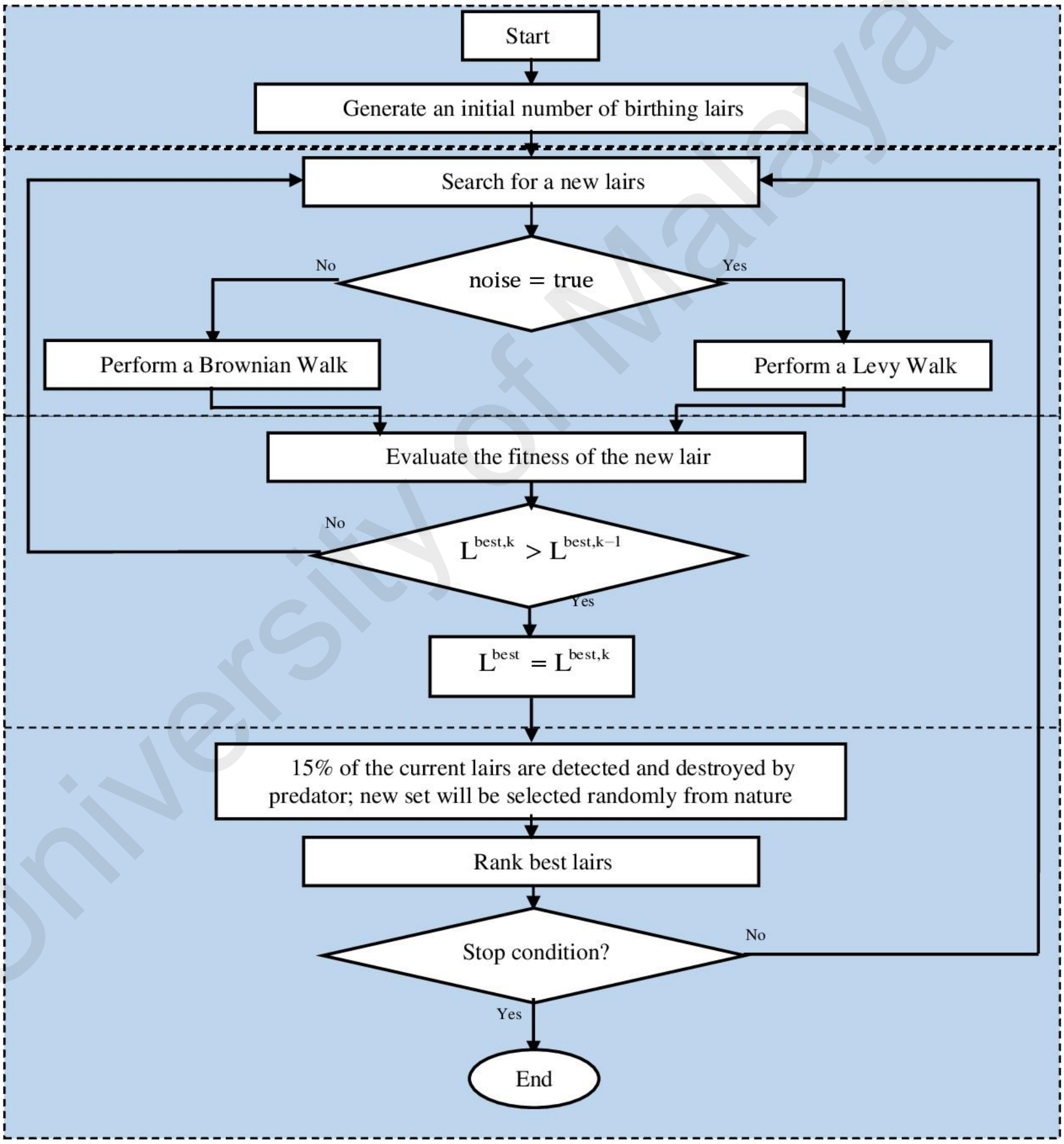


Figure 3.8: Data flow in the RSS algorithm

Figure 3.8 below shows the data flow of the proposed algorithm. Like other metaheuristic algorithms, the proposed algorithm starts with initial birthing lairs containing seal pups. To make the terminology clear and easy, the following simple terms were used. Each lair represents a solution. The quality of the lairs represents the quality of the solution, and thus the suitability of the lair for seal pupping. The RSS in this study can be described as an iterative algorithm based on population. Despite other population-based algorithms such as GA, where the reproduction of new generations ensures generating new solutions, the RSS is based only on seal pups life cycle. As all population based algorithms, RSS starts with an initialized number of lairs.

Certain studies about asymptotic probability convergence theories considering the underlying operations which are characterized by a Markov nature, requires to be balanced, and thus resulting in the algorithm wasting a lot of its efficiency (Benhamou, 2007; Nolting et al., 2015; Yang et al., 2014). The power of stochastic algorithms mainly is based on the fact that the probabilistic natures of the algorithms guarantee that the algorithms do not necessarily get trapped at local optima (Benhamou, 2007; Nolting, 2013; Xu & Zhang, 2014). The RSS consists of two search states that alternate randomly via the noise emitted by predators. This can provide a balance between exploitation and exploration of the search, and thus the probability to get local optima easily is very low.

3.3. Evaluation Method

In order to evaluate the performance of the proposed algorithm, benchmark test functions and a real data case study were applied on the proposed algorithm to measure its capabilities in terms of solving optimization problems.

3.3.1. Benchmark test Functions

In metaheuristic algorithms, the only way to test new proposed algorithms is based on using benchmark test functions, where the performance is measured by using several parameters. A comprehensive set of fifteen benchmark test functions were used to test the performance of the proposed algorithm as practice in the literature (Adorio & Diliman, 2005; Jamil & Yang, 2013; X. Li et al., 2013; Liang, Qu, Suganthan, & Hernández-Díaz, 2013; Molga & Smutnicki, 2005; Rajabioun, 2011; Rardin & Uzsoy, 2001; Yang, 2012).

Table 3.1: List of Benchmark Test Functions

Test function	F_id	S	f_*	n
$f(x)=\sum_{i=1}^{n} x_i^2$	F_1	$[-5.12,5.12]$	0	30
$f(x,y)=-\cos(x)\cos(y)\exp\left[-(x-\pi)^2-(y-\pi)^2\right]$	F_2	$[-100,100]$	-1	30
$f(x)=\sum_{i=1}^{n}\left[-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right]$	F_3	$[-500,500]$	$-418.98n$	30
$f(x)=-20.\exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}+(20+e)\right)$	F_4	$[-32.76,32.76]$	0	30
$f(x,y)=-\sin(x)\sin^{2m}\left(\frac{x^2}{\pi}\right)-\sin(y)\sin^{2m}\left(\frac{2y^2}{\pi}\right)$	F_5	$[0,5]$	-1.8013	30
$f(x)=10n+\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)\right]$	F_6	$[-5.12,5.12]$	0	30
$f(x)=\sum_{i=1}^{n}\sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{2m}$	F_7	$[0,\pi]$	-4.6877	30
$f(x,y)=\sum_{i=1}^{5}i\cos[(i+1)x+1]\sum_{i=1}^{5}\cos[(i+1)y+1]$	F_8	$[-10,10]$	-186.730	30
$f(x,y)=\sum_{i=1}^{n-1}\left[(1-x_i)^2+100(x_{i+1}-x_i^2)^2\right]$	F_9	$[-10,10]$	0	30
$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	F_{10}	$[-600,600]$	0	30
$f(x)=(x_1-1)^2+\sum_{i=2}^{d}i(2x_i^2-x_{i-1})^2$	F_{11}	$[-10,10]$	0	30
$f(x)=\sin^2(\pi\omega_1)+\sum_{i=1}^{d-1}(\omega_i-1)^2\left[1+10\sin^2(\pi\omega_i+1)\right]+$ $(w_d-1)^2\left[1+\sin^2(2\pi\omega d)\right]$	F_{12}	$[-10,10]$	0	30
$f(x)=\sum_{i=1}^{d}\left(\sum_{j=1}^{d}(j^i+\beta)\left(\left(\frac{x_i}{j}\right)^i-1\right)\right)^2$	F_{13}	$[-d,d]$	0	30
$f(x)=\left(\sum_{i=1}^{5}i\cos((i+1)x_1+i)\right)\left(\sum_{i=1}^{5}i\cos((i+1)x_2+i)\right)$	F_{14}	$[-10,10]$	-186.730	30
$f(x)=\sum_{i=1}^{n}x_i^2$	F_{15}	$[-5.12,5.12]$	0	30

Table 3.1 presents the benchmark test functions used in this experimental study. The selected functions fulfill the requirements of uni-objective and multi-objective

problems, and thus it is very important to highlight that the main target of this benchmarking test is to check whether the proposed RSS algorithm is able to solve uni-objective and multi-objective optimization problems.

The values of n represent the dimension of the problem, f_* indicates the optimum value of the test function, and S indicates the search space bounds. The optimum values of the function $F_1, F_4, F_6, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{15}$ is at $f_* = 0$, for F_2 is at $f_* = -1$, for F_3 is at $f_* = -418.982n$, for F_5 is at $f_* = -1.8013$, for F_7 is at $f_* = -4.6877$, for F_8 is at $f_* = -186.730$, for F_{14} is at $f_* = -186.730$.

The test of efficiency and validation of new optimization algorithms is often implemented using a set of standard benchmark test functions selected from the literature. The number of used test functions in majority of published papers is varied between few to about twelve test functions (Jamil & Yang, 2013). Moreover, they should be diverse in term of search and unbiased.

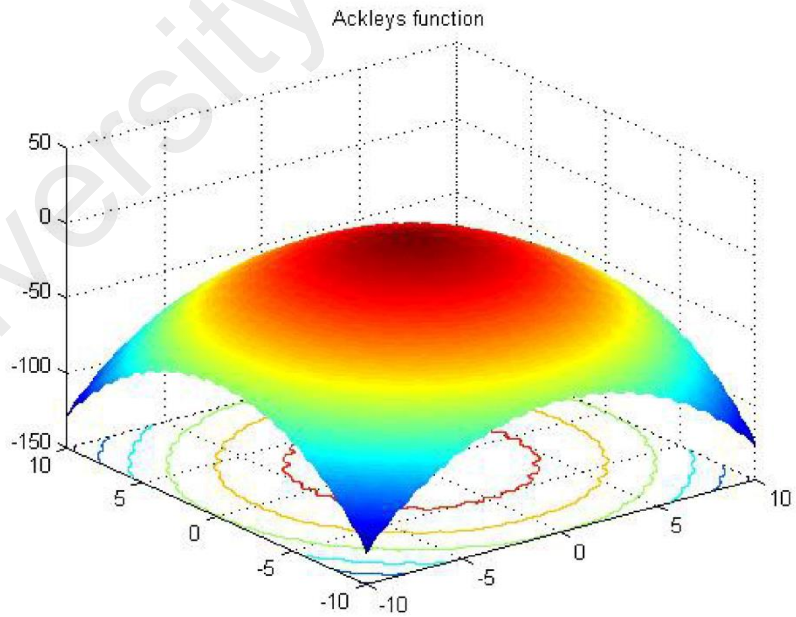


Figure 3.9: Example of Uni-modal function by Ackley's function (Adorio & Diliman, 2005)

It is very important to highlight that there is no specific standard set of test functions in the literature (Jamil & Yang, 2013). A function with unique optimum is called uni-modal as shown in Figure 3.9. The multi-modal functions are characterized with more than one local optimum as illustrated in Figure 3.10 below. It is shown that multi-modal functions are considered as the most difficult category of optimization problems (Jamil & Yang, 2013). These functions are used to examine the capability of the proposed algorithm to escape local optima traps. In case where the balance between exploration and exploitation of the search space is not designed well, the algorithm won't be able to search the landscape of the test function effectively.

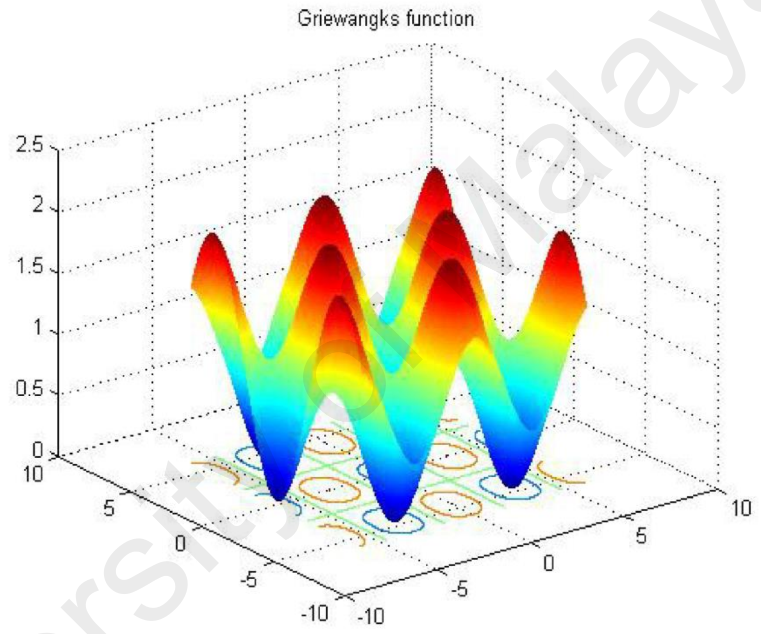


Figure 3.10: Example of Multi-modal function by Griewangk's function (Molga & Smutnicki, 2005)

All the benchmark test functions are presented in Appendix A. (Please refer to Figures A.1 to A.14 in Appendix A).

3.3.2. Statistical Measurements

The reported results in the experimental results chapter are featured with the following performance indexes during 100 iterations: the Average Best-so-far (AB) solution, the Median Best so-far (MB), the Standard Deviation (SD) of the best-so-far solution, the Variance (Var) of the best-so-far solution and the Solution Quality (SQ) for each function. During each run the best value of M is saved; thus during 100 times run, 100 best values are produced. The Average best is computed from the mean of 100 best values. The Median best is the midpoint of 100 best values. The SD is the standard deviation of 100 best values and the Var is just the square of the SD . The mathematical formulations are defined as below (Cuevas et al., 2014; Leung & Wang, 2001; Yang, 2011):

$$AB = \frac{\sum_{i=1}^{n} f_{best\ i}}{n} \tag{3.8}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{n} \left(f_{best\,i} - AB\right)^2}{n-1}} \tag{3.9}$$

$$MB = f_{best\,k}\text{, where } k = \left(\frac{n+1}{2}\right) \tag{3.10}$$

where $f_{best\,i}$ is the best value of each run and n is the number of run. The solution quality is the final fitness value at the end of 100 iterations. The convergence speed is represented by the number of iterations required to reach the goal (BoussaïD et al., 2013; Senington, 2013; Yang, 2011; Yazdani & Jolai, 2016).

$$Conv = \sum_{i=1}^{n} iter_i \tag{3.11}$$

where *iter* denotes iterations and *n* denotes the number of iterations. These metrics are used in order to show to what extent the solution quality and the convergence speed are improved compared to the existing methods. The focus of this study is on improving the solution quality and convergence speed; however, the variation and time consumption are also calculated to prove the maturity and diversity of the proposed algorithm. For maturity, it is assessed using solution quality and convergence speed. Diversity is assessed using the output of variations to measure the spread of the search. The pseudo code for all methods is provided separately. The results are compared with those obtained by the baseline algorithms. Figure 3.11 depicts the evaluation method of the RSS algorithm.

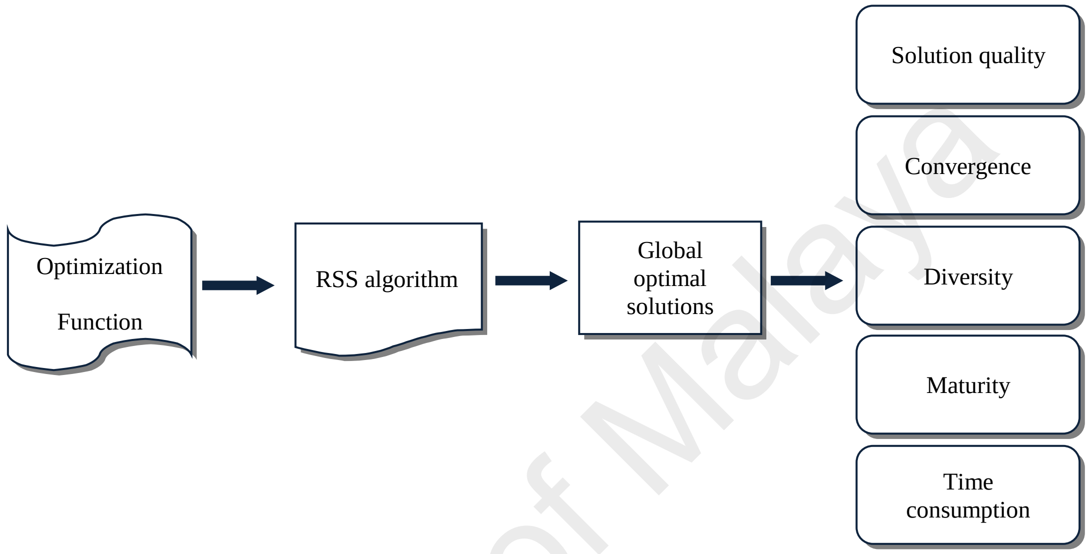


Figure 3.11: Evaluation steps

From Figure 3.11, the evaluation steps required to measure the performance of the proposed algorithm RSS are based on computing five metrics, which will be explained in details in the next chapter.

3.3.3. Case Study: RSS for Data Clustering

In order to validate the proposed algorithm and check whether the proposed RSS algorithm is able to solve real optimization problems, a case study for data clustering is presented. A total of seven datasets from University of California Irvine UCI datasets website (Lichman, 2013; Newman, Hettich, Blake, & Merz, 1998) were used to evaluate the quality of the clustering-based RSS. More details about each component of the case study are explained in Chapter 6.

3.4. Summary

The chapter explains the research methodology of the thesis. A search model namely the sensitive search model and a derived algorithm is developed based on the problem statement and the research objectives. The algorithm is called RSS which is a metaheuristic algorithm nature-inspired with the ability to solve global optimization problems while it is featured by an adaptive balance between exploitation and exploration. In RSS, a composite search model called the sensitive search model is proposed by modeling the movement of the ringed seal to solve the issue of balancing exploitation and exploration. In the proposed approach, the core engine is based on the sensitive search model, in which the search switches between normal state and urgent state. The states are modeled by using Brownian walk to perform exploitation and Levy walk to perform exploration. For modeling the noise, which switches between both states normal and urgent, a random Gaussian distribution is used to adaptively switch the search states. Furthermore, the evaluation method is explained, which is based on fifteen benchmark test functions with five metrics measured to evaluate the performance of the proposed algorithm. In order to validate RSS, a data-clustering case study is presented in Chapter 5. A total of seven datasets from UCI are used to evaluate the quality of the clustering-based RSS.

CHAPTER 4: EXPERIMENTAL EVALUATION AND ANALYSIS

4.1. Introduction

This chapter discusses the experimental evaluation of the proposed algorithm. The implementations of the proposed RSS algorithm as well as the experiments were implemented as described in Chapter 3. The proposed algorithm RSS is derived from the sensitive search model, then a series of experiements is realized to compare between the proposed algorithm and the baseline algorithms.

The proposed RSS algorithm has been compared with GA, PSO and CS in order to solve standard benchmark functions representing several types of optimization problems. Evaluating of global optimization results is one of the difficulties in machine learning. As explained in Chapter 3, there are some evaluation metrics for measuring global optimization performance. The evaluation process of RSS includes:

- **Time Consumption Analysis:** In order to use a method that is not dependent on machine performance, time consumption is measured using number of iterations as practice in the literature (Cuevas et al., 2014; Mirjalili, 2016; Rajabioun, 2011).
- **Convergence Analysis:** The convergence speed visualized and analyzed in order to check the speed quality of optimization (Gao et al., 2015; Rajabioun, 2011).
- **Diversity Analysis:** Diversity refers to spread of search during search for solutions. It is measured by using variations of the obtained solutions (Gao et al., 2015; Tan, Shi, & Niu, 2016).

- **Solution Quality Analysis:** The solution quality is computed for two reasons: the first to measure the quality of the global optimum, the second is to measure the maturity of the search (Eiben & Smit, 2011; Gao et al., 2015).
- **Maturity Analysis:** It is measured by using the convergence and the solution quality of the search (Črepinšek et al., 2013; Cuevas et al., 2014).

4.2. Implementation and Environment

The RSS is implemented as well as the comparison with the GA, PSO and CS algorithms in Matlab with graphical interface of R language. All the source codes are presented in Appendix B. The GA, PSO and CS are among the remarkable algorithms that are used as benchmark for global optimization algorithms (Fong et al., 2018; Gandomi, Yang, & Alavi, 2013; Juan et al., 2015; Sheng, Wang, Zhou, & Zhou, 2014; Yang & Deb, 2009). The GA, PSO and CS have a long chain of comparisons with existing algorithms in the literature (Cuevas et al., 2014; Findik, 2015; Fong et al., 2018; Mirjalili, 2015, 2016; Mirjalili et al., 2014; Rajabioun, 2011; Yang, 2010b; Yang & Deb, 2009). All experiments were conducted on Intel i3 CPU 1.90 GHZ machine with 4GB memory, running on Windows10.

4.3. Performance comparison with other metaheuristic algorithms

A description of the Bivariate Michalewicz test function F_5 (Molga & Smutnicki, 2005) is presented in Figure 4.1. It is considered as a multi-objective function, which have n local optima. This function is characterized by the parameter m that defines the ruggedness (steepness) of the valleys. The setting of m with a large value conducts to uneasy search. When the m value is very large, the functions perform as a needle in the haystack, something that is so difficult to find, especially because the area of search is too large (Jamil & Yang, 2013).

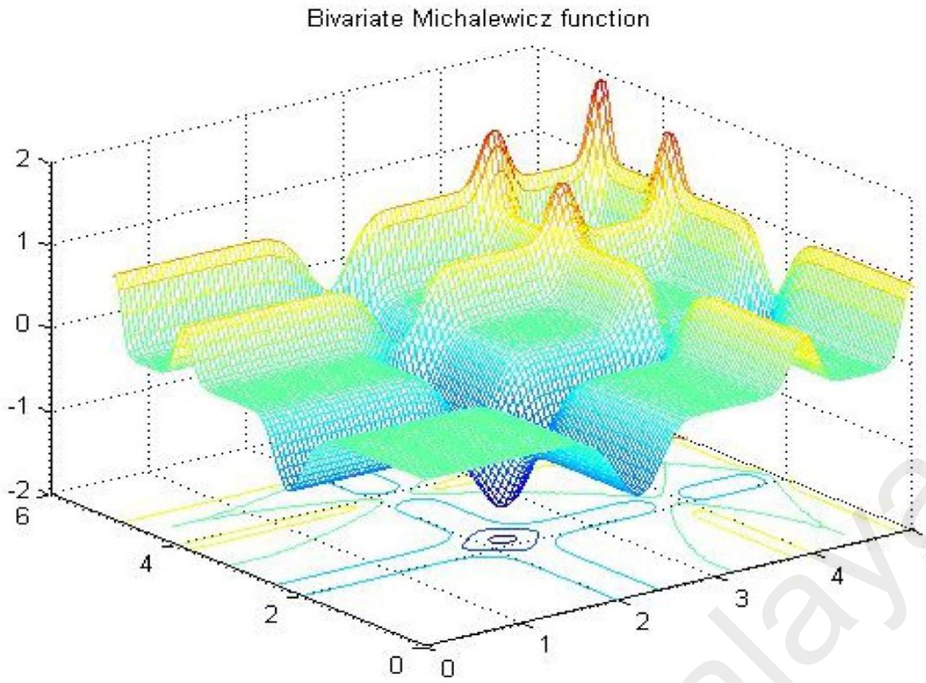


Figure 4.1: Plot of F_5 function in 2D for $m=10, n=5$

The search area is bounded to a hypercube, where $(x,y)\in[0,5]\times[0,5], i=1,\cdots,n$, and the value of m=10. The global minimum is approximated by $f_* = -1.8013$ for $n=2$. The landscape of the function F_5 equation is described in Figure 4.2.

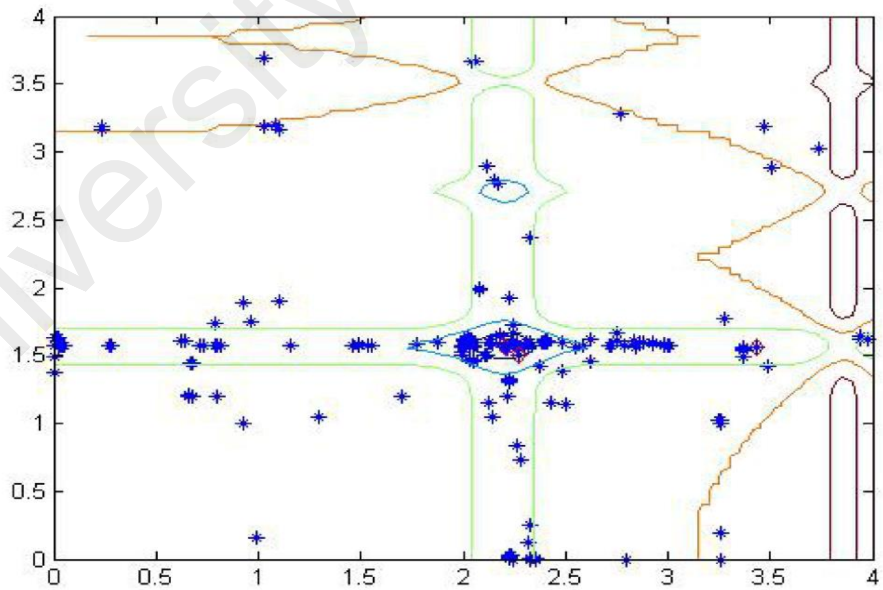


Figure 4.2: Searching for a new solutions by using Ringed Seal Search, final achieved solutions are highlighted with a diamond

From Figure 4.2, shows that the used lairs are converging towards the global optimum. The figure also shows that the lairs are distributed at different local optima. This feature demonstrates the ability of RSS to deal with multi-objective problems and escape local optima traps. Escaping from local optima is particularly related to the optimal balance between exploitation and exploration, which is realized via the sensitive search model. As a result, modeling of the external noise via a uniform distributed pseudorandom integer is efficient for the imitation of the switch between normal state and urgent state.

In order to confirm the efficiency of the sensitive search model to deal with uni-objective and multi-objective problems, a series of simulations were tested with a varied number of lairs: $l = 5$ until $l = 200$. The results show that an efficient result for the majority of the optimization problems is achieved when $l = 10$. The results also show that the convergence is not affected by changing the parameters values. The following section introduces the performance of the proposed algorithm to other metaheuristic algorithms based on the standard problems (test functions).

The RSS algorithm is applied to fifteen test functions whose results have been compared to those obtained by the GA, the PSO and the CS. These algorithms are considered as the most popular baseline approaches in many optimization applications (Cuevas et al., 2014; Rajabioun, 2011). In order to make the comparison more valuable CS was selected. CS is considered as a hill climbing variant that comprises the brood parasitic behavior of cuckoo birds, and it uses Levy flights to enhance the balance between exploitation and exploration of the search space (Yang, 2010a). The overall comparison and the parameters of each algorithm were set to be compatible with their original setting. The maximum number of iterations for the test functions was set to 100. These criterion also have been selected to fulfill the requirements of similar works highlighted in the literature (Cuevas et al., 2014; Juan et al., 2015; Mafarja & Mirjalili, 2017; Mirjalili, 2016; Rajabioun, 2011; Shahrzad Saremi et al., 2017; Yang & Deb, 2009; Yazdani & Jolai, 2016).

4.3.1. Parameter Setting

The parameters setting for each algorithm during this comparison described below:

1) GA: the parameters of GA are set to $G_i = 100$ and the population size $\alpha = 20$; where the total number of iterations is set to 100 for all the test functions.
2) PSO: the velocity, social and cognitive parameters are set to 2.
3) CS: The parameters consist of the number of the nest, which is set to 15 nests, and the rate of detection $p_\alpha = 0.25$.
4) RSS: Two parameters have been tuned up in RSS: the mortality rate of the seal pups: $rate = 15\%$, and the initial number of the birthing lairs: $l = 10$.

The experimental comparisons between these metaheuristic algorithms with the proposed RSS were developed according to the type of the test function: uni-objective such as F_1 and F_2, or multi-objective such as: F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{14} and F_{15}. The reason behind choosing only two uni-objective functions is that uni-objective problems are easy to solve (the landscape is not complex). In some literatures it is called smooth problem containing only one global optimum. In contrast, the multi-objective problems were tested with eight test functions representing variant complex problems. The test consists of comparing the RSS with other algorithms such as GA, PSO and CS. The reported results in the next sections are featured with the following performance indexes during 100 iterations: the Average Best-so-far (AB) solution, the Median Best so-far (MB), the Standard Deviation (SD) of the best-so-far solution, the Variance (Var) of the best-so-far solution and the Solution Quality (SQ) for each function. During each run the best value of M is saved; thus during 100 times run, 100 best values are produced. The Average best is computed from the mean of 100 best values. The Median best is the midpoint of 100 best values. The SD is the standard deviation of 100 best values and the Var is just the square of the SD.

4.3.2. Time Consumption Analysis

In order to measure the capability of the RSS to achieve the optimal solution, the number of iterations is computed to measure the time required to find the optimal solution. In Figures 4.3 to 4.6, the test function F_9 is used as an example to demonstrate the time consumption performance of GA, PSO, CS and RSS, where less iterations refers to less time consumption and vice versa.

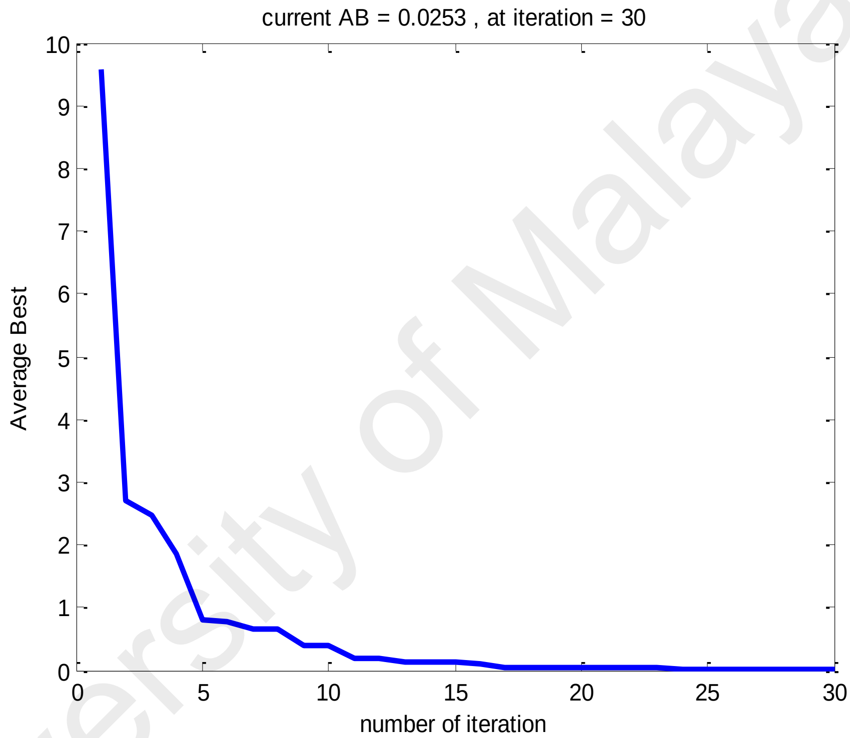


Figure 4.3: Time consumption of F_9 using RSS

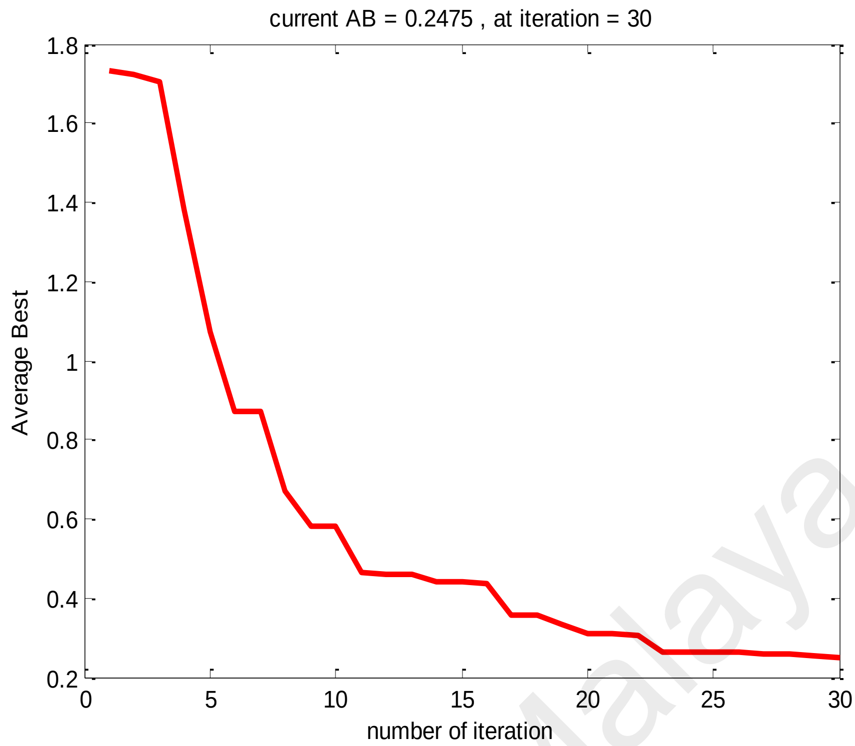


Figure 4.4: Time consumption of F_9 using CS

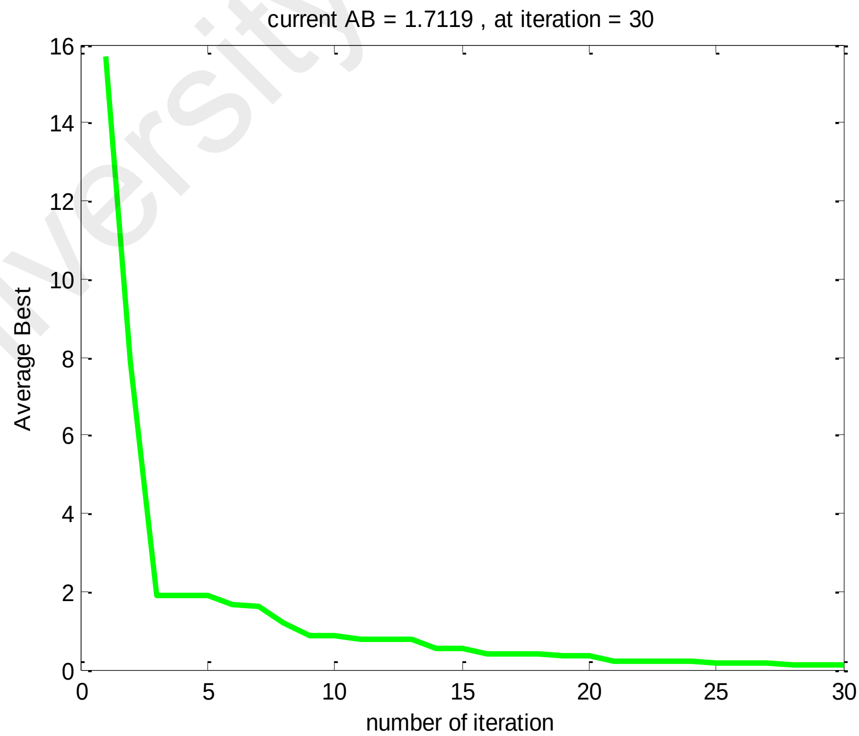


Figure 4.5: Time consumption of F_9 using PSO

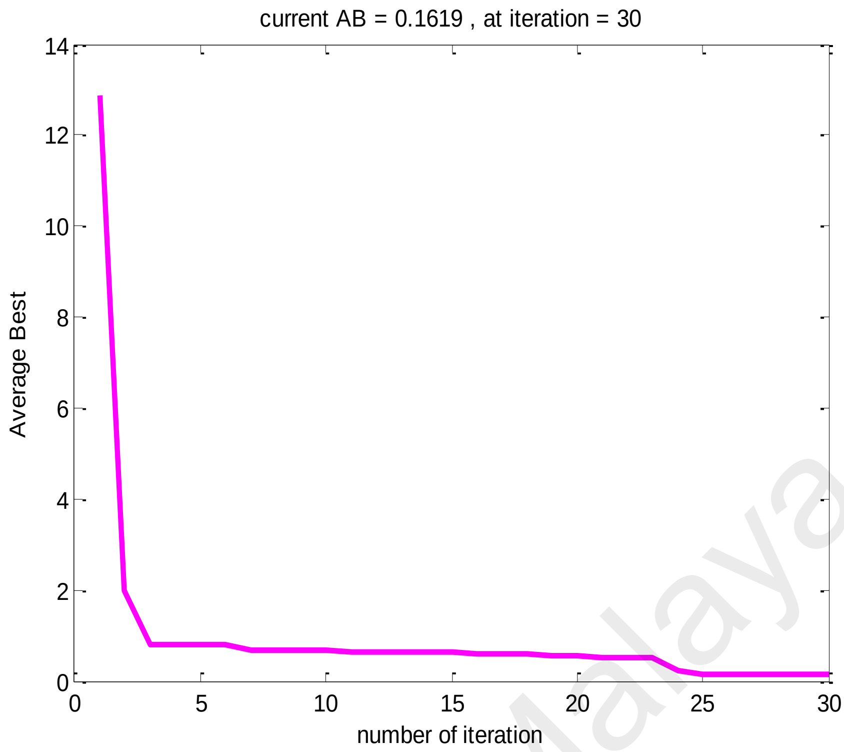


Figure 4.6: Time consumption of F_9 using GA

The Figures 4.3 to 4.6, show a sample of time consumption plot of F_9 function using RSS, CS, PSO and GA during 30 iterations. It can be seen clearly in Figure 4.3 that RSS has reached a global minimum before the 30th iteration and the obtained AB value is 0.0253. On the other hand, with a same landscape problem, in Figure 4.4, CS algorithm approximately reached a global minimum at the 30th iteration with a value AB equal to 0.2475.

The time consumption of F_9 function using PSO achieved almost the global minimum at 30th iteration and the AB value is 1.7119 as shown in Figure 4.5. On the other hand, Figure 4.6 shows GA reached a global minimum at 30th iteration and the AB value is 0.1619. In conclusion, the F_9 test function shows how the RSS converged quickly to the global optimum compared to PSO, GA and CS. For more test results, GA, PSO, CS and RSS were applied on more test functions mentioned in the next sub-sections to compare the performance for each algorithm. The test functions divided into two groups: uni-objective test functions and multi-objective test functions.

4.3.3. Uni-Objective Test Functions

This experiment is applied on uni-objective test functions F_1 and F_2 in order to find the global minimum, where GA, PSO, CS and RSS are required to find the minimum value for each test function. The minimum values can be positive or negative, depending on the nature of the test function. It is known that uni-objective test functions are smooth problems and easy for finding the global minimum or maximum values. In this category of test functions, the challenge is only to find one global minimum.

Table 4.1: AB, MB, SD, Var and SQ results using uni-objective functions

Function ID		GA	PSO	CS	RSS	Optimal Value f_*
F_1	AB	0.2096	0.2096	0.1584	**0.1516**	0
	MB	**0.0537**	**0.0537**	0.9852	0.9472	
	SD	0.0360	0.0360	0.1858	**0.1850**	
	Var	0.0012	0.0012	0.0345	**0.0342**	
	SQ	0.0257	0.0001	**0**	**0**	
F_2	AB	**-0.9934**	**-0.9934**	-0.6262	-0.6299	-1
	MB	**-1.0000**	**-1.0000**	-0.8049	-0.8141	
	SD	0.0660	0.0660	0.3894	**0.3892**	
	Var	0.0043	0.0043	0.1516	**0.1514**	
	SQ	-0.0377	-0.9724	-0.7039	**-0.9962**	

From Table 4.1, the optimal minimal values of AB, MB, SD, Var and SQ are reported. In F_1 the RSS outperforms GA PSO and CS in terms of AB although in MB GA and PSO achieved 0.0537 and RSS achieved 0.9852, this difference is justified by a better SQ equals to 0 and it demonstrates that SD and Var achieved by RSS reflects an optimal exploration (spread out) better than GA, PSO and CS.

In F_2, the GA and PSO achieved better global minimum in terms of AB and MB compared to RSS. However, GA, PSO and CS provide low solution quality, which indicate a premature convergence for GA, PSO and CS. In contrast, RSS shows a mature convergence, where the global minimum is lower in terms of AB and MB but it achieves an optimal variance Var = 0.1514 and better solution quality SQ compared to GA, PSO and CS. This can be justified by the fact that uni-objective test functions do not require a lot of exploration, which is the case for RSS where the search is divided into two main states: exploration and exploitation. This result can have an accurate interpretation when it is combined with the convergence rate, solution quality and the variance. The SQ and Var results will be analyzed in the following sub-sections.

The visualization of the obtained results of AB, MB and SD from F_1 and F_2 will provide more insights about the comparison of the GA, PSO, CS and RSS.

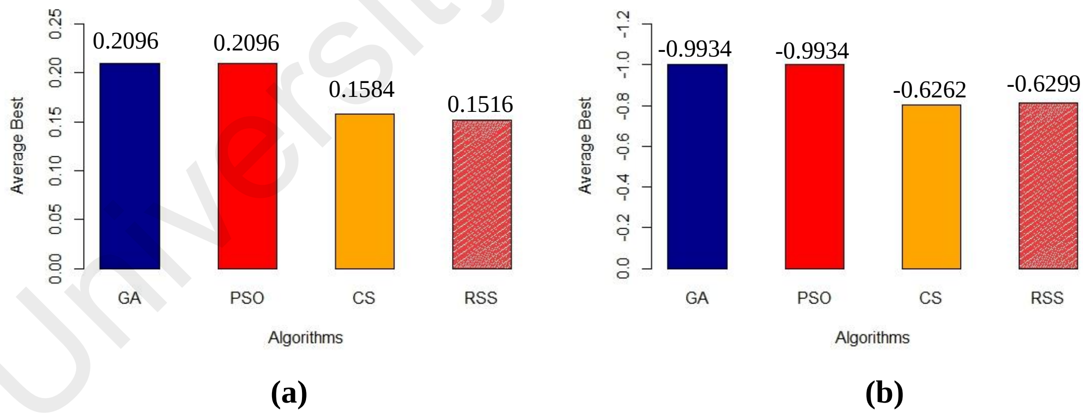
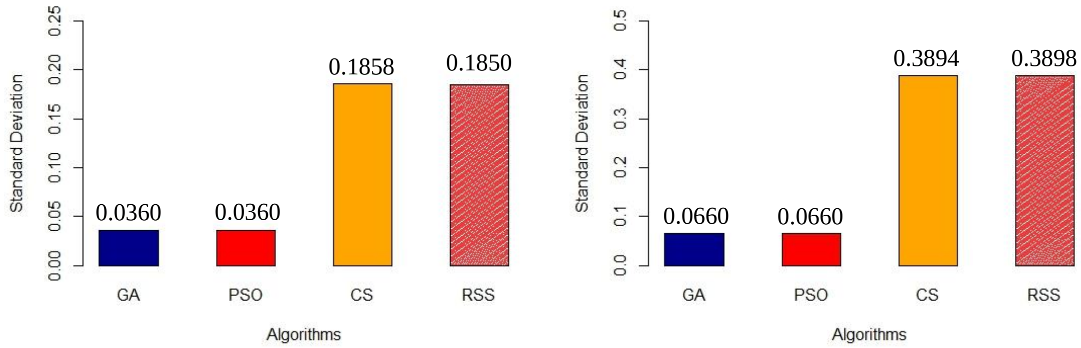
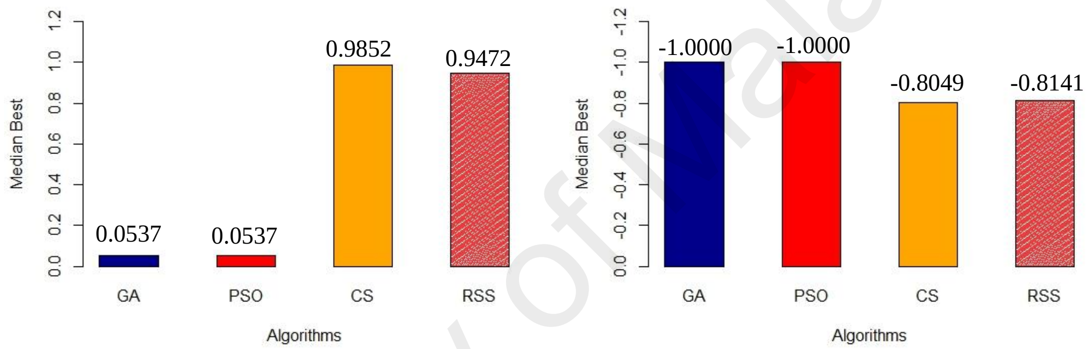


Figure 4.7: Average best of the global optimal for (a) F_1 and (b) F_2



(a) **(b)**

Figure 4.8: Standard deviation of the global optimal for (a) F_1 and (b) F_2



(a) **(b)**

Figure 4.9: Median best of the global optimal for (a) F_1 and (b) F_2

Figures 4.7, 4.8 and 4.9 illustrate the plot of AB, MB and SD outputs of the global minimum of F_1 and F_2. One may wonder that this visualization does not illustrate the maturity or premature convergence of GA, PSO, CS and RSS. The answer is that more analysis will be introduced in the next sub-sections when the corresponding results will be combined with SQ and Var and analyzed accordingly. In the following subsection, the convergence is measured and analyzed.

4.3.3.1. Convergence Analysis

The convergence of the GA, PSO, CS and RSS is computed in order to measure the speed capability of RSS compared to other algorithms in terms of finding the global minimum. The smaller the number of required steps, the higher the convergence speed as illustrated in Figure 4.10 and Figure 4.11.

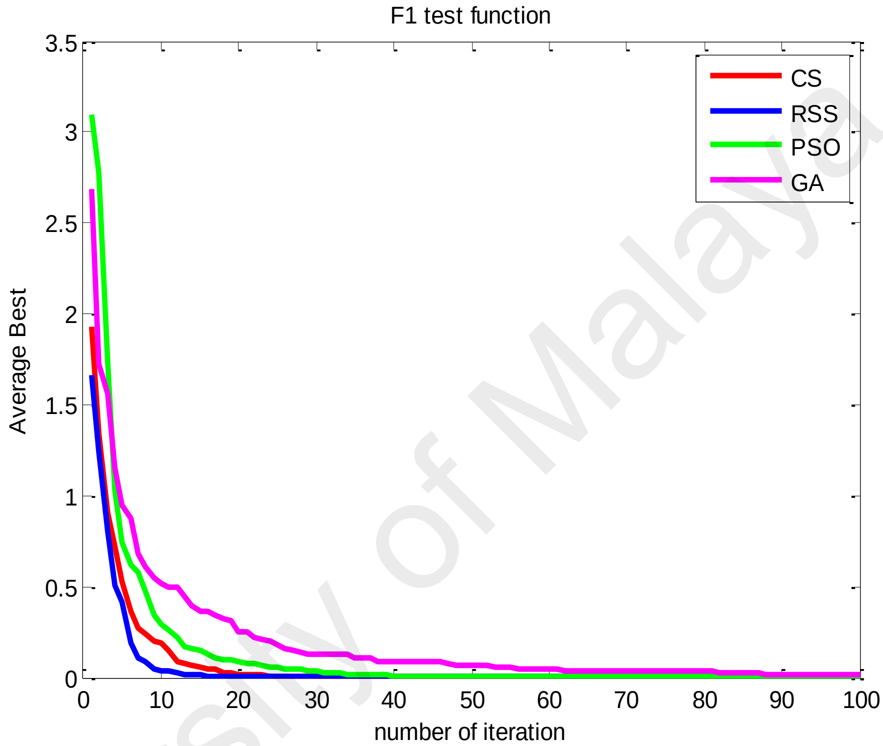


Figure 4.10: Average best convergence of F_1

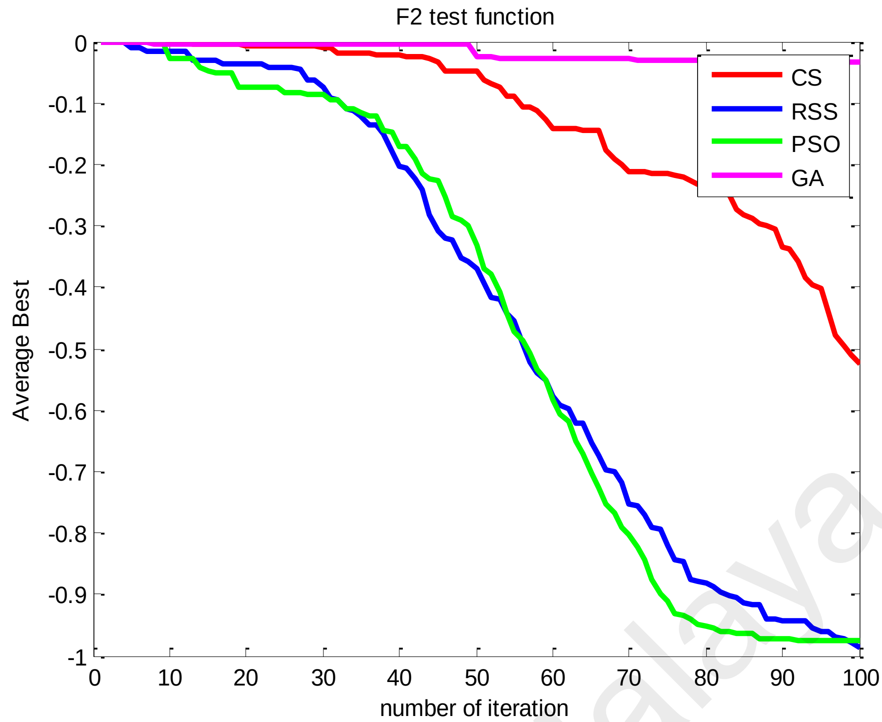


Figure 4.11: Average best convergence of F_2

From Figure 4.10 and Figure 4.11, the RSS outperforms CS, PSO and GA for both F_1 and F_2. The difference in terms of convergence can be seen clearly in F_1 where the optimal value is $f_* = 0$. However, in F_2 where the optimal value is $f_* = -1$ the convergence of RSS is slow at the beginning compared to PSO but it overtakes PSO before the end of the iterations. This outperformance of RSS in terms of convergence can be interpreted by the fact that the RSS is based on a composite search modeled using the sensitive search model, where Levy walk and Brownian walk were employed to model a balanced exploitation-exploration. For more performance tests, the variance is calculated to measure the dispersion of the achieved solutions.

4.3.3.2. Diversity Analysis

In order to evaluate the diversity of the search, the variance outputs of the achieved solutions were analyzed and reported in Table 4.1. The results of F_1 and F_2 were plotted in Figure 4.12 and Figure 4.13 in order to compare the performance of RSS with GA, PSO and CS.

Figure 4.12 and Figure 4.13 show the variance achieved during searching for the optimal solution by using the test functions F_1 and F_2. It is noticed that RSS and CS outperformed GA and PSO. High variance refers to high diversity, where the similar diversity result between RSS and CS is related to the fact that both RSS and CS used Levy for exploration of the search.

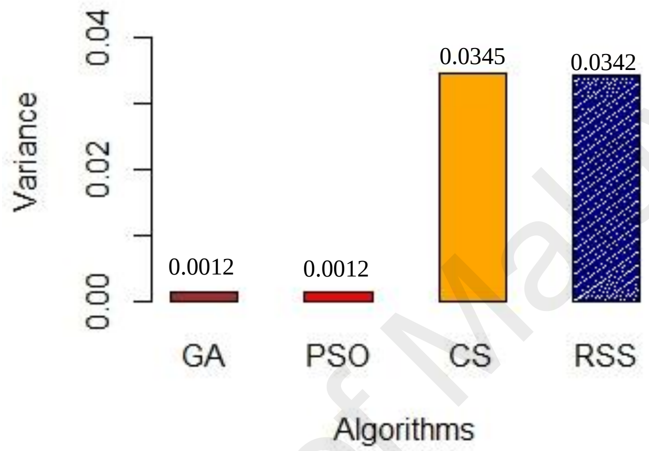


Figure 4.12: Variance results by using F_1

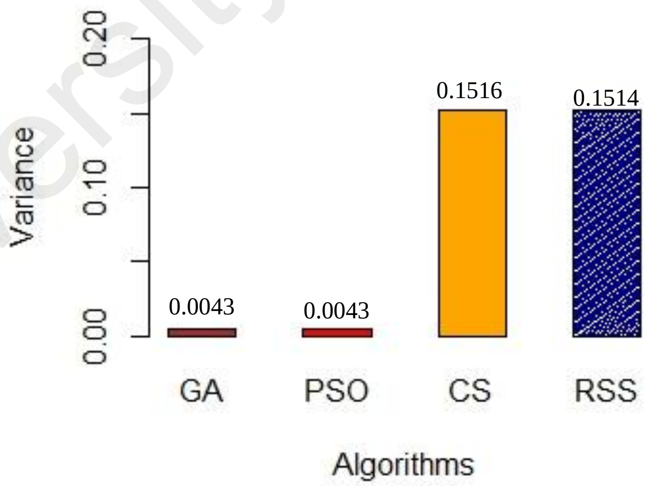


Figure 4.13: Variance results by using F_2

This outcome, demonstrates that RSS shows a high rate of diversity in uni-objective functions compared to other algorithms, and linking this outcome with the results of Table 4.1 can help to interpret very well the RSS results, where a lot of diversity results in a lower AB, SD and MB. It is very important to highlight that there is no loss of diversity reported in this results. However, this outcome can have different interpretation if it is linked with other measurements, where higher or lower values of convergence, AB, SD, MB and SQ can define the maturity of the search. In the following sub-section, the solution quality results are introdued.

4.3.3.3. Solution Quality Evaluation

In this section, the SQ outputs were evaluated in order to check the ability of RSS to achieve better solutions compared to other algorithms.

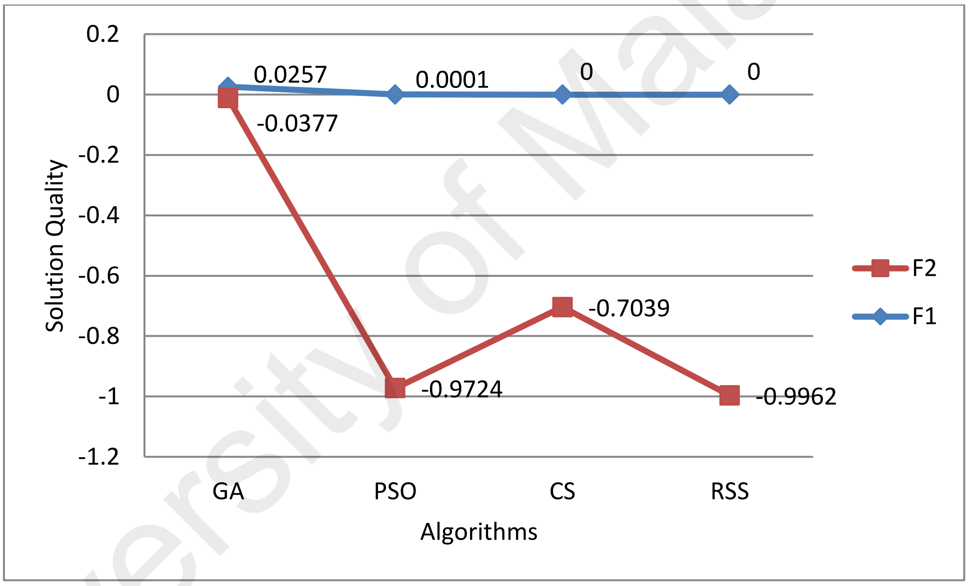


Figure 4.14: Solution quality results for F_1 and F_2

Figure 4.14 shows that RSS in F_1 is better than GA and PSO. However, RSS shows similar performance with CS, where SQ is almost equals the optimal solution $f_* = 0$. This output demonstrated that RSS is able to solve F_1 test function.

In F_2, where the global optimum is $f_* = -1$. The RSS outperforms other algorithms with an achieved SQ = -0.9962. Based on the results obtained in AB, MB, Convergence and SQ, the maturity of RSS can be evaluated compared to other algorithms in the following sub-section.

4.3.3.4. Maturity Evaluation

The maturity of the search is evaluated using the results obtained from the convergence rate and the solution quality. The SQ results were compared with those obtained by AB to check the maturity of the search. The AB and SQ results were plotted to visualize the differences among them.

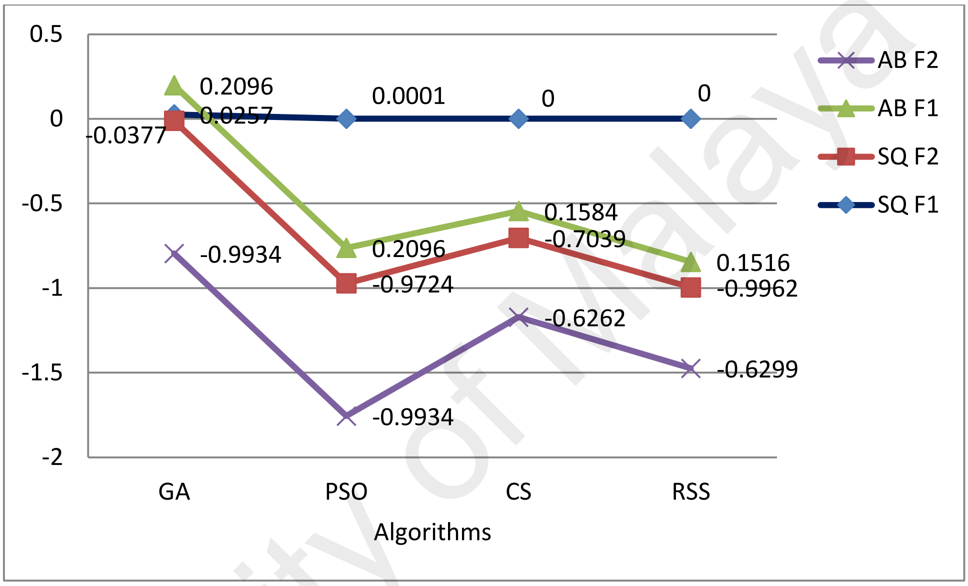


Figure 4.15: Evaluating maturity at F_1 and F_2 using AB and SQ

In Figure 4.15, comparing AB and SQ results for F_1 shows that the fast AB convergence of RSS provides a better solution quality, which is equal to the global optimum $f_* = 0$. This result can be interpreted as a mature convergence for RSS in F_1 (mature search). For F_2, comparing AB and SQ results indicate that RSS shows a slow AB convergence however it ends the search with better solution quality compared to other algorithms. In metaheuristics, such performance indicates that RSS has a mature search, i.e. RSS in F_2 is slow in achieving the global optimum but it has the ability to provide a high solution quality. In contrast, PSO as an example could achieve fast AB convergence; however, it provides a low solution quality compared to RSS.

4.3.4. Multi-Objective Test Functions

In this experiment, multi-objective test functions were used to evaluate the performance of the RSS compared to GA, PSO and CS. A set of thirteen benchmark test functions: $F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}$ and F_{15} were used to test the ability of RSS to deal with multi-objective optimization problems.

Table 4.2: AB, MB and SD results using multi-objective functions

Function ID		GA	PSO	CS	RSS	Optimal Value f_*
F_3	AB	-820.7881	-718.5607	-827.2707	**-831.1258**	$-418.98n$
	MB	-831.4344	-719.5274	-837.6642	**-837.9523**	
	SD	24.5092	117.2421	29.5045	**26.0190**	
F_4	AB	3.8122	0.4933	0.0270	**0.0105**	0
	MB	3.7101	0.0279	0.0123	**0.0028**	
	SD	2.0933	1.5303	0.0360	**0.0259**	
F_5	AB	-1.5819	-1.5933	-1.6026	**-1.6026**	-1.8013
	MB	-1.5906	-1.6026	-1.6026	**-1.6026**	
	SD	0.0228	0.0526	**0**	**0**	
F_6	AB	1.3062	9.8941	0.1493	**0.0159**	0
	MB	1.1959	3.6502	0.0135	**0.0052**	
	SD	0.9546	11.1731	0.3092	**0.0216**	
F_7	AB	-1.7725	-1.7847	-1.8013	**-1.8013**	-4.6877
	MB	-1.7878	-1.8013	-1.8013	**-1.8013**	
	SD	0.0443	0.0928	**0**	**0**	
F_8	AB	-181.0069	-79.1157	-209.2064	**-210.1891**	-186.730
	MB	-185.0235	-27.8028	-210.3642	**-210.4089**	
	SD	26.0941	109.9932	3.7531	**0.8273**	
F_9	AB	0.1736	16.9143	0.0208	**0.0033**	0
	MB	0.0964	0.3611	0.0009	**0.0001**	
	SD	0.1782	72.6749	0.0452	**0.0093**	
F_{10}	AB	0.6182	1.3113	0.0223	**0.0173**	0
	MB	0.6226	1.0656	0.0215	**0.0149**	
	SD	0.3043	1.4230	0.0140	**0.0123**	
F_{11}	AB	0.3577	6.2987	0.0001	**0**	0
	MB	0.1872	0.0001	**0**	**0**	
	SD	0.5382	31.7590	0.0004	**0**	
F_{12}	AB	0.0467	0.0035	**0**	**0**	0
	MB	0.0248	**0**	**0**	**0**	
	SD	0.0527	0.0214	**0**	**0**	
F_{13}	AB	0.7116	10.1861	0.0002	**0**	0
	MB	0.3604	0.0341	**0**	**0**	
	SD	0.8461	39.4115	0.0007	**0.0001**	
F_{14}	AB	-7.0474	-8.5533	-11.0052	**-11.0289**	-186.730
	MB	-5.2529	-11.0297	-11.0288	**-11.0309**	
	SD	2.9045	4.3184	0.0616	**0.0052**	
F_{15}	AB	-177.8809	-95.2011	**-186.6910**	-186.6111	0
	MB	-180.4464	-137.1709	**-186.7237**	-186.6907	
	SD	9.3851	92.5425	0.0987	**0.1759**	

Table 4.2 displays the results of AB, MB and SD for the global optimal results. The results in the table show that RSS received the best AB, MB and SD in most of test functions except F_{15}, where the result will be decorticated in the upcoming sub-sections for maturity and convergence analysis. The superiority of RSS over CS, PSO and GA can be seen in F_3, F_4, F_6, F_8, F_{10}, F_{11}, F_{13} and F_{14} where RSS was able to reach the global optimum during 100 iterations better than other algorithms. For example, in F_6 RSS could reach an AB of 0.0159, in contrast, CS only achieved 0.1493, PSO achieved 9.8941 and GA achieved 1.3062. In table 4.3, the variance and the solution quality were measured for this category of functions in order to get more evidences about the outperformance of RSS.

Table 4.3: Var results using multi-objective functions

Function ID	GA	PSO	CS	RSS
F_3	600.7000	13745.71	870.5155	**<u>676.9883</u>**
F_4	4.3819	2.3418	0.0012	**<u>0.0006</u>**
F_5	0.0005	0.0028	**<u>0</u>**	**<u>0</u>**
F_6	0.9112	124.8382	0.0956	**<u>0.0005</u>**
F_7	0.0020	0.0086	0	**<u>0</u>**
F_8	0.0681	1.2099	0.0014	**<u>0.0001</u>**
F_9	0	5.2816	**<u>0</u>**	**<u>0</u>**
F_{10}	0.0926	2.0248	0.0002	**<u>0.0002</u>**
F_{11}	0.0003	1.0086	**<u>0</u>**	**<u>0</u>**
F_{12}	0.0028	0.0005	**<u>0</u>**	**<u>0</u>**
F_{13}	0.0007	1.5533	**<u>0</u>**	**<u>0</u>**
F_{14}	8.4359	18.6484	0.0038	**<u>0</u>**
F_{15}	0.0881	8.5641	**<u>0</u>**	**<u>0</u>**

Table 4.3 shows the variance values of the test functions. These results demonstrate the superiority of RSS compared to GA, PSO and CS. This can be justified by the diversification and the optimal balance between exploration and exploitation by RSS during the search for the global optimum. Moreover, this findings indicates that the RSS has an optimal diversity even though in some test functions the RSS achieved a low variance; however it shows better solution quality.

The balance between exploitation and exploration during the search is achieved through the sensitive search model, where the search can take two different states: urgent or normal. It is very important to highlight the importance of the obtained results in term of measuring the ability of RSS to escape poor local optima traps and its ability to locate a near-global optimum.

Tables 4.2 and Table 4.3 indicate that RSS, CS, PSO and GA were able to find the global optimum during 100 iterations. However, RSS shows an outperformance in terms of the AB solutions in most of the test functions. Furthermore, the measurement of the SD, MB and Var clarified how RSS search is spread out. In order to evaluate the diversification of the search, the SD and Var results were linked with the AB and SQ results. In the following section, the RSS algorithm is explained and how it converged and consumed less number of iterations to achieve the global optimum.

4.3.4.1. Convergence Analysis

The convergence of the GA, PSO, CS and RSS is illustrated in order to evaluate the capability of RSS to find the global minimum within a reasonable time compared to other algorithms. The convergence is measured by using the number of required steps to achieve the global minimum.

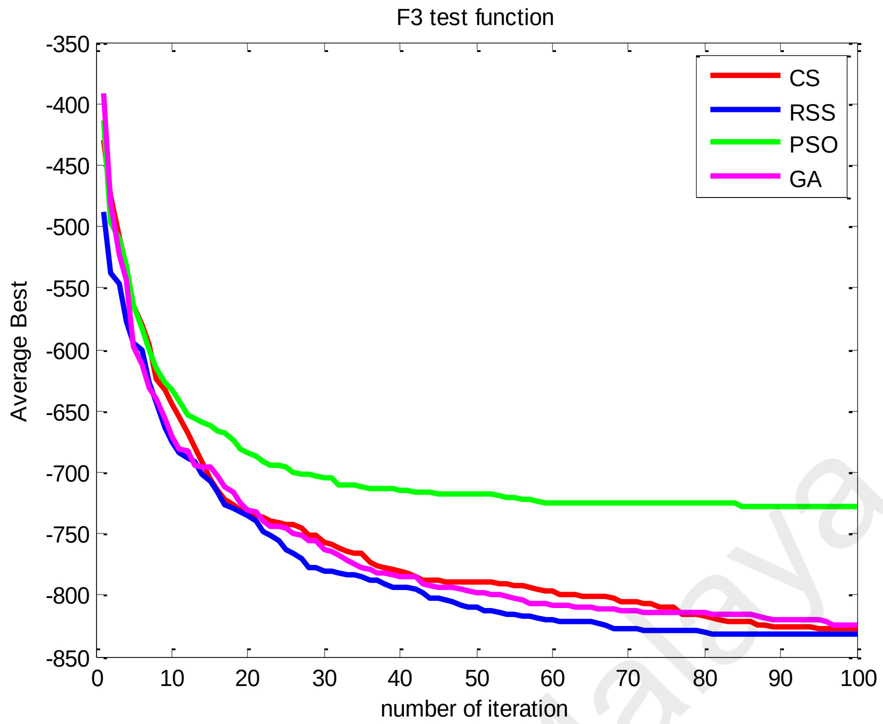


Figure 4.16: Average best convergence of F_3

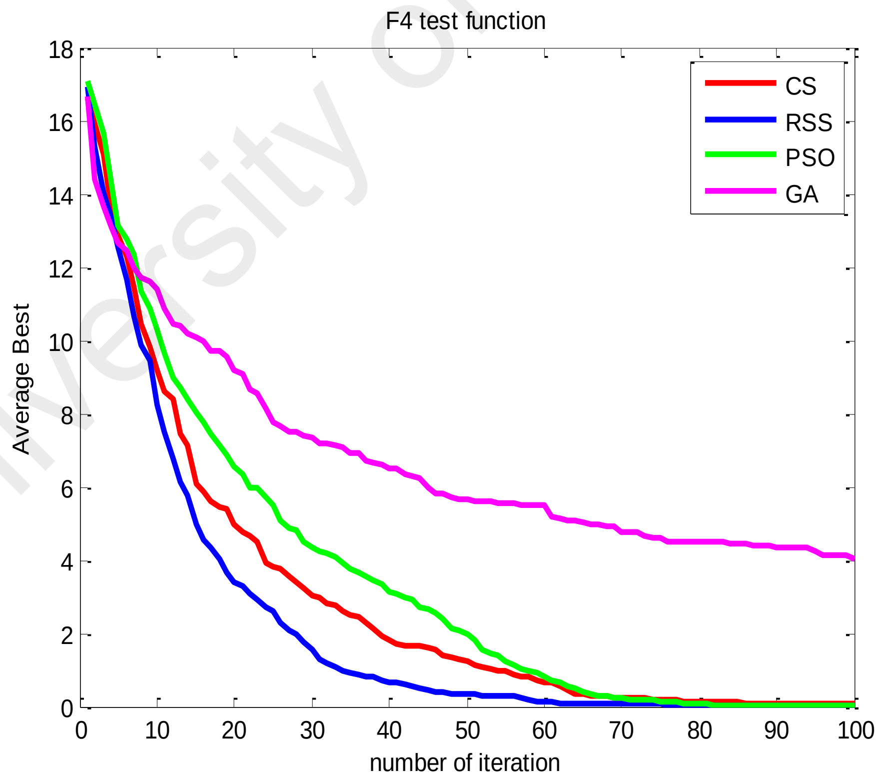


Figure 4.17: Average best convergence of F_4

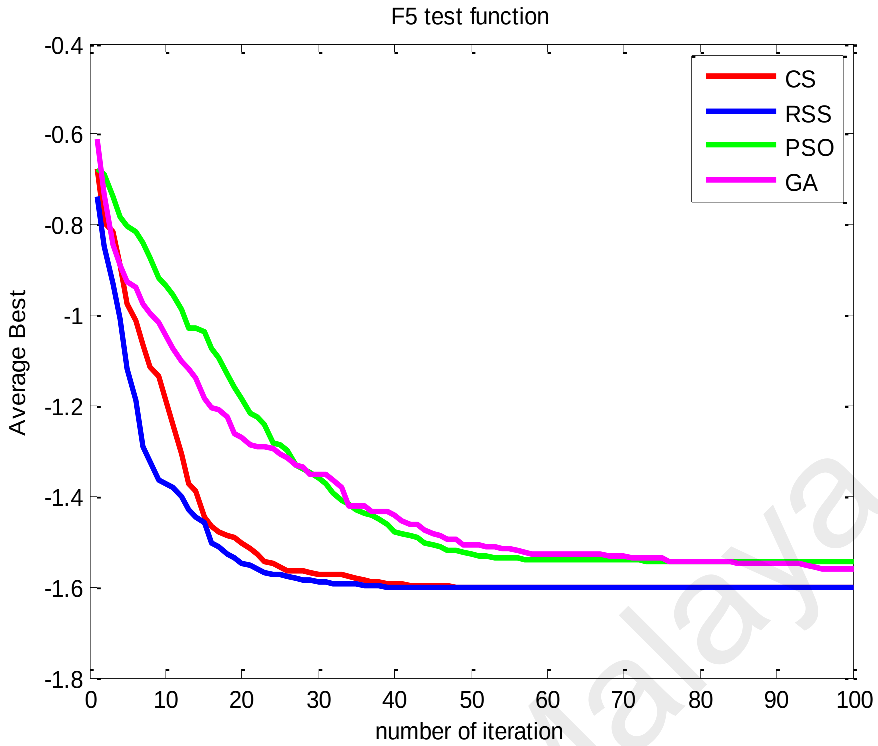


Figure 4.18: Average best convergence of F_5

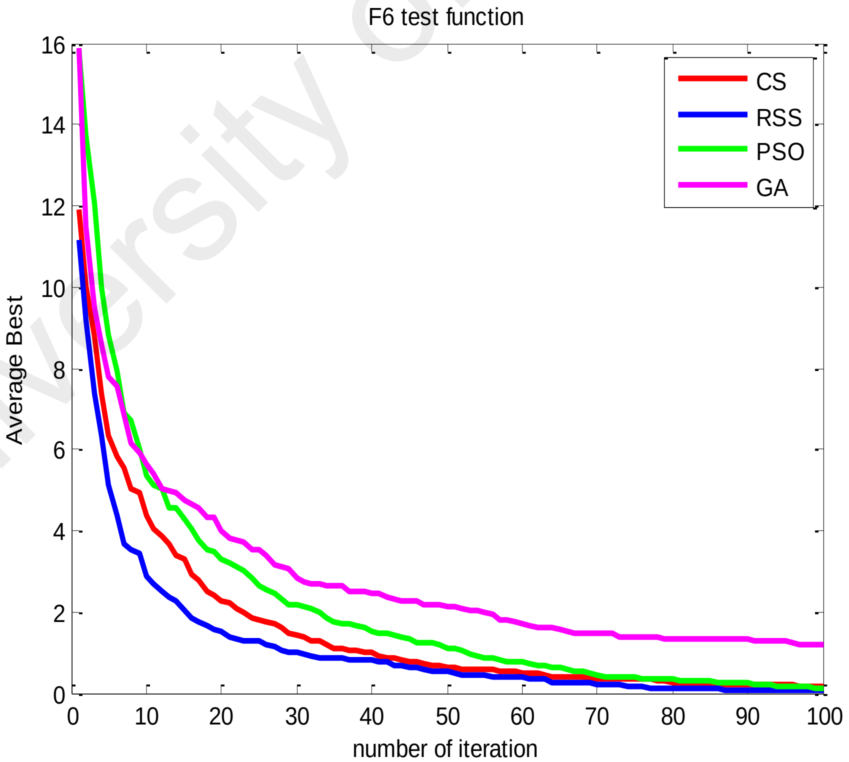


Figure 4.19: Average best convergence of F_6

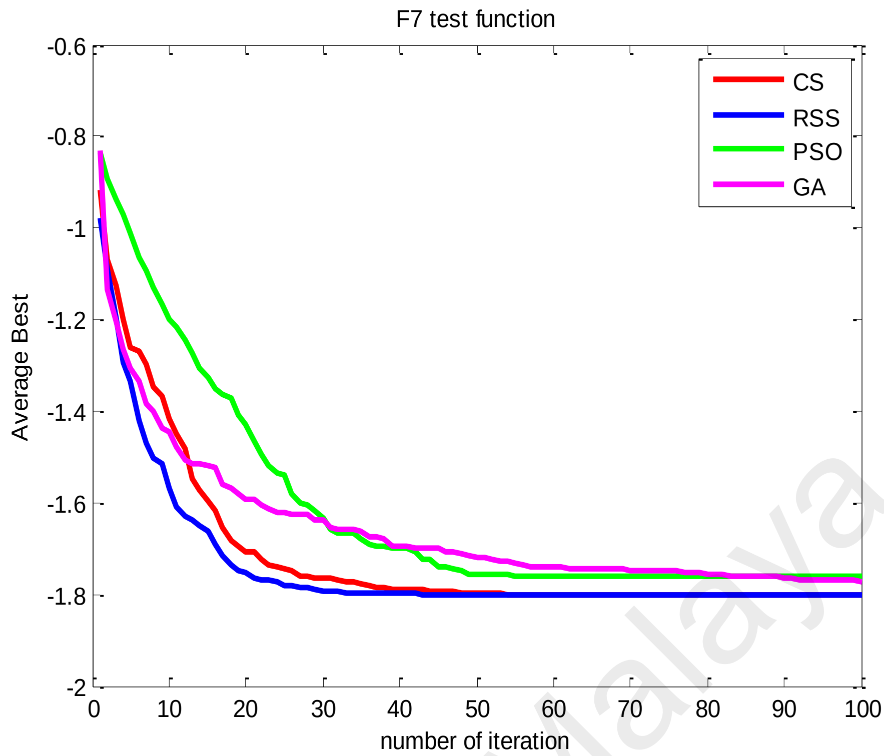


Figure 4.20: Average best convergence of F_7

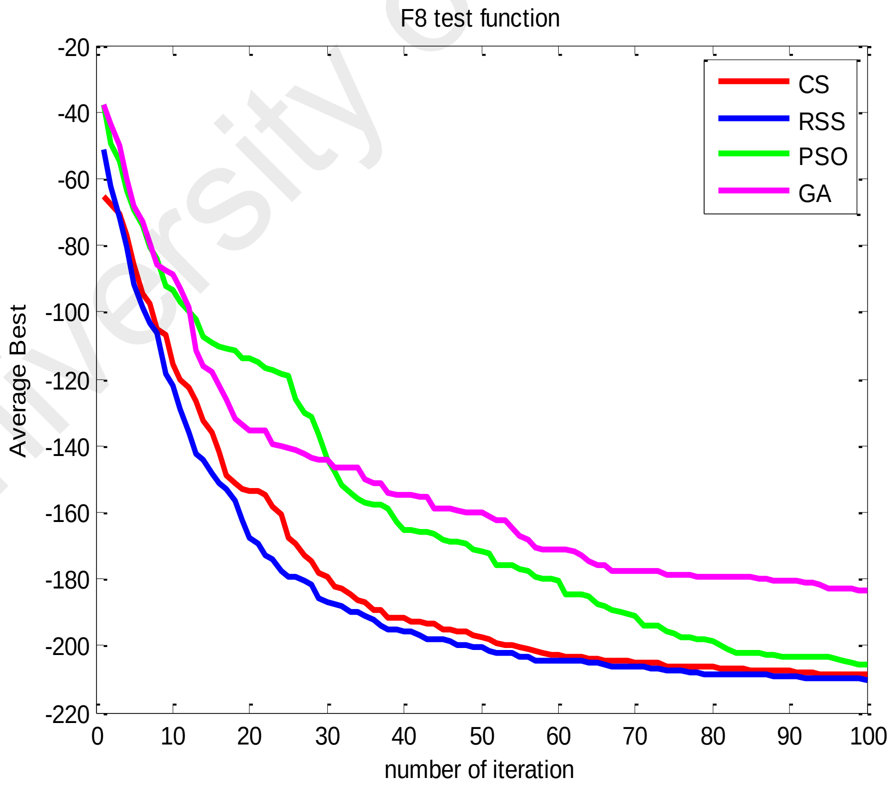


Figure 4.21: Average best convergence of F_8

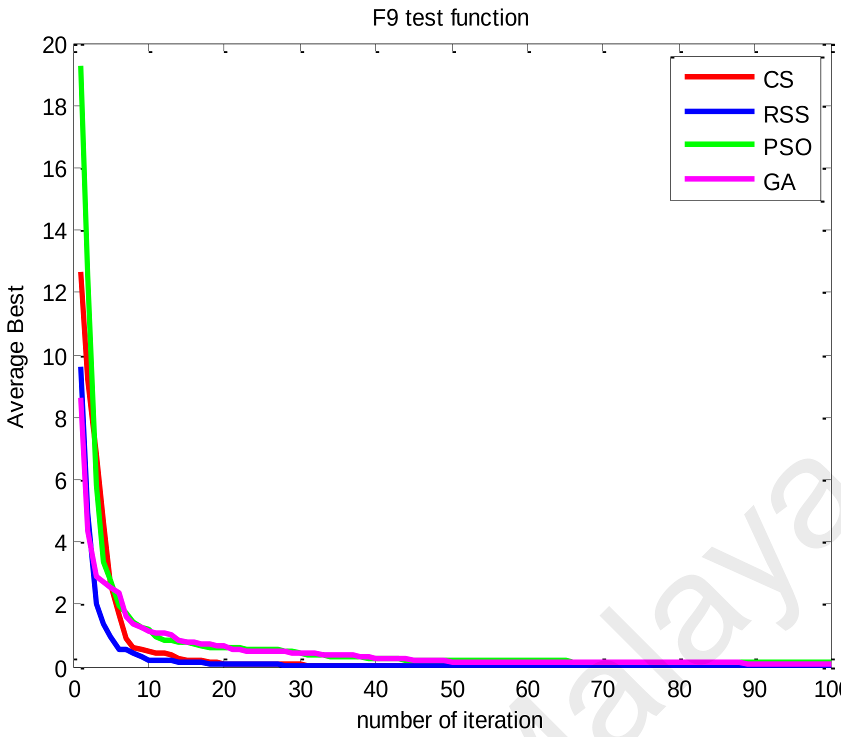


Figure 4.22: Average best convergence of F_9

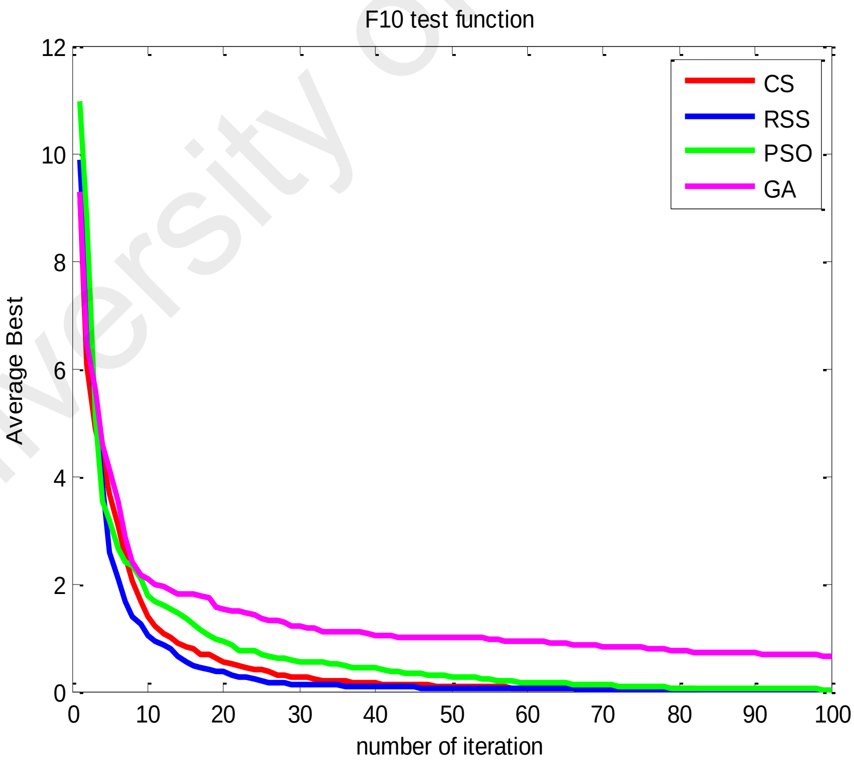


Figure 4.23: Average best convergence of F_{10}

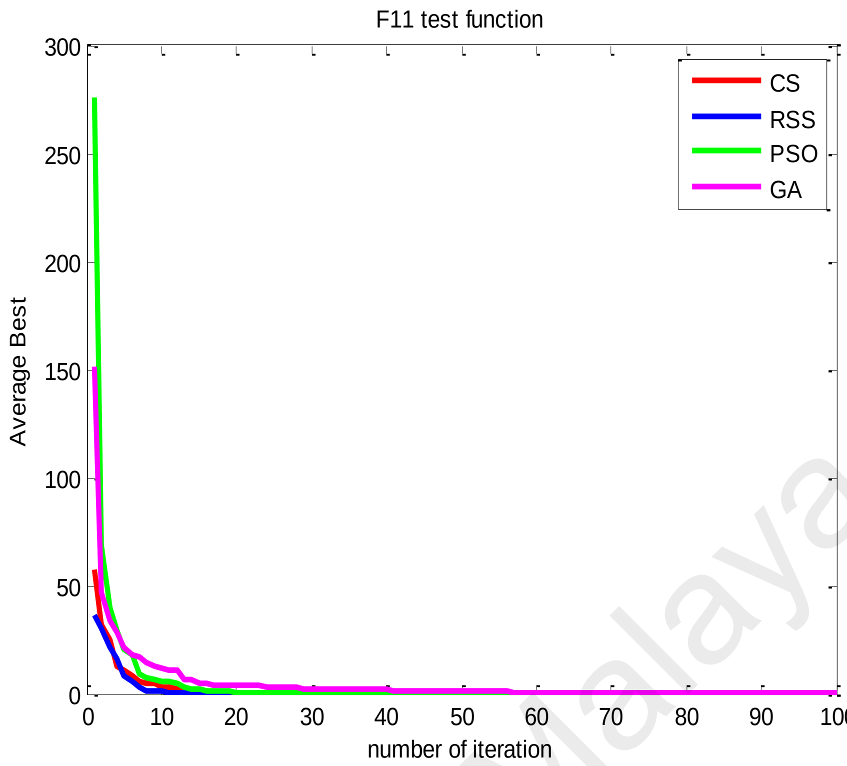


Figure 4.24: Average best convergence of F_{11}

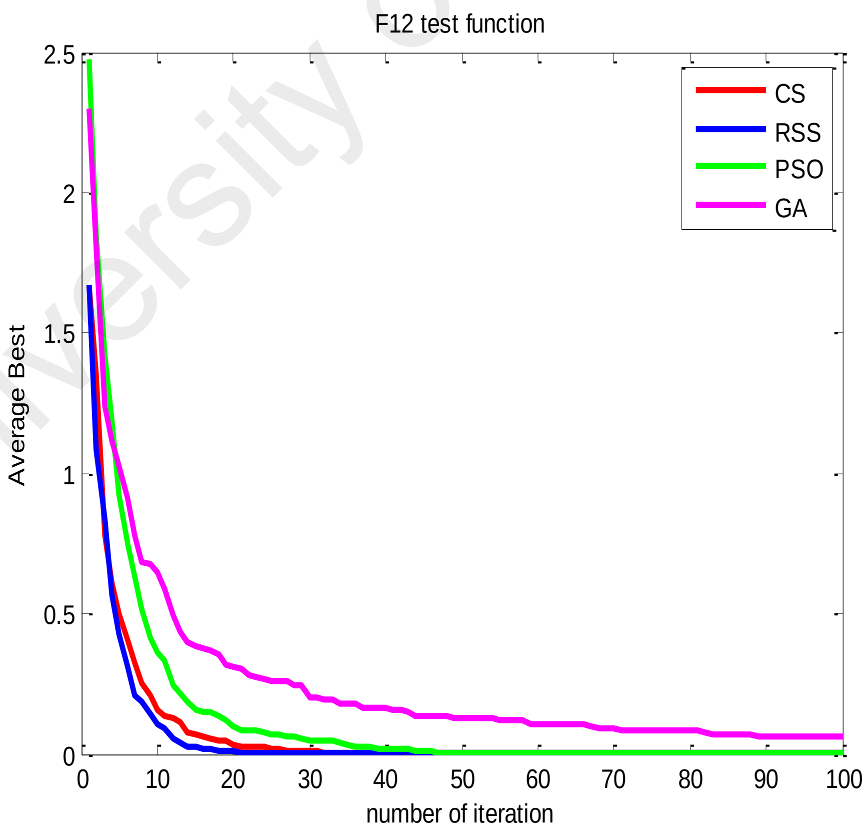


Figure 4.25: Average best convergence of F_{12}

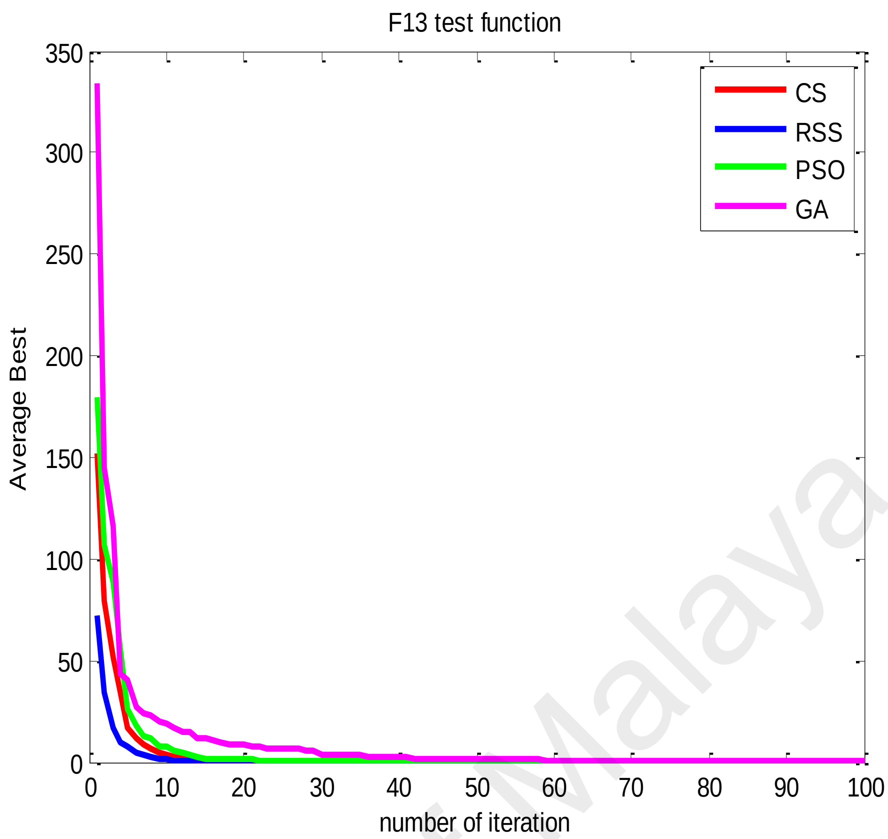


Figure 4.26: Average best convergence of F_{13}

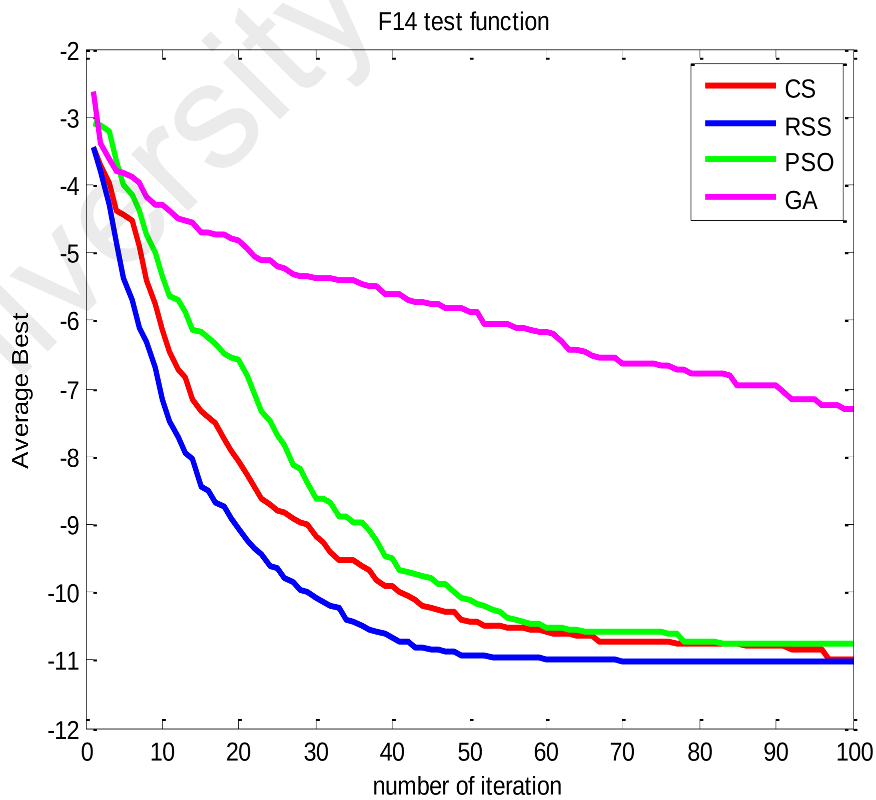


Figure 4.27: Average best convergence of F_{14}

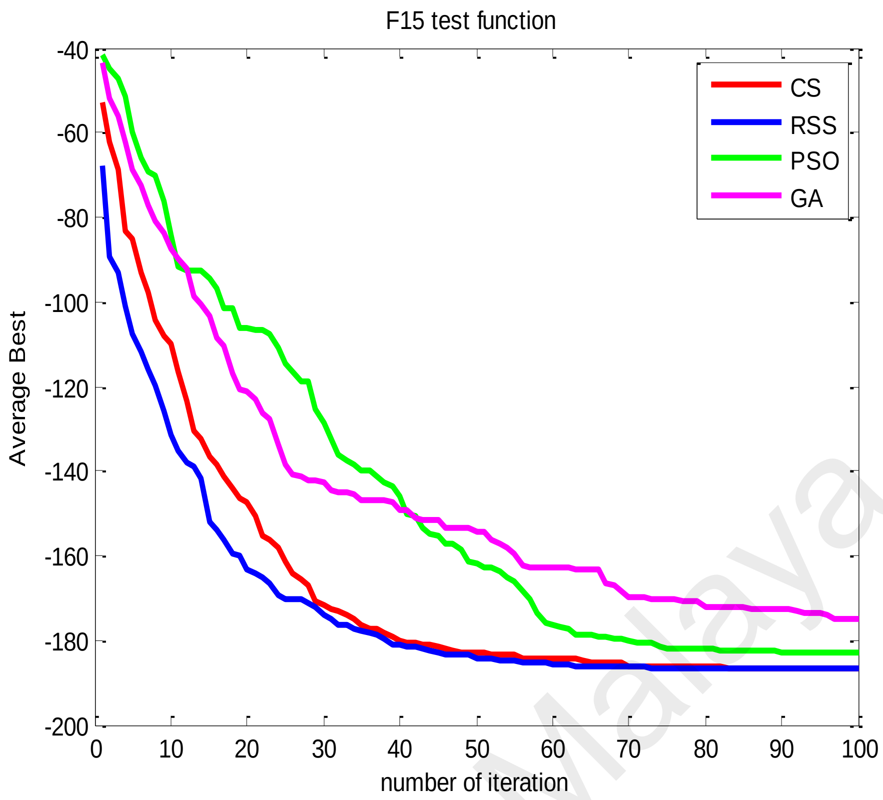


Figure 4.28: Average best convergence of F_{15}

Figure 4.16 to Figure 4.28 describe the convergence rates based on the average best outputs for GA, PSO, CS and RSS algorithms considering the functions F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{14} and F_{15}. Thus, the smaller the number of required steps, the higher the convergence speed. The results show that RSS consumes less time to reach the global optimum. This is a proof that RSS outperforms GS, PSO and CS in terms of convergence to the global optima. The evidence shows how RSS quickly converged to the global optimum compared to other algorithms. During little iteration RSS can have the ability to reach the global optimum. This can be justified by the optimal exploitation-exploration based on the sensitive search model inspired from seal movement. It is worth noting that, faster convergence does not necessarily mean an optimal output. In fact, too fast convergence may lead to the problem of prematureness, which leads the search to be trapped at local optima positions (Črepinšek et al., 2013; Gao et al., 2015; Yang, Cui, Xiao, Gandomi, & Karamanoglu, 2013). As shown in

Table 4.2, Table 4.3 and Table 4.4, the SQ values demonstrate that the final achieved positions are equal or quite near the optimal values. This is a proof that the RSS mechanism escaped local optima traps.

4.3.4.2. Diversity Analysis

In this analysis, the spread out of the search of RSS is described using variance outputs and it is compared with GA, PSO and CS. To uncover the underlying mechanism of the proposed algorithm, the optimization process is examined in terms of variance point of view. For the sake of simplicity, in the following, the results for the function F_{15} were presented.

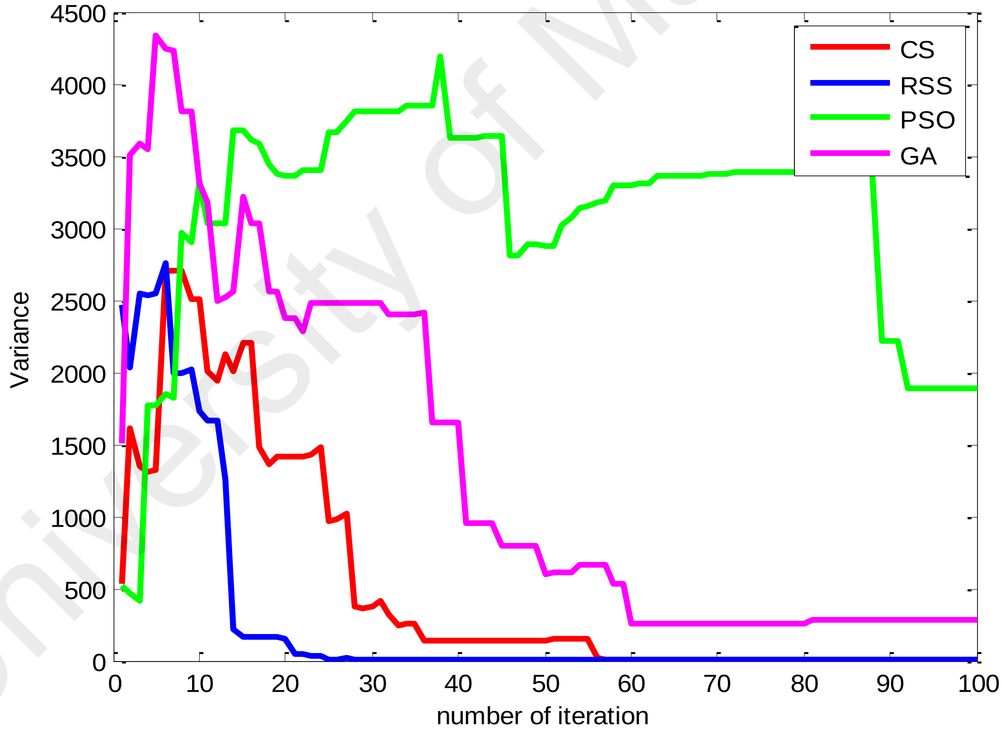


Figure 4.29: The variance during different iterations of GA, PSO, CS and RSS

From Figure 4.29, it is noticed that the variance results confirm the convergence results of RSS obtained in Section 4.3.4.1.The RSS variance converges to 0 before the 30th iteration, because RSS required more exploitation and less exploration. This indicates that the RSS could find the optimal value quickly which can be interpreted as a fast convergence. Matching the convergence result with the variance result indicates that the RSS has an optimal diversity. Moreover, this result is related to the fact that RSS search is based on two search modes: normal state and the urgent state, where exploitation and exploration are balanced by using the sensitive search model. Variance can varies between high and low values depends on the state of the search.

4.3.4.3. Solution Quality Evaluation

The solution quality of RSS is evaluated and compared with GA, PSO and CS in order to check the ability of RSS to reach better solutions quite near the global optimal minimum result.

Table 4.4: SQ results using multi-objective functions

Function ID	GA	PSO	CS	RSS	Optimal Value f_*
F_3	-812.2354	-765.3373	-831.8597	**-832.6857**	$-418.98n$
F_4	3.9664	0.7778	0.0171	**0.0072**	0
F_5	-1.5317	-1.5209	**-1.6026**	**-1.6026**	-1.8013
F_6	1.1525	4.3005	0.1450	**0.0174**	0
F_7	-1.7737	-1.7181	**-1.8013**	**-1.8013**	-4.6877
F_8	-167.0768	-88.2854	**-210.2348**	-209.8650	-186.730
F_9	0.1694	106.0209	0.0218	**0.0052**	0
F_{10}	0.6132	0.7230	0.0185	**0.0169**	0
F_{11}	0.3733	0.2911	0.0002	**0**	0
F_{12}	0.0591	**0**	**0**	**0**	0
F_{13}	0.2663	5.0913	0.0001	**0**	0
F_{14}	-7.5363	-8.2977	-10.3955	**-11.0303**	-186.730
F_{15}	-176.5184	-117.7560	-186.6132	**-186.6440**	0

From Table 4.4, the proposed algorithm RSS shows better SQ results in most of the test benchmark functions, where the achieved solution quality results were quite similar to the optimal minimal solution f_* of each test function. It is very important to highlight that the results show that SQ of some test functions are quite similar, which is the case for both RSS and CS. The similarity in terms of SQ in some functions might be related to the fact that both algorithms RSS and CS used Levy for exploration.

4.3.4.4. Maturity Evaluation

In order to measure the maturity of the proposed algorithm RSS, the achieved solution quality SQ is compared for each algorithm and the AB convergence rate. If an algorithm achieves better convergence but its SQ is less than other algorithm, the algorithm can be considered immature. In case of an algorithm, outperforming other algorithms in terms of solution quality, then the algorithm is considered mature, regardless the convergence. The maturity comparison is illustrated in the following plots.

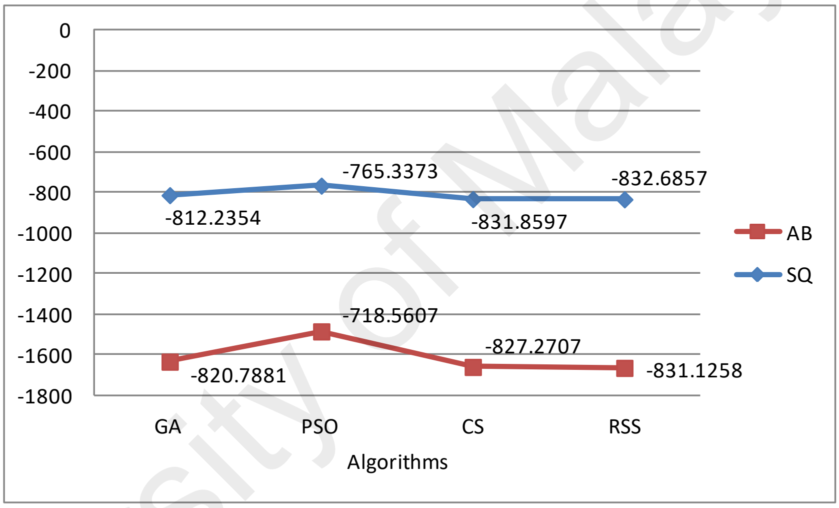


Figure 4.30: Evaluating maturity at F_3 using AB and SQ

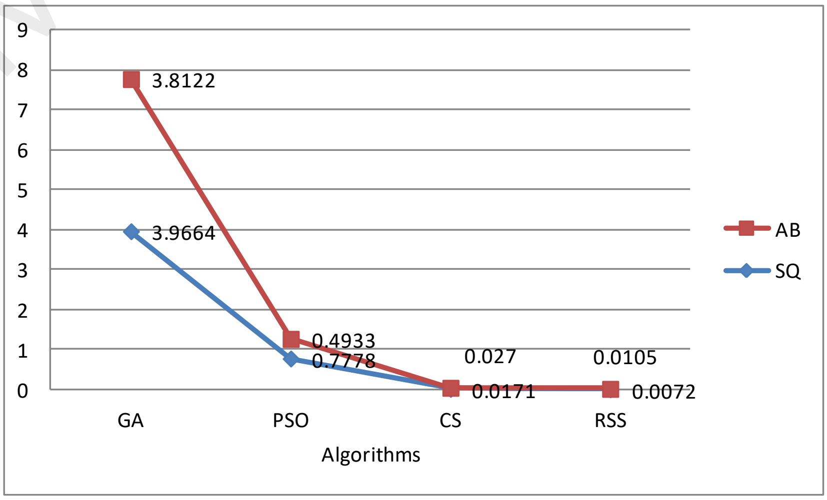


Figure 4.31: Evaluating maturity at F_4 using AB and SQ

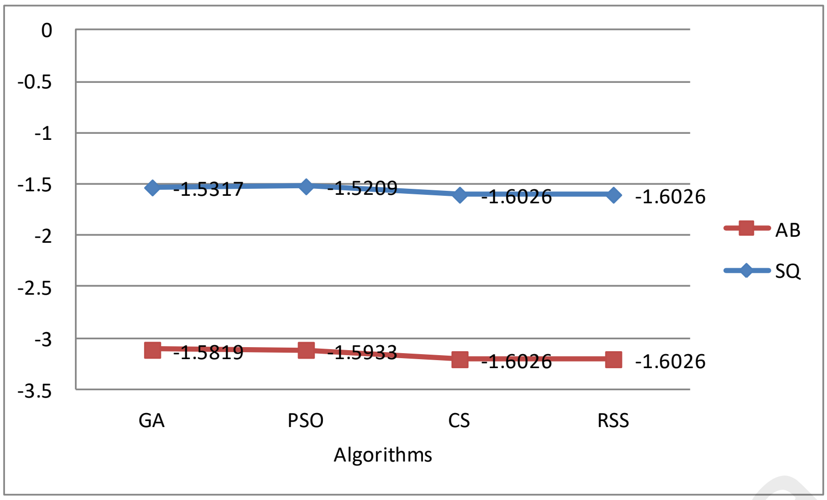


Figure 4.32: Evaluating maturity at F_5 using AB and SQ

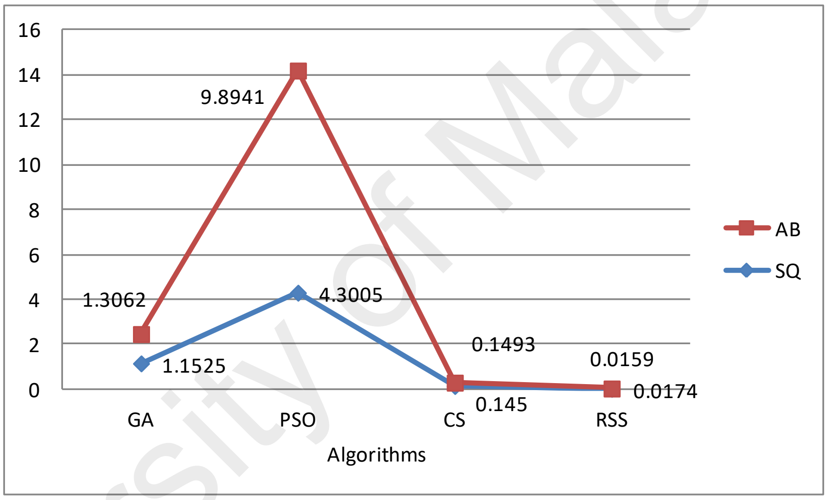


Figure 4.33: Evaluating maturity at F_6 using AB and SQ

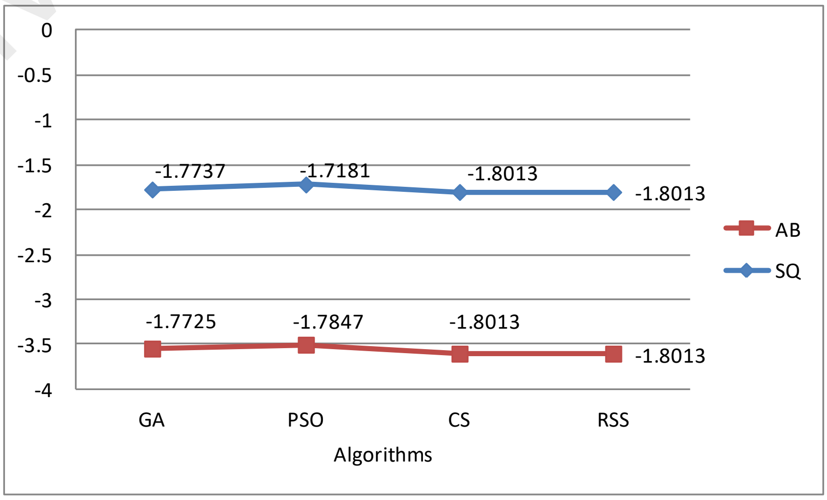


Figure 4.34: Evaluating maturity at F_7 using AB and SQ

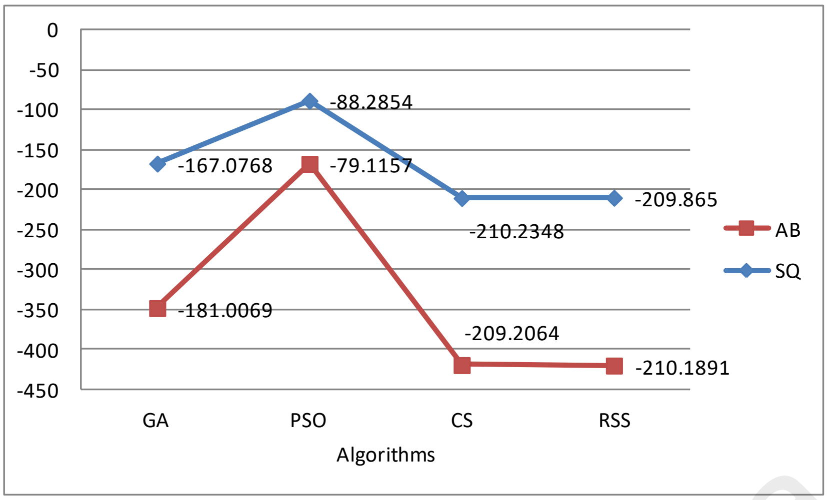


Figure 4.35: Evaluating maturity at F_8 using AB and SQ

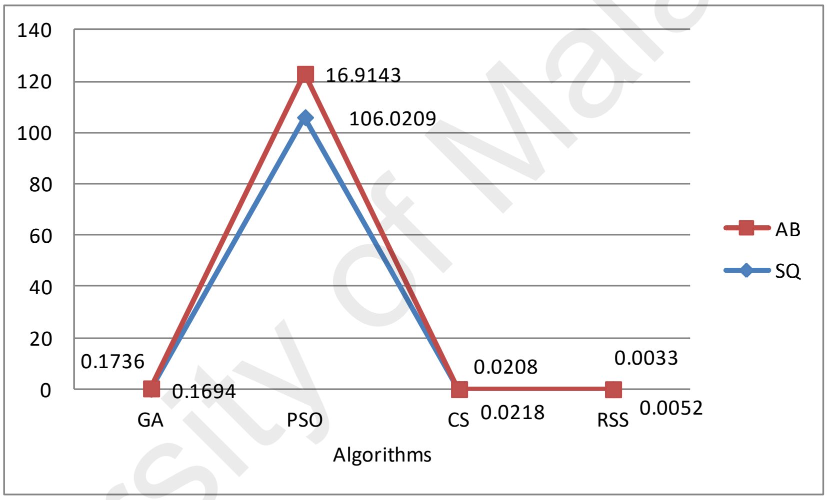


Figure 4.36: Evaluating maturity at F_9 using AB and SQ

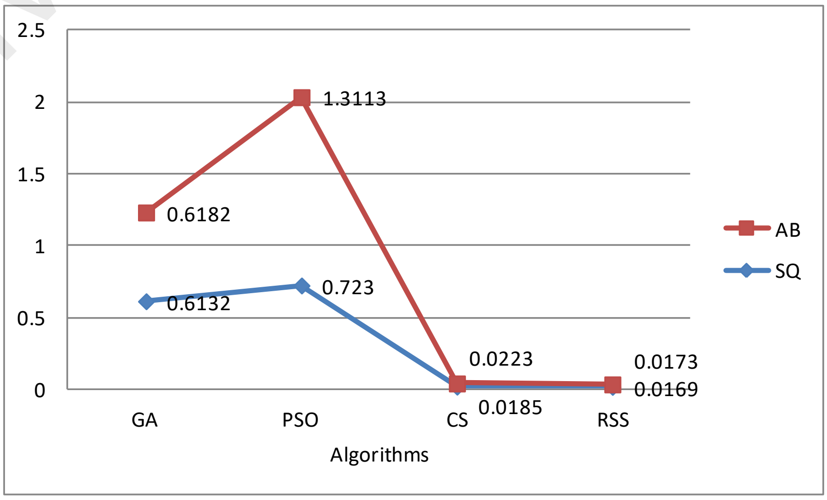


Figure 4.37: Evaluating maturity at F_{10} using AB and SQ

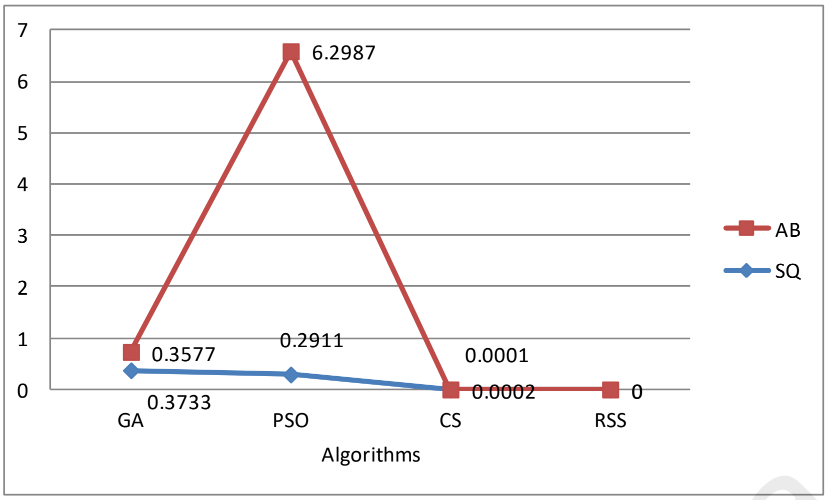


Figure 4.38: Evaluating maturity at F_{11} using AB and SQ

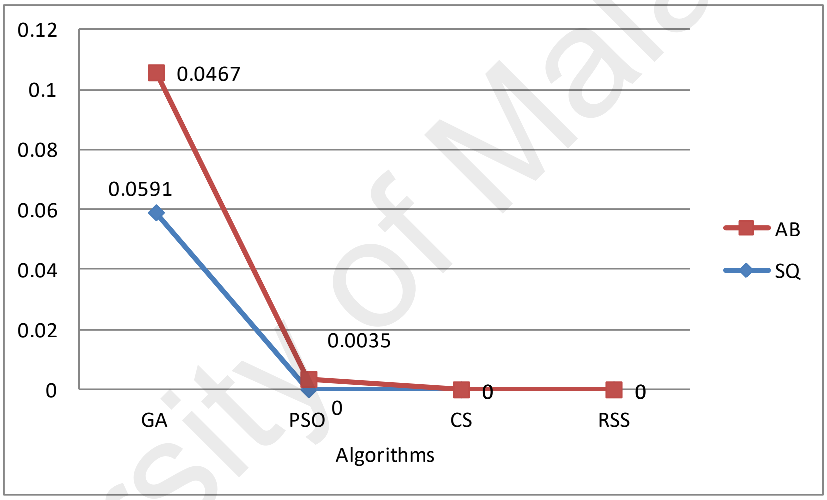


Figure 4.39: Evaluating maturity at F_{12} using AB and SQ

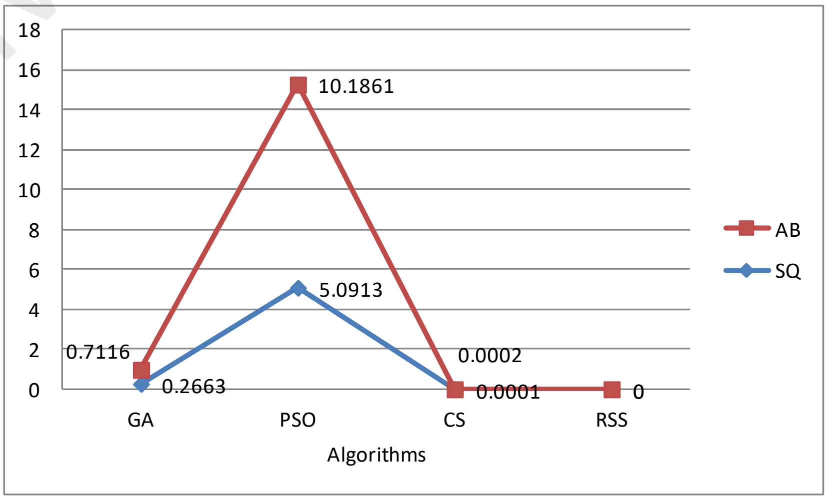


Figure 4.40: Evaluating maturity at F_{13} using AB and SQ

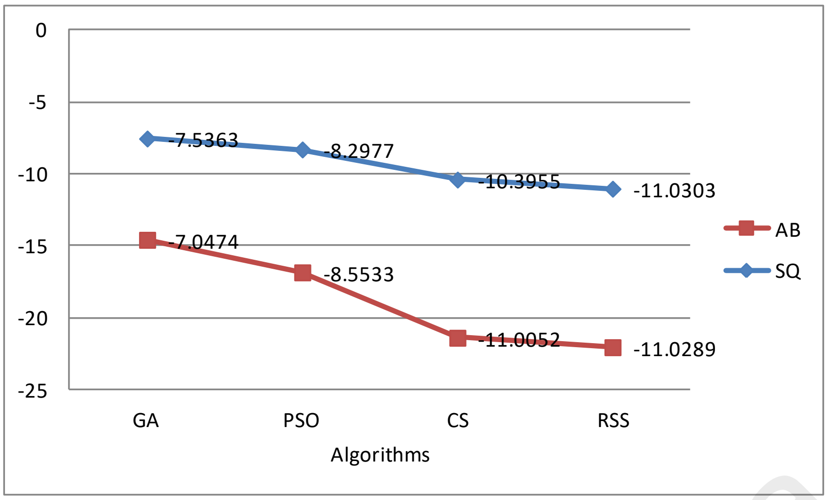


Figure 4.41: Evaluating maturity at F_{14} using AB and SQ

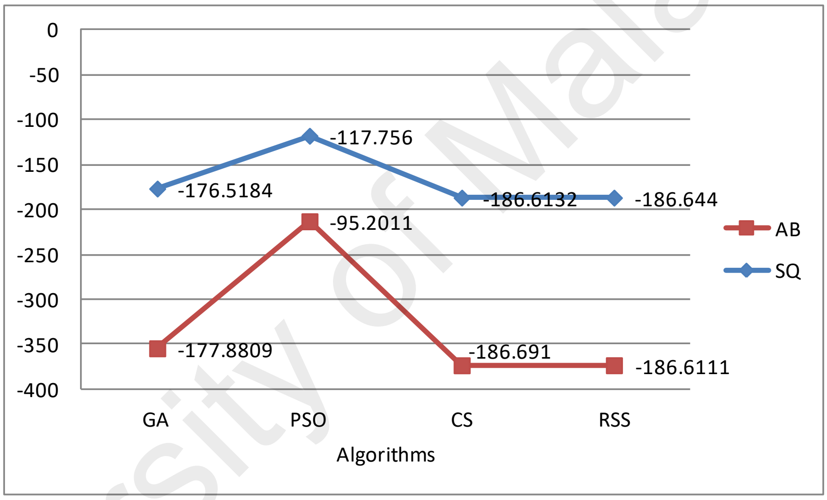


Figure 4.42: Evaluating maturity at F_{15} using AB and SQ

From Figures 4.30 to 4.42, it is noticed that RSS shows a mature search in most of the test functions compared to other algorithms. For example, in F_3 the fast AB convergence of RSS is compatible with the obtained SQ of the RSS, which indicates that the search for the global minimum was mature. The same observation is noticed for F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13} and F_{14} where the fast convergence of the search provided the best solution quality. This can be justified by the fact that RSS which is based on Levy and Brownian has the ability to explore and exploit the search space optimally. It is noticed also that in F_{15} (Figure 4.42), RSS shows a slow convergence compared to CS; however, the achieved solution quality were better compared to CS and other algorithms as well. This indicated that, even though the search of RSS is slow but at the end, it is able to deliver better solution quality.

4.4. Summary

This chapter reports the results of the proposed algorithm Ringed Seal Search (RSS) based on the Sensitive Search Model that aims to solve global optimization problems. The Sensitive Search Model combines Levy walk and Brownian walk to adaptively balance exploitation and exploration.

The experiments were divided into two parts: in the first part, only uni-objective test functions were used and in the second part, multi-objective test functions were used. Five well known evaluation metrics including Time consumption, Convergence, Diversity, Solution quality and Maturity are selected to show the high quality of the proposed algorithm. Three baseline algorithms GA, PSO and CS are used for comparison, where the metrics are calculated on selected number of iteration fixed at 100 and the parameter setting of the algorithms are tuned up as practice in the literature.

In the first part, uni-objective test functions were used to test the proposed algorithm. Evaluations were conducted on two uni-objective test functions show that the RSS achieves much better results compared to its homologs. In the second part, a set of thirteen multi-objective test functions were used. Evaluations on several test functions using the evaluation metrics show the RSS outperforms its competitive algorithms.

The last part was devoted to show the improvement of the proposed algorithm compared to GA, PSO and CS. It is found that the RSS which is built on the sensitive search model via Levy walk and Brownian, not only able to solve global optimization problems and adaptively balance exploitation and exploration, but also delivers better improvement compared to its homologs.

CHAPTER 5: A CASE STUDY – RSS FOR DATA CLUSTERING

5.1. Introduction

The purpose of this case study is to validate the proposed RSS algorithm. This is a part of the research objectives stated in Chapter 1. It is very important to show that RSS is able to solve real optimization problems such as data clustering, which is considered as an unsupervised learning technique used in various domains. Several kinds of approaches to clustering were introduced in the past twenty years since the introduction of the k-means approach. Since then, optimization algorithms have been applied to data clustering problems to find optimal cluster groups via various objective functions. This chapter introduces a new clustering algorithm based on the RSS, which is characterized by a fast convergence to the global optimum. Accordingly, this feature is utilized in the proposed algorithm to find cluster centres for data points. This is done by placing each object in its respective cluster centre using the Euclidean distance measure. A total of seven benchmark datasets were used to test and calculate the accuracy and internal and external indexes. The experimental results were tabulated and analysed using the benchmark datasets. Finally, under the given set of parameters, the proposed RSS-based clustering algorithm can be used for data clustering.

5.2. Formulation of the Data Clustering Optimization Problem

The aim of the proposed formulations is to express the real objective of the clustering in the objective function, which calculates the similarity between objects of the dataset. From this perspective, let D be a dataset represented by

$$D_{m\times n} = \left[D_{1\times n}, D_{2\times n}, \cdots, D_{m\times n}\right], \tag{5.1}$$

where m denotes the dataset objects and n represents the ensemble of features. A dataset object can feature any number of dimensions, which are called attributes or features. In general, the task of clustering (it is denoted as a function f) is to find a way in which, based on the similarity measurement, a sample of points can be grouped into a specified cluster, $\kappa \in (\beta \subseteq \{D_1, D_2, \cdots, D_n\})$, where β is the output space. The task is to compute a function $(f \in \beta) : x \rightarrow y$, where β is the function space. A function f is determined according to the nature of the problem, so f can cluster a sample of data points (x, y).

$$(x_1, y_1), \cdots, (x_m, y_m) \tag{5.2}$$

The partition approach consists of clustering the dataset D into κ clusters, where $\kappa \leq m$.

$$C_\kappa \neq \phi, \ \ \kappa = 1,2,\cdots,\kappa , \tag{5.3}$$

where C_κ represents the centre of the κ^{th} cluster and $k = 1,2,\cdots,\kappa$ is the number of clusters provided. Then, the intersection between the κ^{th} cluster and another cluster group is viewed as an empty group.

$$\{C : C \in \kappa\} \wedge \{C' : C' \in \kappa'\} \Rightarrow C_\kappa \cap C_{\kappa'} = \phi , \ \kappa \neq \kappa' . \tag{5.4}$$

Hence, gathering the ensemble of cluster groups produces all the elements of the dataset, given by

$$\sum_{\kappa=1}^{\kappa} C_\kappa = D . \tag{5.5}$$

The process of creating cluster centres is dependent on the similarity measure between the dataset objects. Let f be a fitness function. The grouping of the dataset objects can be considered as an optimization problem by minimizing the following function:

$$\underset{C_\kappa}{Min}\left[f\left(D_{m\times n},C_\kappa\right)\right]. \tag{5.6}$$

In this study, the adaptation is carried out by optimizing (minimizing) the sum of all set instances of the Euclidean distance. It is shown that the Euclidean distance is easy to compute and performs well with datasets with compact or isolated clusters (Cha, 2007). It is considered as the objective function and can be represented by

$$Dist\left(m_i, m_j\right)=\left(\sum_{k=1}^{n}\left|m_{ik}-m_{jk}\right|^2\right)^{\frac{1}{2}}. \tag{5.7}$$

Here, m represents the objects, n is the number of attributes, and m_{ik} denotes the value of the k^{th} attribute of the object i. The process of finding cluster centres can be formulated as an optimization problem, where the objective function is defined by Equation 5.7.

5.3. Proposed Data Clustering Approach

The clustering techniques group the objects into classes or clusters, which are formed based on a particular algorithm. The datasets that were considered contain numerical information on classes for each dataset. In this study, RSS is proposed to build a new clustering approach. The RSS algorithm were used to compute the optimal solution of the clustering objective function expressed in Equation 5.7, where the data points are represented by lairs. The RSS search converges towards these points, finally forming the centres of the clusters. The proposed approach iterates until the stopping criterion is met. The flow chart of the proposed algorithm is given below.

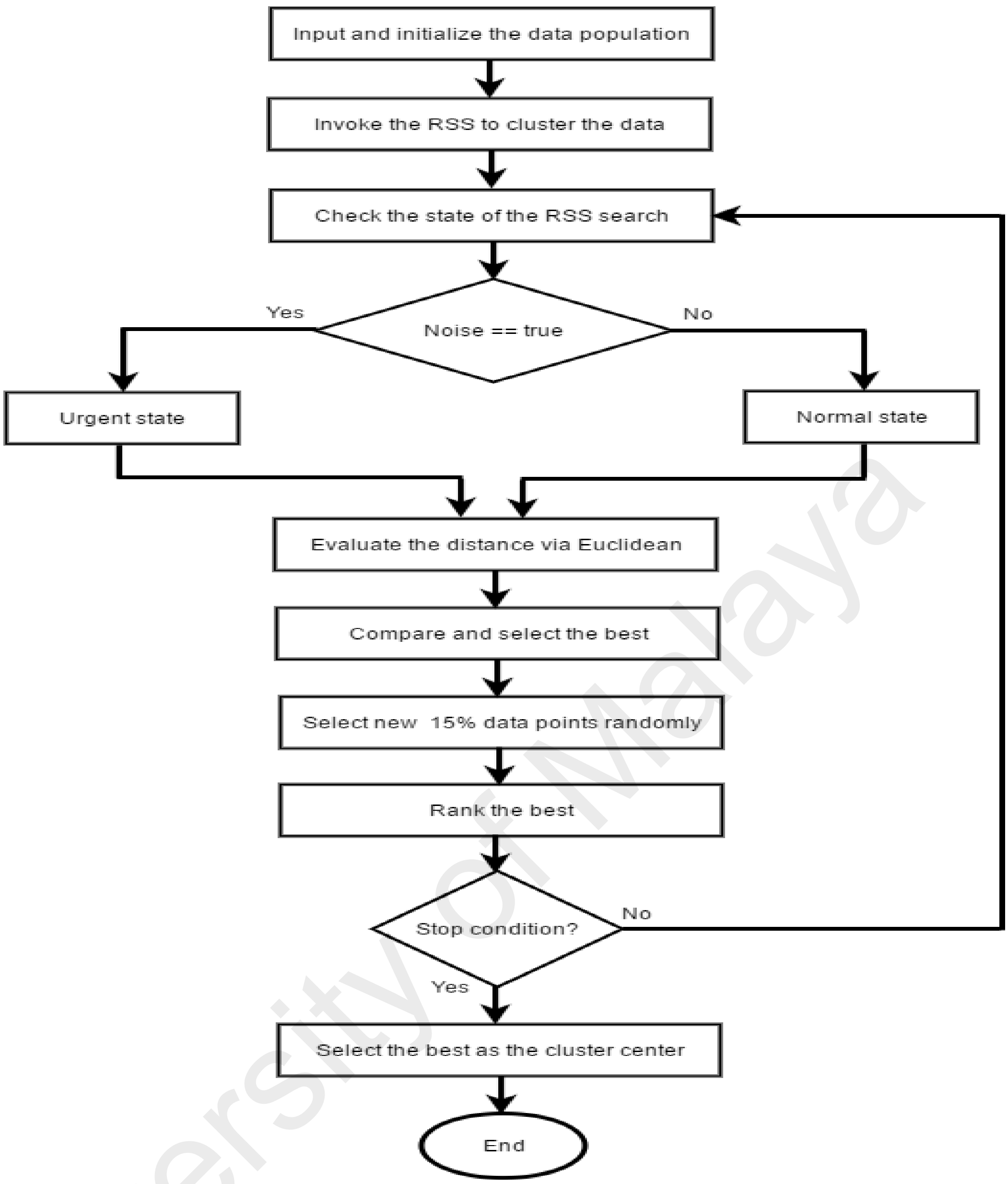


Figure 5.1: Computing the centroids by using the RSS

Figure 5.1 illustrates the centroid computational process using RSS. In the same way as clustering algorithms, the proposed RSS-based clustering begins with inputting and initializing a data population set. Then, the RSS is invoked iteratively to search for the best centre of the data. The RSS search begins by checking the state of the search, where the search state can be normal or urgent. In both search states, a new data point can be randomly selected and evaluated using the defined objective function (Euclidean). In order to rank the best data points, a comparison is required between the selected data point and the ensemble of the data points in terms of the Euclidean distance. The process repeats until the stopping criterion (number of iterations) is met. The location of the best data point is considered as the cluster centre. The RSS aims to create a set of clusters, as illustrated in Figure 5.2. The search for solutions is started by initializing a set of lairs, represented by

$$L_i, i = (1,2,\cdots,n) \tag{5.8}$$

The lairs are distributed randomly, and each lair l contains many chambers m. For example, a lair i is an array of $[1 \times m]$, representing the current existing lair l of a habitat.

$$L = [1 \times m]. \tag{5.9}$$

Such values are randomly and uniformly distributed in the search space between the pre-defined lower bound Lb_j and the upper bound Ub_j, as illustrated by the following expression:

$$L_i = Lb + (Ub - Lb).rand\left(size\left(Lb\right)\right), \text{ where } i = (1,2,\cdots,n), \tag{5.10}$$

where i represents the number of the lair and n indicates the number of initialized lairs.

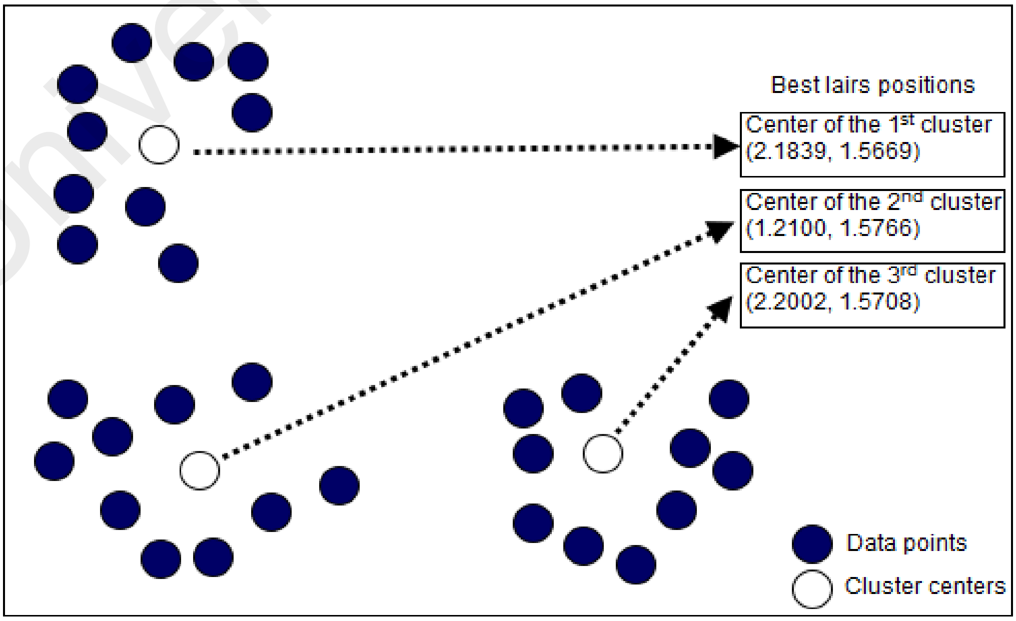


Figure 5.2: The encoding of a clustering problem in RSS

In Figure 5.2, the ability of RSS to find the global optimum is utilized as a data clustering approach. Each group of lairs is defined by κ centroid vectors. It is possible to encode a group of lairs for a two-dimensional problem as shown in Figure 5.2, where the cluster centres are represented by the coordinates (2.1839, 1.5669), (1.2100, 1.5766), and (2.2002, 1.5708).

The movement from one lair to a new lair requires a specific search pattern. During the generation of new solutions (new lairs) $x^{(g+1)}$ for, say, a seal i, a new lair is found based on the following equation:

$$x_i^{g+1} = x_i^g + \alpha \oplus \Delta x, \tag{5.11}$$

Here α is the step size, which is related to the search pattern during the normal or urgent state.

$$\Delta x = \begin{cases} Levy(\lambda), & \omega = 1 \\ Brownian\ (\lambda), & \omega = 0 \end{cases}, \tag{5.12}$$

where ω is considered as a pseudo-random integer from a uniform discrete distribution. In the case of a Lévy walk, the random walk is characterized by a step size calculated from a probability distribution with an inverse power-law tail as below (Benhamou, 2007; Viswanathan et al., 2000).

$$\text{Levy} \sim u = t^{-\lambda}, \tag{5.13}$$

where $1 < \lambda < 3$, and t represents the flight length. In the case of a Brownian walk (Benhamou, 2007), the search for a new chamber inside the structure of a multi-chambered lair is characterized by the step size described as below.

$$S = (k * \text{randn}(d, \text{Ndots})) \tag{5.14}$$

Here, k is the standard deviation of the normal distribution for the diffusion rate coefficient, d is the dimensions of the problem, and $Ndots$ represents the number of particles of the Brownian walk in the search space.

```
Ringed Seal Search Data Clustering algorithm
Input: (data,k,pop,NIT)
Output: K clusters
Begin
1. Define the objective function based on the clustering equation
2. Set the k number of clusters centroids to be found
3. Initialize randomly the cluster centroids k
4. Compute the Euclidean distance between the k centroids and the data
   points
5. Assign each data point to the nearest centroid
6. Invoke the RSS to optimize the centroids
   RSS input: (@objective_Function,pop,Data_Size, NIT)
   While ( stopping criterion)
     a) Check the state of the search and perform a specific search pat-
        tern
            State = Check_State()
            if state == 'NormalState'
                 Perform Brownian walk
            end
            if state == 'UrgentState'
                 Perform Levy walk
            end
     b) Evaluate the fitness
        If L^{best,k} > L^{best,k-1}
          Choose the new lair, L^{best} = L^{best,k}
        else
          go to  a
        Endif
     c) 15% of the used lairs are detected and destroyed by bear, another
        new set will be selected randomly from nature;
     d) Rank the solutions;
    Endwhile
7.  return the best lair position as the new centroid;
8.  update the centroids positions
9.  if not stopping criterion goto 4
end
```

Figure 5.3: RSS data clustering pseudo-code

Figure 5.3 describes the main skeleton of RSS data clustering. Starting the algorithm requires a set of input and output parameters. For the inputs, the initial number of lairs is represented by the data file, the number of clusters k, and the size of the data population pop. Setting the number of iterations NIT is also required to adjust the stopping criterion. For the outputs, it is represented by k clusters of the dataset. An example is presented in Figure 5.4.

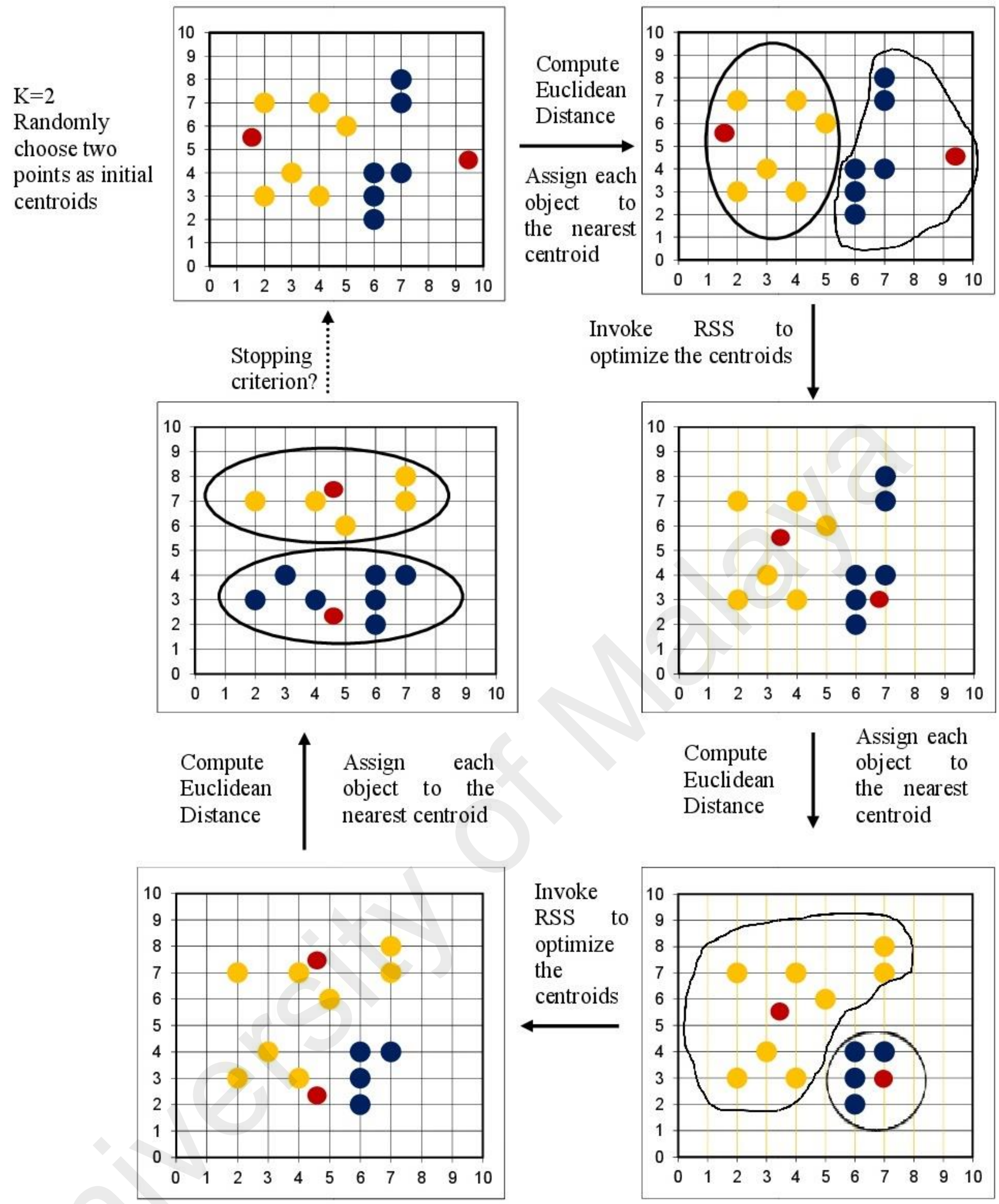


Figure 5.4: Example of RSS data clustering

From Figure 5.4 the RSS-based data clustering algorithm starts with initializing the centroids randomly. Then the distance between the centroids and each data point is computed using the Euclidean distance. Based on that, each data point is assigned to the nearest centroid. Finding the best location of centroids is an iterative process and it is formulated as an optimization problem. RSS which is considered as an optimization algorithm is invoked to find the centroid positions where the distance between all data points is minimized using the Euclidean. After finding the new centroids locations, the distance between data points and the centroids is computed, before assigning the data points to the nearest centroid.

5.4. Experiment

In this section, the results obtained when searching for the solution to the problem formulated in Equation 5.7 were introduced. The RSS approach introduced in this study was compared to well-known baseline approaches such as PSO, GA, and CS, as is common practice in the literature, especially when a new algorithms like ours is being proposed (Karafotias et al., 2015; Senthilnath, Das, Omkar, & Mani, 2012; Van der Merwe & Engelbrecht, 2003; Yang & Deb, 2009). Furthermore, the dataset specifications were used for the evaluation and the analysis of the results are introduced.

5.4.1. Data Specification

The experimental validation is performed on seven different datasets with a variety of levels of complexity, namely Iris, Lung Cancer, Cancer, Ecoli, Ionosphere, Breast Cancer, and Yeast. These datasets are available from the UCI Machine Learning Repository. The datasets are taken from the UCI Learning Repository (Newman et al., 1998). The datasets used in this study can be illustrated as follows.

Table 5.1: Specifications of the clustering dataset

Dataset	Number of objects	Classes (*k*)	Attributes (*dim*)	*k*×*dim*
Iris	150	3	4	12
Lung cancer	32	3	56	168
Cancer	683	2	9	18
Ecoli	336	8	8	64
Ionosphere	351	2	34	68
WDBC	569	2	60	120
Yeast	1484	10	8	80

Table 5.1 summarizes the characteristics of the selected dataset, the total number of objects in each dataset, the number of classes *k*, and the number of attributes, which denotes the dataset dimension *dim*. On the other hand, the cluster size parameter is set to be equal to the number of classes in the dataset. Furthermore, the cluster size is denoted by *k* and the dimension (attributes) is represented by *dim*. The datasets used in this study can be described as follows:

Dataset 1: Iris Plants Database (Iris). This dataset contains 150 objects with four attributes and three classes, where each class refers to a type of iris species.

Dataset 2: Lung Cancer dataset. This dataset contains 32 objects with 56 attributes and three classes. The data illustrate three types of pathological lung cancers.

Dataset 3: Cancer: This dataset consists of 683 objects, where the number of classes is two and the number of attributes is nine. The samples were collected periodically in the form of clinical reports. Therefore, the data reflect this chronological grouping of the data.

Dataset 4: Ecoli. This dataset is obtained from cellular localization sites of proteins. After pre-processing, the data comprise 336 labelled examples, which are described by eight classes and eight attributes. However, three classes are represented by only two, two, and five patterns. These nine examples of data are neglected by using only 327 patterns, five classes, and seven attributes.

Dataset 5: Ionosphere. This dataset consists of radar data, which are collected by a special system. It contains 351 objects and two classes and is represented by 34 attributes.

Dataset 6: Wisconsin Dataset for Breast Cancer: These data consist of 683 object features calculated from Fine Needle Aspirate (FNA) images of breasts. There are two categories: malignant, with 444 objects, and benign, with 239 objects. Each type of class contains nine features.

Dataset 7: Yeast dataset. The objective of these data is similar to that of the Ecoli data, and consists of determining the cellular localization of the yeast proteins (A. C. Tan & Gilbert, 2003).

5.4.2. Evaluation Metrics

In the comparative study, four performance metrics commonly used to evaluate clustering methods were adopted: Accuracy, Rand Index, Davies–Bouldin index and Dunn index. To assess the clustering accuracy, the Overall Clustering Accuracy (OCA) were used (Handaga, Herawan, & Deris, 2012; Olson & Delen, 2008). It is defined by

$$\left(\text{Overall Clustering Accuracy}\right)_i = \frac{\sum_{i=1}^{n}\left(\text{True Clustering}\right)_i}{\left(\text{Total number of cases}\right)_i}, \qquad (5.15)$$

where i denotes the class number and n represents the total number of classes. To evaluate the clustering results based on the data that were clustered, internal evaluation metrics were utilized. In contrast, to evaluate the clustering results based on the data that were not utilized for clustering, external evaluation metrics were utilized.

5.4.2.1. Internal Evaluation Metrics

There are several internal evaluation metrics that can be applied to evaluate a clustering approach and to analyse the efficiency of the clustering of the data points. It is shown that these metrics give the best score to the algorithm that builds clusters with a high similarity value within a cluster and a low similarity value between clusters (Kovács, Legány, & Babos, 2005). In this study, two internal evaluation metrics were used: the Davies–Bouldin (DB) index and the Dunn index (Halkidi, Batistakis, &

Vazirgiannis, 2001; Kovács et al., 2005). The DB index is represented by the following formula:

$$DB = \frac{1}{\kappa}\sum_{i=1}^{\kappa}\max_{j\neq 1}\left(\frac{\sigma_i + \sigma_j}{d\left(c_i, c_j\right)}\right) \tag{5.16}$$

here κ is the total number of clusters, c_κ is the centroid of the κ^{th} cluster, σ_κ is the average distance between all elements in the κ^{th} cluster and the centroid c_κ, and $d\left(c_i, c_j\right)$ represents the distance between two different centroids. It was noted earlier that algorithms with low intra-cluster distances and high inter-cluster distances show very low values of the DB index (Kovács et al., 2005). In view of this, the data clustering algorithm that provides a set of clusters with a small value of the DB index is considered the best.

The Dunn index consists of determining the dense and well-separated clusters (Halkidi et al., 2001). It is denoted as the ratio between the minimal inter-cluster distance and the maximal intra-cluster distance. The Dunn index can be expressed by the following equation:

$$DI = \frac{\min_{1\leq i<j\leq n} d\left(i, j\right)}{\max_{1\leq \kappa\leq n} d'\left(\kappa\right)}\ . \tag{5.17}$$

here, $d(i, j)$ denotes the inter-cluster distance and $d(\kappa)$ denotes the intra-cluster distance of the κ^{th} cluster. As stated in the literature, internal metrics focus on finding clusters with high intra- and low inter-cluster similarity. As a consequence, algorithms that provide high values of the Dunn index are considered the best.

5.4.2.2. External Evaluation Metrics

The external evaluation consists of evaluating the clustering results based on the data that were not used for clustering. In this study, an external metric is utilized, the Rand index, to measure the rate of correct decisions affected by the algorithm (Santos & Embrechts, 2009).

$$RI = \frac{TP + TN}{TP + FP + FN + TN}, \tag{5.18}$$

where TP represents the total number of true positive values, TN denotes the number of true negative values, FP denotes the number of positive values, and FN represents the number of false negative values.

5.4.3. Experimental Setup

The clustering algorithms used for the comparison in this study were programmed in Matlab (R2009b). The results were obtained after running the algorithms on a system based on a 2.66 GHz Intel Core 2 Quad CPU with 2 GB of RAM running Microsoft Windows 7. The effectiveness of the clustering outputs depends upon the initial parameter settings, including the population size of the metaheuristic algorithms. Therefore, the parameter settings for GA, PSO, CS, and RSS are listed in the following table.

Table 5.2: Parameter settings adopted for the comparison

Algorithms	Parameters
GA	Population size = 10, crossover = 0.8, Mutation = 0.005,
PSO	W_{min} = 0.4, w_{max} = 0.9, population size = 10, social = cognitive = velocity = 2
CS	Population size = 10, ρ = 0.25, α =1
RSS	Population size =10

Table 5.2 shows the parameter settings adopted for the comparison between RSS and other baseline approaches reported in the literature (Binu, 2015; De Falco, Della Cioppa, & Tarantino, 2007; Kuo & Lin, 2010; Yang & Deb, 2009). For GA, the population size is fixed at 10, the crossover rate is set to 0.8, and the mutation rate is set to 0.005 based on the suggestion given in (Kuo & Lin, 2010). For PSO, the tuned parameters are W_{min} = 0.4, W_{max} = 0.9, population size = 10, and social = cognitive = velocity = 2, as stated in (De Falco et al., 2007). For CS, the population size is set to 10, the rate of detection of eggs ρ is set to 0.25, and the step size of the Lévy flight search is set to $\alpha = 1$, as suggested in (Yang & Deb, 2009). For RSS, the population size is set to 10 as highlighted in Chapter 4. The total number of iterations is set to 50, which is a particularly common choice for achieving global convergence with all optimization algorithms.

5.4.4. Results and Discussion

In this section, the results obtained using the proposed RSS algorithm when searching for the solution to the clustering problem formulated in Equation 5.7 were discussed. These experiments were carried out in order to test the efficiency of the RSS algorithm for small- and medium-scale dimensions. The algorithms were each run 50 times on seven dataset problems. Furthermore, the results achieved by RSS were compared with those obtained by the three baseline clustering approaches: GA, PSO, and CS.

5.4.4.1. Clustering Accuracy Analysis

In this section, Equation 5.15 is selected as the performance indicator, where a high output value indicates better performance. Furthermore, the experiments were conducted for 50 runs and the average was computed as the total accuracy achieved, as illustrated below.

Table 5.3: Accuracy index of RSS compared to GA, PSO, and CS

Algorithms	IRIS	Lung	Cancer	Ecoli	Ionosphere	WDBC	Yeast
GA	0.7	0.40741	0.56369	0.48214	0.69516	0.2974	0.36321
CS	0.72	0.48148	0.53001	0.3869	0.66667	0.3452	0.33356
PSO	0.75333	0.44444	0.95315	0.49702	0.62108	0.298	0.3686
RSS	**0.82**	**0.59259**	**0.95754**	**0.69345**	**0.70655**	**0.3682**	**0.37871**

Table 5.3 summarizes the experimental results of the Iris, Lung, Cancer, Ecoli, Ionosphere, WDBC, and Yeast datasets. The Cancer dataset achieved an estimated maximum best accuracy of 0.95754. However, the lowest accuracy value was 0.2974, which was achieved by GA on the WDBC dataset. The second highest value of accuracy was achieved by PSO on the Cancer dataset. For GA and CS, the most accurate results achieved were only 0.7 and 0.72, respectively, as successive attempts of 50 iterations failed to achieve good results compared with those obtained by the proposed RSS algorithm. The performance of each clustering algorithm was assessed on seven different datasets as inputs and the clustering outputs are visualized in terms of accuracy as shown below.

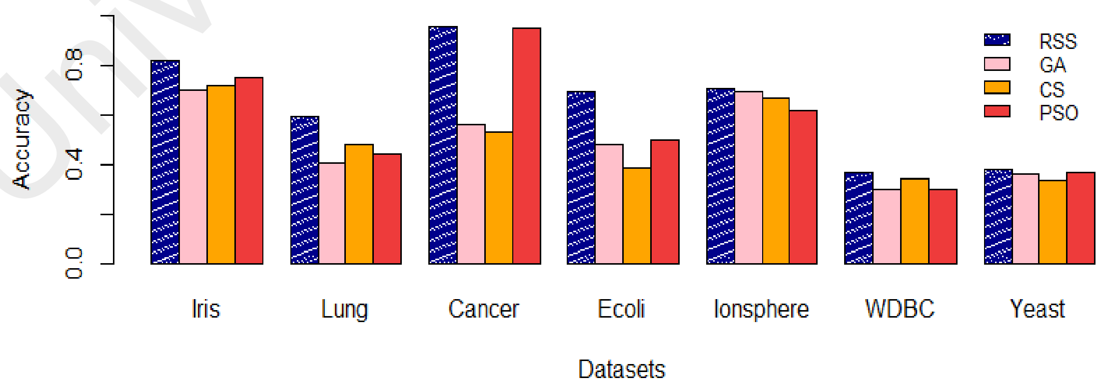


Figure 5.5: Accuracy results of RSS compared with GA, PSO, and CS

From Figure 5.5, it is clear that RSS outperforms the other algorithms in terms of accuracy. This can be clearly seen for all seven datasets, with the highest accuracy rate being achieved for Cancer and the lowest for WDBC. In fact, these outputs agree with the convergence rate of RSS algorithm reported in Chapter 4. Moreover, this may be discussed in terms of the solution quality achieved by each algorithm, where RSS clustering is able to provide a better solution quality, as stated in Chapter 4 .

5.4.4.2. Rand Index Analysis

Since evaluating the performance of the RSS compared to other algorithms on several datasets according to the different accuracy values, the most straightforward way of using these values may lead to a strong analysis of the achieved results. This implies that an external evaluation measure should be used in order to refine the analysis. In this section, the performance of the RSS will be evaluated in terms of the Rand index, where the percentage of correct decisions made by the algorithms is computed as described in Equation 5.18. The obtained results are listed in the table below.

Table 5.4: The achieved Rand index of RSS compared to the other algorithms

Algorithms	IRIS	Lung	Cancer	Ecoli	Ionosphere	WDBC	Yeast
GA	0.76063	0.38462	0.50739	0.74328	0.57496	0.58433	**0.67646**
CS	0.78201	0.44729	0.50107	0.70624	0.55429	0.57633	0.6109
PSO	0.76134	0.48148	0.91056	0.74675	0.52798	0.55213	0.61514
RSS	**0.82362**	**0.61254**	**0.91857**	**0.84577**	**0.58414**	**0.62022**	0.6121

Here, Table 5.4 illustrates the results of comparison of RSS with other clustering algorithms in terms of the Rand index applied on the seven datasets. According to the Rand index, the performance of the RSS algorithm gives consistent and improved outputs compared to GA, PSO, and CS on all the datasets used except for the Yeast dataset. RSS obtained the highest value of 0.91857 on the Cancer, while PSO obtained a slightly lower value of 0.91056 on the same dataset. In contrast, the results achieved with CS and GA were lower, at 0.78201 and 0.76063, respectively. There is an obvious resemblance between the Rand index results and the accuracy results, which provides evidence of the superior performance of RSS compared to GA, PSO, and CS.

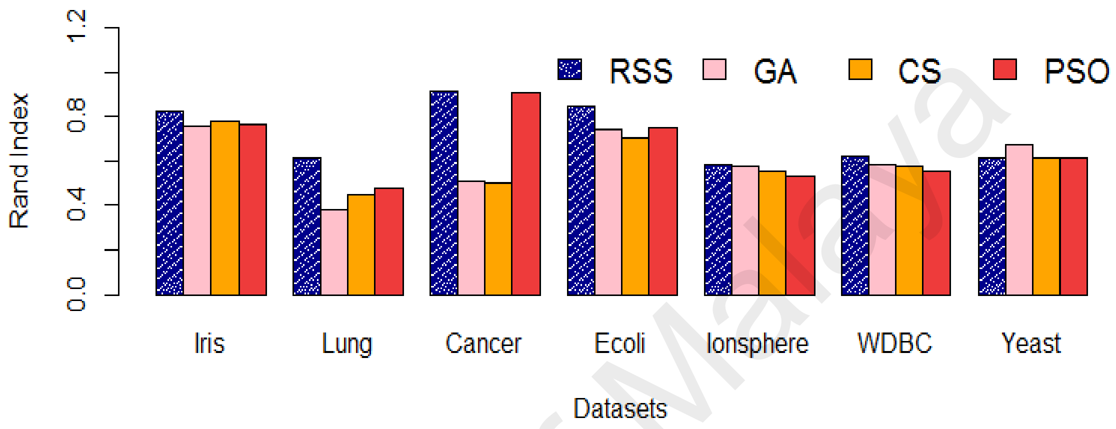


Figure 5.6: The achieved Rand index of RSS compared to the other algorithms

Figure 5.6 shows the Rand index results of RSS compared to GA, PSO, and CS. The results were obtained on the seven different datasets, where the clustering problem is formulated to find the centroid of each dataset. As can be observed from the figure, the proposed RSS algorithm provided higher Rand index values than the other algorithms on most datasets except Yeast dataset. In contrast, the other three algorithms, GA, PSO, and CS, achieved lower values of the Rand index, except on the Yeast dataset where GA performs better. In order to give credence to the outputs obtained by the RSS, further compelling evidence is provided in the following subsections.

5.4.4.3. Dunn Index Analysis

One strategy to give more credibility to the obtained results consists in determining the well-separated and dense clusters by applying the Dunn index. The achieved results are illustrated below.

Table 5.5: Dunn index results of RSS compared to GA, PSO, and CS

Algorithms	IRIS	Lung	Cancer	Ecoli	Ionosphere	WDBC	Yeast
GA	0.051976	0.4533	0.038837	0.0438	0.03082	0.11597	0.016476
CS	0.054671	0.38818	0.039498	0.0384	0.043811	0.11808	0.016844
PSO	0.038405	0.4533	0.054924	0.0405	0.043622	0.10065	0.009621
RSS	**0.076344**	**0.50422**	**0.1148**	**0.0507**	**0.054445**	**0.11964**	**0.02237**

Here, Table 5.5 shows the Dunn index results of RSS on several datasets compared with GA, PSO, and CS. RSS was able to achieve better results than the other algorithms on most of the datasets. The best score was reported for the Lung dataset and was 0.50422. The lowest score was obtained by PSO on the Yeast dataset. These results closely match those obtained with the Rand index and corroborate the other metrics used in these experiments. In order to obtain further insight from this metric, the results are visualized below.

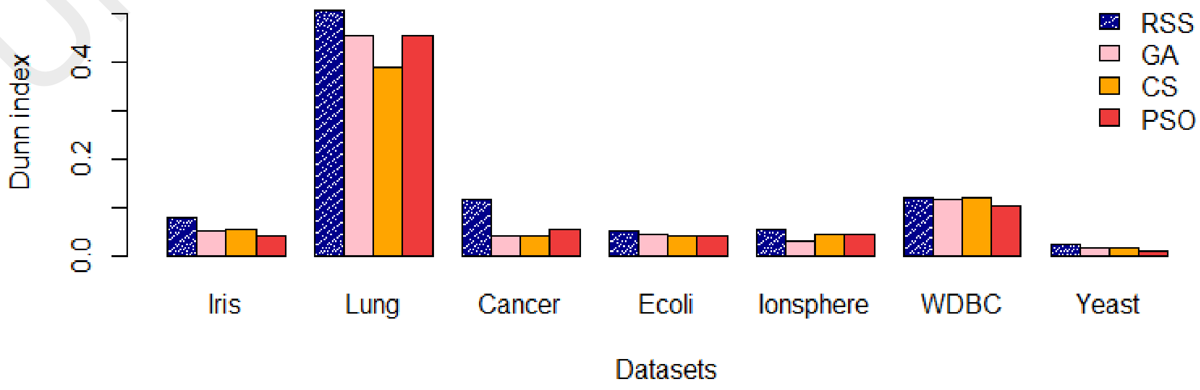


Figure 5.7: Dunn index of RSS compared to GA, PSO, and CS

From Figure 5.7, it is clear that the Lung dataset provides high values of the results compared to the other datasets, and RSS achieved the highest value, followed by GA and PSO. RSS achieved a better value of the Dunn index on the other datasets than GA, PSO, and CS did. These findings are consistent with the previous index findings and also with the aim of the Dunn metric, where the dense and well-separated clusters are determined.

5.4.4.4. DB Index Analysis

In order to substantiate the outputs, another level of internal evaluation is used by employing the DB index. It is important to emphasize that the lowest value represents the best value of the DB index (Medeiros, Xavier, & Canuto, 2015). This index is known for its ability to measure the intra-cluster distances, which can provide a number of important new insights. The results are listed in the table below.

Table 5.6: DB index results of RSS compared to GA, PSO, and CS

Algorithms	IRIS	Lung	Cancer	Ecoli	Ionosphere	WDBC	Yeast
GA	0.93346	2.0311	2.2944	0.6232	1.1705	1.0871	0.62967
CS	0.82183	0.92856	2.1181	0.6241	2.1359	1.2356	0.63753
PSO	0.57193	0.77988	0.91136	0.4001	1.4851	1.5332	0.74618
RSS	**0.36049**	**0.77476**	**0.62628**	**0.2886**	**1.1047**	**0.86649**	**0.57309**

The results in Table 5.6 show that when RSS is compared with the other clustering algorithms in terms of the DB index, RSS performs better. The best score is obtained by RSS on the Ecoli dataset, while the other algorithms, GA, PSO, and CS, performed relatively less well. These DB index outputs prove that RSS provides the lowest intra-cluster distances and corroborates the superior performance of the RSS results according to the Dunn index. It is worth examining the DB index results more closely using the following bar graph.

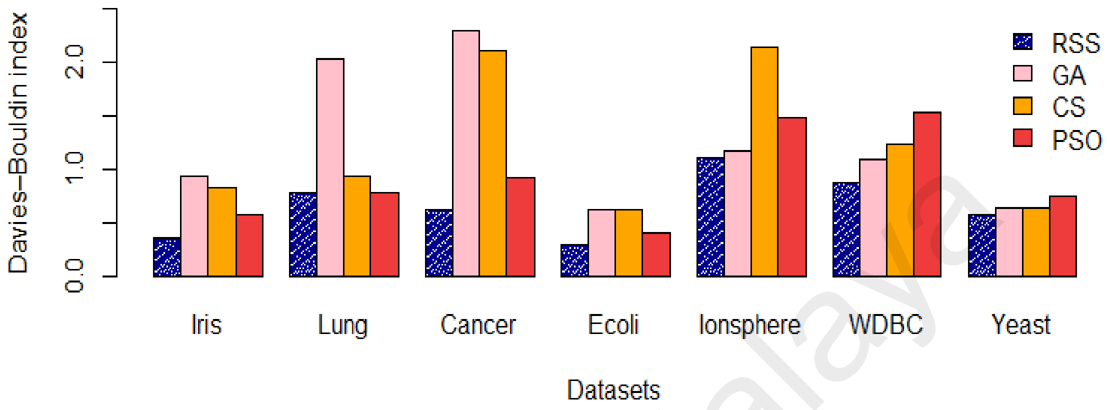


Figure 5.8: Dunn index of RSS compared to GA, PSO, and CS

From Figure 5.8, the small value of the objective function indicates a high quality of clustering. The smallest value was achieved by RSS on the Ecoli and Iris datasets. The bar graph also highlights clearly the datasets where the performance was low, especially Lung, Cancer, and Ionosphere, for which the GA and CS results were quite poor. The reason behind the unsatisfactory results of GA, CS, and PSO is not completely understood. However, RSS was able to deliver better DB values, in agreement with the results of other indexes used in this experimental study.

5.5. Significance

The significance points of this case study are listed as follows:

1) RSS for data clustering: A new data clustering algorithm via the global optimization algorithm RSS is introduced. Its features improved accuracy and fast convergence and outperforms GA, PSO, and CS on most of the datasets investigated.

2) Evaluation: To validate the RSS, seven datasets and four internal and external metrics were synthetically used. Then the performance outputs were widely analysed in terms of various aspects such as the solution quality, convergence speed, accuracy, Rand index, Dunn index, and DB index. The results showed that clustering-based RSS performs better in most of metrics, except in Rand index where clustering-based GA performs better than clustering-based RSS in Yeast dataset.
3) Fewer parameters need to be set up for RSS compared to the GA, PSO, and CS clustering approaches. Therefore, finding the right value of initial setting for RSS is quite easy, making it applicable to several clustering problems.

5.6. Summary

This study has introduced a novel clustering approach based on RSS. The approach consists of using the RSS algorithm to find data point centroids. It is shown that the RSS has the capability to provide a fast convergence featured by an optimal trade-off between the solution quality and the convergence maturity of the achieved solutions. The first step in building the RSS-based data clustering requires the formulation of the clustering problem as an optimization problem. In this study, the objective function was carried out by optimizing the sum of all set instances of the Euclidean distance in order to find the data point centroids. Four metrics were used in order to test the proposed approach in comparison with other approaches. The Rand index, DB index, Dunn index, and accuracy metric were applied on seven benchmark datasets: Iris, Lung, Cancer, Ecoli, Ionosphere, WDBC, and Yeast.

The experimental results showed that the RSS-based data clustering performed better than the other clustering algorithms based on GA, PSO, and CS on most of the datasets. This superior performance is visualized in four figures and discussed, and the interpretation of the clustering results matched the convergence rate of the optimization algorithms used in the proposed algorithm RSS. Furthermore, the results of the proposed approach demonstrate that using optimization algorithms featuring an optimal exploitation–exploration balance can lead to better clustering results.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1. Overview

This study proposed a metaheuristic approach for global optimization using a composite search model called the sensitive search model inspired from the ringed seal movement, where an adaptive balance between exploitation and exploration characterizes the search for solutions. The proposed algorithm not only can solve global optimization problems, but also can provide a low time consumption, fast convergence, optimal solution quality, optimal diversity and mature convergence compared to other baseline approaches.

6.2. Summary of Results

The results were summarized by answering research questions from Chapter 1 as follows:

Research Question 1: How to develop a composite search model to adaptively balance exploitation and exploration?

There are different methods to build a composite search model. One of the well-known methods is based on animal movement. The movement of the ringed seal is modeled, which is characterized by a composite search and sensitivity to external noise. A comparative study is presented in Section 2.6.3 to select the most suitable random walk to model the ringed seal movement based on exploitation-exploration requirements. The proposed composite search will be called the sensitive search model, where the search is able to adaptively balance exploitation and exploration.

Research Question 2: How to derive a metaheuristic algorithm for solving global optimization based on the composite search model?

According to the developed composite search, which is called the sensitive search model, a metaheuristic algorithm called Ringed Seal Search (RSS) were derived. The methodology to develop RSS is described in details in Chapter 3. In this context, the focus was on inputs and outputs of RSS since the search for solution is developed in the RSS. The inputs were represented by a set of population and the outputs are represented by the best lairs.

Research Question 3: What would be the optimum way to test the developed metaheuristic algorithm?

One of the well-known methods to test metaheuristics especially when new algorithms like ours are proposed is using benchmark test functions. A comprehensive set of fifteen benchmark test functions, collected from references were used to test the performance of the proposed algorithm. According to the references mentioned above, the selected functions fulfill the requirements of uni-objective and multi-objective problems. It is very important to highlight that the main target of this benchmarking test is to check whether the proposed algorithm RSS is able to solve uni-objective and multi-objective optimization problems.

6.3. Achievements of Objectives

The objectives of this research are as follows:

i. To develop a composite search model based on animal movement to adaptively balance exploitation and exploration.

ii. To derive a metaheuristic algorithm for solving global optimization based on the composite search model.

iii. To test, validate and compare the effectiveness of the metaheuristic algorithm in terms of time, solution quality, convergence, maturity and diversity of the search.

A composite search model based on the ringed seal movement called the sensitive search model, proposed to adaptively balance exploitation and exploration in order to achieve the first objective. The proposed composite search model was designed, implemented and embedded later in the proposed metaheuristic algorithm, which will be called Ringed Seal Search (RSS). The development of the proposed composite search model can be described as follows:

- A detailed review about modeling search approaches based on animal movement (cf. Section 2.6).
- A comparative study between the existing random walks to refine the most suitable random walks that can be employed to build a composite search model where exploitation and exploration can be adaptively balanced (cf. Section 2.6.3).
- Designing and Modeling of the sensitive search model is described. The formal definition and the mathematical model of the sensitive search model is represented to explain how exploitation and exploration are balanced via the sensitive search model (cf. Section 3.2.4).

A metaheuristic algorithm called Ringed Seal Search (RSS) is derived based on the developed composite search in order to achieve the second objective. The proposed algorithm was designed and implemented by using several modules doing different tasks. Each module represents a function that is developed independently such as: Generating Initial Lairs, Seal’s Search for Lairs, Random Noise, Normal State, Urgent

State, Best Lairs Updating, Abandoning Worst Lairs and Convergence to optimal lairs. (cf. Section 3.2.4.4).

The proposed algorithm RSS has high quality in terms of solution quality, convergence, time consumption, diversity and maturity. This outperformance is due to the proposed composite search model which is called the sensitive search model, where balance between exploitation and exploration is considered. This has been proved in the experimental evaluation explained in Chapter 4. This has answered the third objective. the proposed algorithm scored a high improvement compared to other baseline algorithms GA, PSO and CS.

6.4. Contributions

There are a number of metaheuristic algorithms for global optimization. However, most of them have difficulties in terms of tuning up balance between exploitation and exploration and most of them are requiring predefinition of parameters, which results in loss of diversity, high time consumption, immature search and slow convergence (BoussaïD et al., 2013; Črepinšek et al., 2013; Gandomi, Yang, Alavi, & Talatahari, 2013; Gao et al., 2015; Yang et al., 2014; Yazdani & Jolai, 2016). On the other hand, there are some search models, which are not compatible with exploitation-exploration requirements since they have some problems such as supporting only one search pattern exploitation or exploration and predefinition of parameters tuning. Yet there is no focus in the literature on such issues. Therefore, a new metaheuristic for global optimization inspired from the movement of the ringed seal were presented in this thesis to overcome the aformentioned problems. Furthermore, according to global optimization problems, the challenges in exploitation exploration balance and tuning had to be considered in the proposed algorithm (Črepinšek et al., 2013). The specific contribution of this thesis can be summarized as follows:

5. A new composite search model called the sensitive search model featured by an adaptive balance between exploitation and exploration.
6. A new metaheuristic algorithm, RSS, for global optimization is derived from the composite search model.
7. The number of parameters to tune for the proposed algorithm RSS is only one parameter, making RSS to be less sensitive to parameters settings compared to PSO, GA and CS.
8. An extensive evaluation based on five metrics: time consumption, convergence, diversity, solution quality and maturity. Moreover, a validation based on a clustering optimization problem were introduced.

6.5. Significance

The proposed RSS algorithm is not only able to solve uni-objective and multi-objective global optimization problems, but it is capable to adaptively balance exploitation and exploration, which results in fast convergence, high solution quality, optimal time consumption, mature convergence and optimal diversity. The significance of such algorithms can be summarized into two main points:

- From a heuristic search point of view, RSS algorithm represents an innovative way to solve tuning balance between exploitation and exploration. Moreover, this research constitutes the second attempt in metaheuristic algorithms that addresses the problem of tuning balance between exploration and exploitation after a first attempt presented by Yang et al. (2014).
- From optimization applications point of view, the results obtained in this study showed that RSS has the potentials to be used in solving several optimization problems such as cancer classification applications (Marisa et al., 2013), optimization of web service composition processes (Chifu, Pop,

Salomie, Suia, & Niculici, 2012), vehicle routing system applications (Vidal, Crainic, Gendreau, Lahrichi, & Rei, 2012), design of embedded systems (Kumar & Chakarverty, 2011), collective robotic search applications (Doctor, Venayagamoorthy, & Gudise, 2004), data clustering applications (Alok, Saha, & Ekbal, 1701, 2015; Saha, Alok, & Ekbal, 2015; Senthilnath, Das, Omkar, & Mani, 2013), digital games applications (Singh & Deep, 2015), medical images applications (Chuang, Lin, Chang, & Yang, 2012), etc.

6.6. Limitation of Current Study

- In this research, power-law distributions were used to build a composite search model. However, in mathematics, there is several probability distributions and more analysis is required to determine if other kinds of distributions can increase the quality.
- New particular measures are needed to understand how different RSS components participate to exploration and exploitation during the search for the global optimal solution. Moreover, these measurements are needed to understand how exploitation and exploration are balanced during uni-objective and multi-objective problems.
- It is observed that the current composite search model, the sensitive search model of RSS is not able to demonstrate that the regions previously visited are not revisited again, especially if there is switching between search states as the proposed algorithm.

6.7. Recommendations and Future Directions

The researcher strongly believes that the proposed algorithm in this research not only able to adaptively balance exploitation and exploration during searching for global optimal solutions but also has the capability to highly improves the quality of the global optimal solutions for uni-objective and multi-objective optimization problems. Hopefully, the proposed algorithm is used by other researchers in other real applications with uni and multi-objectives optimization problems or in applications with requirements of tuning of exploitation and exploration balance. Apart from the above improvements, some important refinements were discussed as follows:

- In real applications, there is other different kinds of optimization problems with several dimensions. Expending the proposed algorithm to be applicable for other high dimensions is required for solving extremely complex optimization problems.
- Deep learning is gaining more areas in several domains, especially where NP-hard problems are occurring. Therefore, incorporating deep learning in the process of exploitation-exploration during searching for optimal solutions is a further research topic.
- With the emergence of big data, which is evolving and changing, global optimization is a specific approach to deal with lot of applications especially the parallel processing requirements. Therefore, extending proposed algorithms to meet the requirements for parallel processing in big data is another interesting issue for research.

Hopefully the research presented in this dissertation will inspire more research from the researchers and practitioners in the field.

REFERENCES

Abbass, H. A. (2001). *MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach.* Paper presented at the Evolutionary Computation, 2001. Proceedings of the 2001 Congress on.

Adorio, E. P., & Diliman, U. (2005). Mvf-multivariate test functions library in c for unconstrained global optimization. *Quezon City, Metro Manila, Philippines*, 44.

Affenzeller, M., Wagner, S., & Winkler, S. (2008). *Evolutionary systems identification: New algorithmic concepts and applications*: INTECH Open Access Publisher.

Alba, E., & Dorronsoro, B. (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *Evolutionary Computation, IEEE Transactions on, 9*(2), 126-142. doi: 10.1109/TEVC.2005.843751

Alok, A. K., Saha, S., & Ekbal, A. (1701). Multi-objective semi-supervised clustering for automatic pixel classification from remote sensing imagery. *Soft Computing*, 1-19.

Alok, A. K., Saha, S., & Ekbal, A. (2015). A new semi-supervised clustering technique using multi-objective optimization. *Applied Intelligence*, 1-29.

Arditi, R., & Dacorogna, B. (1985). Optimal foraging in nonpatchy habitats. I. Bounded one-dimensional resource. *Mathematical biosciences, 76*(2), 127-145.

Auger, A., Schoenauer, M., & Teytaud, O. (2005). *Local and global order 3/2 convergence of a surrogate evolutionary algorithm.* Paper presented at the Proceedings of the 7th annual conference on Genetic and evolutionary computation.

Bartumeus, F., & Catalan, J. (2009). Optimal search behavior and classic foraging theory. *Journal of Physics A: Mathematical and Theoretical, 42*(43), 434002.

Bartumeus, F., Catalan, J., Fulco, U., Lyra, M., & Viswanathan, G. (2002). Optimizing the encounter rate in biological interactions: Lévy versus Brownian strategies. *Physical Review Letters, 88*(9), 097901.

Bartumeus, F., Da Luz, M. G. E., Viswanathan, G. M., & Catalan, J. (2005). Animal Search Strategies: A Quantitative Random-Walk Analysis. *Ecology, 86*(11), 3078-3087. doi: 10.1890/04-1806

Bartumeus, F., Raposo, E. P., Viswanathan, G. M., & da Luz, M. G. (2014). Stochastic Optimal Foraging: Tuning Intensive and Extensive Dynamics in Random Searches. *PloS one, 9*(9), e106373.

Bastille-Rousseau, G., Murray, D. L., Schaefer, J. A., Lewis, M. A., Mahoney, S., & Potts, J. R. (2017). Spatial scales of habitat selection decisions: implications for telemetry-based movement modelling. *Ecography*.

Ben Ghalia, M. (2008, 10-13 Aug. 2008). *Particle swarm optimization with an improved exploration-exploitation balance.* Paper presented at the Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on.

Benhamou, S. (1992). Efficiency of area-concentrated searching behaviour in a continuous patchy environment. *Journal of Theoretical Biology, 159*(1), 67-81.

Benhamou, S. (2007). How many animals really do the Levy walk? *Ecology [H.W.Wilson - GS], 88*(8), 1962.

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

Binu, D. (2015). Cluster analysis using optimization algorithms with newly designed objective functions. *Expert systems with Applications, 42*(14), 5848-5859.

Birattari, M., & Kacprzyk, J. (2009). *Tuning metaheuristics: a machine learning perspective* (Vol. 197): Springer.

Blum, C., & Li, X. (2008). Swarm intelligence in optimization *Swarm Intelligence* (pp. 43-85): Springer.

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing, 11*(6), 4135-4151.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR), 35*(3), 268-308.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*: Oxford university press.

BoussaïD, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82-117.

Buldyrev, S. V., Havlin, S., da Luz, M. G. E., Stanley, H. E., Raposo, E. P., & Viswanathan, G. M. (1999). Optimizing the success of random searches. *Nature, 401*(6756), 911-914. doi: 10.1038/44831

Burnwal, S., & Deb, S. (2013). Scheduling optimization of flexible manufacturing system using cuckoo search-based approach. *The International Journal of Advanced Manufacturing Technology, 64*(5-8), 951-959.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City, 1*(2), 1.

Charnov, E. L. (1976). Optimal foraging, the marginal value theorem. *Theoretical population biology, 9*(2), 129-136.

Chen, J., Xin, B., Peng, Z., Dou, L., & Zhang, J. (2009). Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE* *transactions on systems, man, and cybernetics-part A: systems and humans, 39*(3), 680-691.

Chifu, V. R., Pop, C. B., Salomie, I., Suia, D. S., & Niculici, A. N. (2012). Optimizing the semantic web service composition process using cuckoo search *Intelligent distributed computing V* (pp. 93-102): Springer.

Chiroma, H., Khan, A., Abubakar, A. I., Saadi, Y., Hamza, M. F., Shuib, L., . . . Herawan, T. (2016). A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm. *Applied soft computing*.

Chu, S.-C., Tsai, P.-W., & Pan, J.-S. (2006). *Cat swarm optimization.* Paper presented at the Pacific Rim International Conference on Artificial Intelligence.

Chuang, L.-Y., Lin, Y.-D., Chang, H.-W., & Yang, C.-H. (2012). An improved PSO algorithm for generating protective SNP barcodes in breast cancer. *PloS one, 7*(5), e37018.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation, 219*(15), 8121-8144.

Črepinšek, M., Liu, S.-H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR), 45*(3), 35.

Cuevas, E., Echavarría, A., & Ramírez-Ortegón, M. (2014). An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied Intelligence, 40*(2), 256-272. doi: 10.1007/s10489-013-0458-0

Cuevas, E., Osuna, V., & Oliva, D. (2017). *Evolutionary Computation Techniques: A Comparative Perspective* (Vol. 686): Springer.

Damaševičius, R., & Woźniak, M. (2017). *State Flipping Based Hyper-Heuristic for Hybridization of Nature Inspired Algorithms.* Paper presented at the International Conference on Artificial Intelligence and Soft Computing.

De Falco, I., Della Cioppa, A., & Tarantino, E. (2007). Facing classification problems with particle swarm optimization. *Applied soft computing, 7*(3), 652-658.

De Rosa, G. H., Papa, J. P., & Yang, X.-S. (2017). Handling dropout probability estimation in convolution neural networks using meta-heuristics. *Soft Computing*, 1-10.

Deb, Sindhya, K., & Hakanen, J. (2016). Multi-objective optimization *Decision Sciences: Theory and Practice* (pp. 145-184): CRC Press.

Deb, S., Fong, S., & Tian, Z. (2015). *Elephant Search Algorithm for optimization problems.* Paper presented at the Digital Information Management (ICDIM), 2015 Tenth International Conference on.

Dees, N. D. (2009). The role of stochastic resonance and physical constraints in the evolution of foraging strategy.

Deng, Y., Liu, Y., & Zhou, D. (2015). An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering, 2015*.

Doctor, S., Venayagamoorthy, G. K., & Gudise, V. G. (2004). *Optimal PSO for collective robotic search applications.* Paper presented at the Evolutionary Computation, 2004. CEC2004. Congress on.

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE, 1*(4), 28-39.

Du, K.-L., & Swamy, M. (2016a). Bacterial Foraging Algorithm *Search and Optimization by Metaheuristics* (pp. 217-225): Springer.

Du, K.-L., & Swamy, M. (2016b). Particle swarm optimization *Search and Optimization by Metaheuristics* (pp. 153-173): Springer.

Eberhart, R. C., & Kennedy, J. (1995). *A new optimizer using particle swarm theory.* Paper presented at the Proceedings of the sixth international symposium on micro machine and human science.

Edwards, A. M., Phillips, R. A., Watkins, N. W., Freeman, M. P., Murphy, E. J., Afanasyev, V., . . . Viswanathan, G. M. (2007). Revisiting Levy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature, 449*(7165), 1044-1048. doi: http://www.nature.com/nature/journal/v449/n7165/suppinfo/nature06199_S1.html

Eiben, A. E., & Schippers, C. A. (1998). On evolutionary exploration and exploitation. *Fundamenta Informaticae, 35*(1-4), 35-50.

Eiben, A. E., & Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation, 1*(1), 19-31.

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

Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization, 38*(2), 129-154.

Fagan, F., & van Vuuren, J. H. (2013). A unification of the prevalent views on exploitation, exploration, intensification and diversification. *International Journal of Metaheuristics, 2*(3), 294-327.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena, 22*(1), 187-204.

Findik, O. (2015). Bull optimization algorithm based on genetic operators for continuous optimization problems. *Turkish Journal of Electrical Engineering & Computer Sciences, 23*(Sup. 1), 2225-2239.

Fister, I., Yang, Fister, D., & Fister, I. (2014). Cuckoo Search: A Brief Literature Review. In X.-S. Yang (Ed.), *Cuckoo Search and Firefly Algorithm: Theory and Applications* (pp. 49-62). Cham: Springer International Publishing.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

Floudas, C. A., & Pardalos, P. M. (2014). *Recent advances in global optimization*: princeton University press.

Fogel, D. B. (2006). *Evolutionary computation: toward a new philosophy of machine intelligence* (Vol. 1): John Wiley & Sons.

Fong, S., Deb, S., & Yang, X.-S. (2015). A heuristic optimization method inspired by wolf preying behavior. *Neural Computing and Applications, 26*(7), 1725-1738.

Fong, S., Deb, S., & Yang, X.-s. (2018). How Meta-heuristic Algorithms Contribute to Deep Learning in the Hype of Big Data Analytics *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications* (pp. 3-25): Springer.

Freeman, M. P., Stanley, H. E., Watkins, N. W., Murphy, E. J., Afanasyev, V., Raposo, E. P., . . . Viswanathan, G. M. (2007). Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature, 449*(7165), 1044-1048. doi: 10.1038/nature06199

Gandomi, & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation, 17*(12), 4831-4845.

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

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

Gao, Y., Du, W., & Yan, G. (2015). Selectively-informed particle swarm optimization. *Scientific reports, 5*.

Gautestad, A. O. (2012). Brownian motion or Lévy walk? Stepping towards an extended statistical mechanics for animal locomotion. *Journal of the Royal Society, Interface / the Royal Society, 9*(74), 2332-2340. doi: 10.1098/rsif.2012.0059

Gautestad, A. O., & Mysterud, A. (2013). The Levy flight foraging hypothesis: forgetting about memory may lead to false verification of Brownian motion. *Movement Ecology, 1*(1), 9-9. doi: 10.1186/2051-3933-1-9

Ghaemi, M., & Feizi-Derakhshi, M.-R. (2014). Forest optimization algorithm. *Expert systems with Applications, 41*(15), 6676-6687.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*: CRC press.

Gjertz, I., & Lydersen, C. (1986). Polar bear predation on ringed seals in the fast-ice of Hornsund, Svalbard. *Polar Research, 4*(1), 65-68.

Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence, 25*(4), 503-526.

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

Goldberg, D. E. (2013). *The design of innovation: Lessons from and for competent genetic algorithms* (Vol. 7): Springer Science & Business Media.

Gudise, V. G., & Venayagamoorthy, G. K. (2003). *Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks.* Paper presented at the Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems, 17*(2-3), 107-145.

Hammill, M., & Smith, T. (1991). The role of predation in the ecology of the ringed seal in Barrow Strait, Northwest Territories, Canada. *Marine Mammal Science, 7*(2), 123-135.

Handaga, B., Herawan, T., & Deris, M. M. (2012). FSSC: An Algorithm for Classifying Numerical Data Using Fuzzy Soft Set Theory. *International Journal of Fuzzy System Applications (IJFSA), 2*(4), 29-46.

Hills, T. T., & Adler, F. R. (2002). Time's crooked arrow: optimal foraging and rate-biased time perception. *Animal Behaviour, 64*(4), 589-597.

Hills, T. T., Todd, P. M., Lazer, D., Redish, A. D., & Couzin, I. D. (2015). Exploration versus exploitation in space, mind, and society. *Trends in Cognitive Sciences, 19*(1), 46-54. doi: 10.1016/j.tics.2014.10.004

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

Horst, R., & Tuy, H. (2013). *Global optimization: Deterministic approaches*: Springer Science & Business Media.

Humphries, N. E., Queiroz, N., Dyer, J. R., Pade, N. G., Musyl, M. K., Schaefer, K. M., . . . Houghton, J. D. (2010). Environmental context explains Lévy and Brownian movement patterns of marine predators. *Nature, 465*(7301), 1066-1069.

Islam, M. J., Li, X., & Mei, Y. (2017). A Time-Varying Transfer Function for Balancing the Exploration and Exploitation ability of a Binary PSO. *Applied soft computing*.

Ito, H., Uehara, T., Morita, S., Tainaka, K.-i., & Yoshimura, J. (2013). Foraging behavior in stochastic environments. *Journal of ethology, 31*(1), 23-28.

James, J., & Li, V. O. (2015). A social spider algorithm for global optimization. *Applied soft computing, 30*, 614-627.

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150-194.

Jans, R., & Degraeve, Z. (2007). Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research, 177*(3), 1855-1875.

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2*, 62-72.

Karafotias, G., Hoogendoorn, M., & Eiben, Á. E. (2015). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation, 19*(2), 167-187.

Kashiwagi, A., Urabe, I., Kaneko, K., & Yomo, T. (2006). Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PloS one, 1*(1), e49. doi: 10.1371/journal.pone.0000049

Kazimierski, L. D., Abramson, G., & Kuperman, M. N. (2015). Random-walk model to study cycles emerging from the exploration-exploitation trade-off. *Physical Review E, 91*(1), 012124.

Kelly, B. P., Badajos, O. H., Kunnasranta, M., Moran, J. R., Martinez-Bakker, M., Wartzok, D., & Boveng, P. (2010). Seasonal home ranges and fidelity to breeding sites among ringed seals. *Polar Biology, 33*(8), 1095-1109. doi: 10.1007/s00300-010-0796-x

Kennedy, J. (1999). *Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance.* Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.

Kennedy, J., & Eberhart, R. (1995, Nov/Dec 1995). *Particle swarm optimization.* Paper presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on.

Khan, S. U., Yang, S., Wang, L., & Liu, L. (2016). A modified particle swarm optimization algorithm for global optimizations of inverse problems. *IEEE Transactions on Magnetics, 52*(3), 1-4.

Knysh, D. S., & Kureichik, V. M. (2010). Parallel genetic algorithms: a survey and problem state of the art. *Journal of Computer and Systems Sciences International, 49*(4), 579-589. doi: 10.1134/S1064230710040088

Kovács, F., Legány, C., & Babos, A. (2005). *Cluster validity measurement techniques.* Paper presented at the 6th International symposium of hungarian researchers on computational intelligence.

Kovacs, K. M., Lydersen, C., & Gjertz, I. (1996). Birth-site characteristics and prenatal molting in bearded seals (Erignathus barbatus). *Journal of Mammalogy, 77*(4), 1085-1091.

Kumar, A., & Chakarverty, S. (2011). *Design optimization for reliable embedded system using Cuckoo Search.* Paper presented at the Electronics Computer Technology (ICECT), 2011 3rd International Conference on.

Kunnasranta, M., Hyvärinen, H., Sipilä, T., & Medvedev, N. (2001). Breeding habitat and lair structure of the ringed seal (Phoca hispida ladogensis) in northern Lake Ladoga in Russia. *Polar biology, 24*(3), 171-174.

Kuo, R., & Lin, L. (2010). Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. *Decision Support Systems, 49*(4), 451-462.

Le Boeuf, B. J., Crocker, D., Grayson, J., Gedamke, J., Webb, P. M., Blackwell, S. B., & Costa, D. P. (2000). Respiration and heart rate at the surface between dives in northern elephant seals. *Journal of Experimental Biology, 203*(21), 3265-3274.

Leung, Y.-W., & Wang, Y. (2001). An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 5*(1), 41-53.

Li, Wang, L., & Liu, B. (2008). An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans, 38*(4), 818-831.

Li, T., Shao, G., Zuo, W., & Huang, S. (2017). *Genetic Algorithm for Building Optimization: State-of-the-Art Survey.* Paper presented at the Proceedings of the 9th International Conference on Machine Learning and Computing.

Li, X., Tang, K., Omidvar, M. N., Yang, Z., Qin, K., & China, H. (2013). Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. *gene, 7*(33), 8.

Liang, J., Qu, B., Suganthan, P., & Hernández-Díaz, A. G. (2013). Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212*.

Lichman, M. (2013). UCI machine learning repository [http://archive. ics. uci. edu/ml]. University of California, School of Information and Computer Science. *Irvine, CA*.

Lozano, M., & García-Martínez, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research, 37*(3), 481-497.

Lydersen, C., & Gjertz, I. (1986). Studies of the ringed seal (Phoca hispida Schreber 1775) in its breeding habitat in Kongsfjorden, Svalbard. *Polar Research*, *4*(1), 57-63.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection. *Neurocomputing*.

Marisa, L., de Reyniès, A., Duval, A., Selves, J., Gaub, M. P., Vescovo, L., . . . Ayadi, M. (2013). Gene expression classification of colon cancer into molecular subtypes: characterization, validation, and prognostic value. *PLoS medicine, 10*(5), e1001453.

Medeiros, I. G., Xavier, J. C., & Canuto, A. M. (2015). *Applying the Coral Reefs Optimization algorithm to clustering problems.* Paper presented at the Neural Networks (IJCNN), 2015 International Joint Conference on.

Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: simpler, maybe better. *Evolutionary Computation, IEEE Transactions on, 8*(3), 204-210.

Michalewicz, Z. (2013). *Genetic algorithms+ data structures= evolution programs*: Springer Science & Business Media.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228-249.

Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications, 27*(4), 1053-1073.

Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence, 46*(1), 79-95.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.

Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs.

Mullen, R. J., Monekosso, D., Barman, S., & Remagnino, P. (2009). A review of ant algorithms. *Expert systems with Applications, 36*(6), 9608-9617.

Nanda, S. J., & Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation, 16*, 1-18. doi: http://dx.doi.org/10.1016/j.swevo.2013.11.003

Neumüller, C., Wagner, S., Kronberger, G., & Affenzeller, M. (2012). Parameter meta-optimization of metaheuristic optimization algorithms. *Computer Aided Systems Theory–EUROCAST 2011*, 367-374.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). {UCI} Repository of machine learning databases.

Nolet, B. A., & Mooij, W. M. (2002). Search paths of swans foraging on spatially autocorrelated tubers. *Journal of Animal Ecology, 71*(3), 451-462.

Nolting, B. C. (2013). Random search models of foraging behavior: theory, simulation, and observation.

Nolting, B. C., Hinkelman, T. M., Brassil, C. E., & Tenhumberg, B. (2015). Composite random search strategies based on non-directional sensory cues. *Ecological Complexity, 22*, 126-138.

Nurzaman, S. G., Matsumoto, Y., Nakamura, Y., Shirai, K., Koizumi, S., & Ishiguro, H. (2011). From Lévy to Brownian: a computational model based on biological fluctuation. *PloS one, 6*(2), e16168. doi: 10.1371/journal.pone.0016168

Oaten, A. (1977). Optimal foraging in patches: a case for stochasticity. *Theoretical population biology, 12*(3), 263-285.

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*: Springer Science & Business Media.

Pchelkin, A. (2003). Efficient exploration in reinforcement learning based on utile suffix memory. *Informatica, 14*(2), 237-250.

Pedersen, M. E. H., & Chipperfield, A. J. (2010). Simplifying Particle Swarm Optimization. *Applied Soft Computing Journal, 10*(2), 618-628. doi: 10.1016/j.asoc.2009.08.029

Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2005). The bees algorithm. Technical note. *Manufacturing Engineering Centre, Cardiff University, UK*, 1-57.

Pilfold, N. W., Derocher, A. E., Stirling, I., & Richardson, E. (2014). Polar bear predatory behaviour reveals seascape distribution of ringed seal lairs. *Population Ecology, 56*(1), 129-138.

Pilfold, N. W., Derocher, A. E., Stirling, I., Richardson, E., & Andriashek, D. (2012). Age and sex composition of seals killed by polar bears in the Eastern Beaufort Sea. *PloS one, 7*(7), e41429.

Piotrowski, A. P., Napiorkowski, M. J., Napiorkowski, J. J., & Rowinski, P. M. (2017). Swarm Intelligence and Evolutionary Algorithms: Performance versus speed. *Information Sciences, 384*, 34-85.

Plank, M. J., & James, A. (2008). Optimal foraging: Lévy pattern or process? *Journal of The Royal Society Interface, 5*(26), 1077-1086. doi: 10.1098/rsif.2008.0006

Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied soft computing, 11*(8), 5508-5518.

Rao, R. V., & Waghmare, G. (2017). A new optimization algorithm for solving complex constrained design optimization problems. *Engineering optimization, 49*(1), 60-83.

Rardin, R., & Uzsoy, R. (2001). Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics, 7*(3), 261-304. doi: 10.1023/A:1011319115230

Robert, U. G.-A. (2007). Robert Ringed seal pupping lair, with the pup in the lair and the female approaching the haul-out hole from the water Barnes.

Saha, S., Alok, A., & Ekbal, A. (2015). Use of Semi-supervised Clustering and Feature Selection Techniques for Gene-Expression Data.

Santos, J. M., & Embrechts, M. (2009). *On the use of the adjusted rand index as a metric for evaluating supervised classification.* Paper presented at the International Conference on Artificial Neural Networks.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30-47.

Saremi, S., & Sejnowski, T. J. (2016). Correlated percolation, fractal structures, and scale-invariant distribution of clusters in natural images. *IEEE transactions on pattern analysis and machine intelligence, 38*(5), 1016-1020.

Sasena, M. J., Papalambros, P., & Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization, 34*(3), 263-278.

Senington, R. J. (2013). *Hybrid meta-heuristic frameworks: a functional approach*: University of Leeds.

Senthilnath, J., Das, V., Omkar, S., & Mani, V. (2012). *Clustering Using Levy Flight Cuckoo Search.* Paper presented at the BIC-TA (2).

Senthilnath, J., Das, V., Omkar, S., & Mani, V. (2013). *Clustering using levy flight cuckoo search.* Paper presented at the Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012).

Sheng, Z., Wang, J., Zhou, S., & Zhou, B. (2014). Parameter estimation for chaotic systems using a hybrid adaptive cuckoo search with simulated annealing algorithm. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 24*(1), 013133.

Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009). *A dolphin partner optimization.* Paper presented at the Intelligent Systems, 2009. GCIS'09. WRI Global Congress on.

Sims, D. W., Humphries, N. E., Bradford, R. W., & Bruce, B. D. (2012). Lévy flight and Brownian search patterns of a free-ranging predator reflect different prey field characteristics. *The Journal of animal ecology, 81*(2), 432-442. doi: 10.1111/j.1365-2656.2011.01914.x

Singh, G., & Deep, K. (2015). *Role of Particle Swarm Optimization in Computer Games.* Paper presented at the Proceedings of Fourth International Conference on Soft Computing for Problem Solving.

Slak, A., Tavčar, J., & Duhovnik, J. (2014). Case study analysis and genetic algorithm adaptation for job process planning and scheduling in batch production. *Journal of Design Research 9, 12*(1-2), 52-77.

Smit, S. K., & Eiben, A. E. (2009). *Comparing parameter tuning methods for evolutionary algorithms.* Paper presented at the Evolutionary Computation, 2009. CEC'09. IEEE Congress on.

Soler-Dominguez, A., Juan, A. A., & Kizys, R. (2017). A survey on financial applications of metaheuristics. *ACM Computing Surveys (CSUR), 50*(1), 15.

Sorensen, K., Sevaux, M., & Glover, F. (2017). A history of metaheuristics. *arXiv preprint arXiv:1704.00853*.

Spall, J. C. (2005). *Introduction to stochastic search and optimization: estimation, simulation, and control* (Vol. 65): John Wiley & Sons.

Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: a survey. *Computer, 27*(6), 17-26. doi: 10.1109/2.294849

Talbi, E.-G. (2002). A taxonomy of hybrid metaheuristics. *Journal of heuristics, 8*(5), 541-564.

Tan, Shi, Y., & Niu, B. (2016). *Advances in swarm intelligence*: Springer.

Tan, A. C., & Gilbert, D. (2003). *An empirical comparison of supervised machine learning techniques in bioinformatics.* Paper presented at the Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003-Volume 19.

Tyson, R., Wilson, J., & Lane, W. (2011). Beyond diffusion: Modelling local and long-distance dispersal for organisms exhibiting intensive and extensive search modes. *Theoretical population biology, 79*(3), 70-81.

Van der Merwe, D., & Engelbrecht, A. P. (2003). *Data clustering using particle swarm optimization.* Paper presented at the Evolutionary Computation, 2003. CEC'03. The 2003 Congress on.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research, 60*(3), 611-624.

Vignat, C., & Plastino, A. (2006). Power-law random walks. *Physical Review E, 74*(5), 051124.

Viswanathan. (2011). *The physics of foraging: an introduction to random searches and biological encounters*. Cambridge; New York: Cambridge University Press.

Viswanathan, Afanasyev, V., Buldyrev, S. V., Havlin, S., da Luz, M. G. E., Raposo, E. P., & Stanley, H. E. (2000). Lévy flights in random searches. *Physica A: Statistical Mechanics and its Applications, 282*(1), 1-12. doi: 10.1016/S0378-4371(00)00071-6

Viswanathan, Buldyrev, S. V., Havlin, S., da Luz, M. G. E., Raposo, E. P., & Stanley, H. E. (1999). Optimizing the success of random searches. *Nature, 401*(6756), 911-914. doi: Doi 10.1038/44831

Viswanathan, Raposo, E., & Da Luz, M. (2008). Lévy flights and superdiffusion in the context of biological encounters and random searches. *Physics of Life Reviews, 5*(3), 133-150.

Viswanathan, G. M., Da Luz, M. G., Raposo, E. P., & Stanley, H. E. (2011). *The physics of foraging: an introduction to random searches and biological encounters*: Cambridge University Press.

Wang, Y., & Yang, Y. (2009). Particle swarm optimization with preference order ranking for multi-objective optimization. *Information Sciences, 179*(12), 1944-1959.

Weimerskirch, H., Pinaud, D., Pawlowski, F., & Bost, C.-A. (2007). Does prey capture induce area-restricted search? A fine-scale study using GPS in a marine predator, the wandering albatross. *The American Naturalist, 170*(5), 734-743.

Williams, M. T., Nations, C. S., Smith, T. G., Moulton, V. D., & Perham, C. J. (2006). Ringed seal (Phoca hispida) use of subnivean structures in the Alaskan Beaufort Sea during development of an oil production facility. *Aquatic Mammals, 32*(3), 311-324.

Xu, J., & Zhang, J. (2014). *Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis.* Paper presented at the Control Conference (CCC), 2014 33rd Chinese.

Yanagida, T., Ueda, M., Murata, T., Esaki, S., & Ishii, Y. (2007). Brownian motion, fluctuation and life. *Biosystems, 88*(3), 228-242.

Yang. (2010a). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation, 2*(2), 78-84.

Yang. (2010b). *Nature-inspired metaheuristic algorithms*: IEEE.

Yang. (2010c). A new metaheuristic bat-inspired algorithm *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74): Springer.

Yang. (2011). Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation, 3*(2), 77-84.

Yang. (2012). Flower pollination algorithm for global optimization *Unconventional Computation and Natural Computation* (pp. 240-249): Springer.

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

Yang, & Deb. (2009). *Cuckoo search via Lévy flights.* Paper presented at the Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.

Yang, & Deb, S. (2014). Cuckoo search: recent advances and applications. *Neural Computing and Applications, 24*(1), 169-174. doi: 10.1007/s00521-013-1367-1

Yang, Deb, S., & Fong, S. (2014). Metaheuristic algorithms: optimal balance of intensification and diversification. *Applied Mathematics & Information Sciences, 8*(3), 977.

Yang, X.-S., & Hossein. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations, 29*(5), 464-483.

Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of computational design and engineering, 3*(1), 24-36.

Yen, G., Yang, F., Hickey, T., & Goldstein, M. (2001). *Coordination of exploration and exploitation in a dynamic environment.* Paper presented at the Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on.

Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S.-H., & Patton, R. M. (2015). *Optimizing deep learning hyper-parameters through an evolutionary algorithm.* Paper presented at the Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments.

LIST OF PUBLICATIONS

1. **Saadi, Younes**, Iwan Tri Riyadi Yanto, Tutut Herawan, Vimala Balakrishnan, Haruna Chiroma, and Anhar Risnumawan. "Ringed Seal Search for Global Optimization via a Sensitive Search Model." *PloS one 11, no. 1 (2016): e0144371*. ***Published.***
2. Chiroma, Haruna, **Younes Saadi,** Abdullah Khan, Adamu I. Abubakar, Mukhtar F. Hamza, Liyana Shuib, Abdulsalam Y. Gital, Iztok Fister Jr, and Tutut Herawan. "Bio-Inspired Computation: Recent Development on the Modifications of the Cuckoo Search Algorithm." *Applied Soft Computing (2017).* ***Accepted.***
3. Chiroma, Haruna, Abdullah Khan, Adamu I. Abubakar, **Younes Saadi**, Mukhtar F. Hamza, Liyana Shuib, Abdulsalam Y. Gital, and Tutut Herawan. "A new approach for forecasting OPEC petroleum consumption based on neural network train by using flower pollination algorithm." *Applied Soft Computing 48 (2016): 50-58.* ***Published.***
4. **Saadi, Younes**, Iwan Tri Riyadi Yanto, Tutut Herawan, Vimala Balakrishnan, Haruna Chiroma, and Abdullah Khan. "Ringed Seal Search for Data Clustering." *Intelligent Automation & Soft Computing (2017):* ***Under review.***
5. **Saadi, Younes**, Iwan Tri Riyadi Yanto, Tutut Herawan, Vimala Balakrishnan and Haruna Chiroma. "Exploitation-Exploration Balance in Nature-Inspired Algorithms: A Review". *Complexity, Wiley (2017):* ***Under review.***
6. Yanto, Iwan Tri Riyadi, **Younes Saadi**, Dedy Hartama, Dewi Pramudi Ismi, and Andri Pranolo. "A framework of fuzzy partition based on Artificial Bee Colony for categorical data clustering." *ICSITech International Conference*, pp. 260-263. IEEE, 2016. ***Published.***

7. Chiroma, Haruna, Sameem Abdulkareem, Sanah Abdullahi Muaz, Adamu I. Abubakar, Edi Sutoyo, Mungad Mungad, **Younes Saadi**, Eka Novita Sari, and Tutut Herawan. "An intelligent modeling of oil consumption." *Advances in Intelligent Informatics*, pp. 557-568. Springer, Cham, 2015. ***Published.***