ENHANCED HARRIS'S HAWK ALGORITHM FOR CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

SHAYMAH AKRAM YASEAR

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2020

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

SHAYMAH AKRAM YASEAR

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"ENHANCED HARRIS'S HAWK ALGORITHM FOR CONTINUOUS MULTI-OBJECTIVE OPTIMIZATION PROBLEMS"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***02 Jun 2020.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: ***June 02, 2020.***

Pengerusi Viva: *(Chairman for VIVA)*	**Prof. Dr. Zulkhairi Md Dahalin**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Masri Ayob**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Dr. Siti Sakira Kamaruddin**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Ku Ruhana Ku Mahamud**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Dr. Mustafa Muwafak Theab Alobaedy**	Tandatangan *(Signature)*

Tarikh:
(Date) ***June 02, 2020***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Metaheuristik berasaskan kecerdasan kawanan berbilang objektif (MOSI) dicadangkan untuk menyelesaikan masalah pengoptimuman berbilang objektif (MOP) yang mempunyai objektif bercanggah. Algoritma berbilang objektif pengoptimuman Harris hawk (HHMO) adalah algoritma berasaskan MOSI yang dibangunkan berasaskan pendekatan titik rujukan. Titik rujukan ditentukan oleh pembuat keputusan untuk memandu proses pencarian ke kawasan tertentu di *Pareto front* yang sebenar. Walau bagaimanapun, algoritma HHMO menghasilkan penghampiran yang kurang baik untuk *Pareto front* kerana kekurangan perkongsian maklumat dalam strategi kemaskini populasinya, pembahagian parameter penumpuan yang sama rata dan populasi awal yang dihasilkan secara rawak. Algoritma dua langkah menyusun tidak dominasi HHMO yang dipertingkatkan (2S-ENDSHHMO) telah dicadangkan untuk menyelesaikan masalah ini. Algoritma tersebut merangkumi (i) strategi kemas kini penduduk yang meningkatkan pergerakan helang di ruang pencarian, (ii) strategi penyesuaian parameter untuk mengawal peralihan antara penerokaan dan eksploitasi, dan (iii) kaedah menjana populasi dalam menghasilkan penyelesaian calon awal. Strategi kemas kini penduduk menghitung kedudukan baru helang berasaskan teknik *flush-and-ambush* helang Harris, dan memilih helang terbaik berasaskan pendekatan menyususn yang tidak dominasi. Strategi penyesuaian membolehkan parameter berubah secara adaptif berasaskan keadaan ruang carian. Populasi awal dihasilkan dengan menjana nombor kuasi-rawak menggunakan *R-sequence* diikuti dengan menyesuaikan konsep pembelajaran berasaskan pembangkang separa untuk meningkatkan kepelbagaian separuh terburuk dalam populasi helang. Prestasi 2S-ENDSHHMO telah diuji menggunakan 12 MOP dan tiga MOP kejuruteraan. Keputusan yang diperolehi dibandingkan dengan keputusan dari lapan algoritma pengoptimuman berbilang objektif yang terkini. Algoritma 2S-ENDSHHMO dapat menghasilkan penyelesaian yang tidak dikuasai dengan penumpuan dan kepelbagaian yang lebih besar dalam menyelesaikan kebanyakan MOP dan mempamerkan kemampuan yang hebat dalam melompat keluar dari optima tempatan. Ini menunjukkan kemampuan algoritma dalam meneroka ruang carian. Algoritma 2S-ENDSHHMO dapat digunakan untuk meningkatkan proses pencarian algoritma berasaskan MOSI yang lain dan dapat diterapkan untuk menyelesaikan MOP dalam aplikasi seperti reka bentuk struktur dan pemprosesan isyarat.

Kata kunci: Metaheuristik, Kepintaran kawanan, Eksplorasi dan eksplotasi, Pendekatan berasaskan keutamaan, *Pareto front*.

Abstract

Multi-objective swarm intelligence-based (MOSI-based) metaheuristics were proposed to solve multi-objective optimization problems (MOPs) with conflicting objectives. Harris's hawk multi-objective optimizer (HHMO) algorithm is a MOSI-based algorithm that was developed based on the reference point approach. The reference point is determined by the decision maker to guide the search process to a particular region in the true Pareto front. However, HHMO algorithm produces a poor approximation to the Pareto front because lack of information sharing in its population update strategy, equal division of convergence parameter and randomly generated initial population. A two-step enhanced non-dominated sorting HHMO (2S-ENDSHHMO) algorithm has been proposed to solve this problem. The algorithm includes (i) a population update strategy which improves the movement of hawks in the search space, (ii) a parameter adjusting strategy to control the transition between exploration and exploitation, and (iii) a population generating method in producing the initial candidate solutions. The population update strategy calculates a new position of hawks based on the flush-and-ambush technique of Harris's hawks, and selects the best hawks based on the non-dominated sorting approach. The adjustment strategy enables the parameter to adaptively changed based on the state of the search space. The initial population is produced by generating quasi-random numbers using R-sequence followed by adapting the partial opposition-based learning concept to improve the diversity of the worst half in the population of hawks. The performance of the 2S-ENDSHHMO has been evaluated using 12 MOPs and three engineering MOPs. The obtained results were compared with the results of eight state-of-the-art multi-objective optimization algorithms. The 2S-ENDSHHMO algorithm was able to generate non-dominated solutions with greater convergence and diversity in solving most MOPs and showed a great ability in jumping out of local optima. This indicates the capability of the algorithm in exploring the search space. The 2S-ENDSHHMO algorithm can be used to improve the search process of other MOSI-based algorithms and can be applied to solve MOPs in applications such as structural design and signal processing.

Keywords: Metaheuristic, Swarm intelligence, Exploration and exploitation, Preference-based approach, *Pareto front.*

Acknowledgement

In the name of Allah, the Most Gracious and the Most Merciful.

Alhamdulillah, all praises to Allah for the strengths in completing this thesis. Special appreciation goes to my supervisor, Prof. Dr. Ku Ruhana Ku Mahamud, for her supervision and ongoing support. Her invaluable help of constructive comments and suggestions throughout the experimental and thesis works have contributed to the success of this research. Many thanks and appreciation to my co-supervisor, Dr. Mustafa Muwafak Alobaedy, for his support.

My acknowledgment also goes to all the technicians and staff members of School of Computing for their cooperation.

Sincere thanks to all my friends for their kindness and moral support during my study.

Last but not least, my deepest gratitude goes to my beloved parents and sister for their endless love, prayers and encouragement.

To those who indirectly contributed to this research, your kindness means a lot to me. Thank you very much.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ABC	Artificial bee colony
BA	Bat algorithm
DM	Decision maker
EA	Evolutionary algorithm
EHHMO	Enhanced Harris's hawk multi-objective optimizer
2S-ENDSHHMO	Enhanced non-dominated sorting Harris's hawk multi-objective optimizer
ES	Evolutionary strategy
FA	Firefly algorithm
GA	Genetic algorithm
GOA	Grasshopper optimization algorithm
GRA	Grey relational analysis
GRC	Grey relational coefficient
GRG	Grey relational grades
GSA	Gravitational search algorithm
GWO	Grey wolf optimizer
HHMO	Harris's hawk multi-objective optimizer
HHO	Harris's hawk optimization
HV	Hypervolume
IGD	Inverted generational distance
LHS	Latin hypercube sampling
MOABC	Multi-objective artificial bee colony
MOBA	Multi-objective bat algorithm
MODA	Multi-objective dragonfly
MOFA	Multi-objective firefly algorithm
MOGOA	Multi-objective grasshopper optimization algorithm
MOGSA	Multi-objective gravitational search algorithm
MOGWO	Multi-objective grey wolf optimizer
MOO	Multi-objective optimization
MOP	Multi-objective optimization problem
MOPSO	Multi-objective particle swarm optimization
MOSI	Multi-objective swarm intelligence-based
MOSSA	Multi-objective salp swarm algorithm
MOWOA	Multi-objective whale optimization algorithm
NDSHHMO	Non-dominated sorting Harris's hawk multi-objective optimizer
NSABC	Non-dominated sorting artificial bee colony

NSGA-II	Non-dominated sorting genetic algorithm
NSGSA	Dominated sorting gravitational search algorithm
OBL	Opposition-based learning
OPF	Optimal power flow
PD	Pure diversity
PSO	Particle swarm optimization
QRNG	Quasi-random generators
RNG	Random number generator
SD	Standard deviation
SI	Swarm intelligence
SOP	Single-objective problem

List of Appendices

CHAPTER ONE
INTRODUCTION

Researchers in several scientific and technical fields often face problems of increasing complexity. Most of these problems are NP-Completes (a non-polynomial problem), in which the search space for solutions grows exponentially with the dimensions of the problem. These problems can be formulated as an optimization problem. Solving an optimization problem consists of finding a set of possible combinations for decision variables (parameters) that provides the best possible performance for a system. The best solution is determined according to the evaluation criteria given by an objective function (Andréasson, Evgrafov, & Patriksson, 2005; Talbi, 2009).

Based on the type of decision variables in the search space, the optimization problem can be classified as continuous, discrete (integer), or mixed (Talbi, 2009). To solve an optimization problem, a wide range of optimization methods have been proposed (Kheiri, 2018; Li & Zheng, 2017; Pardo, Möller, Neunert, Winkler, & Buchli, 2016). These methods can be classified into two categories: exact and approximate methods (Talbi, 2009).

The exact methods guarantee obtaining the optimal solution; however, they require a very long computation time, which exponentially increases with the size of the problem (Kuo & Zulvia, 2015; Lu et al., 2020). Furthermore, the scope of these methods is limited to certain types of problems. For example, linear programming is only applicable for the optimization of continuous problems whose cost function and

constraints are linear in nature (Momoh, 2015). The application of a simplified linear model for solving a nonlinear problem gives optimal results for the model but, often, does not reflect the real problem. The well-known Branch & Bound (B&B) method (Land & Doig, 1960) depends entirely on the type of optimization problem (Torbaghan & Gibescu, 2017). Furthermore, the algorithm cannot find the global optimum within a reasonable time. These are two examples of recognized conventional optimization methods that can only be used for certain types of problems. In addition, they are difficult to adapt from one problem to another; so, their development can be long and laborious. There is, therefore, a large family of problems difficult to optimize via exact methods. In this case, the use of methods that offer a good quality solution in a reasonable time, are required.

Various approximate methods have been proposed to overcome the limitations of exact methods. In contrast to conventional exact optimization algorithms, the approximate methods, without guaranteeing the optimum, can provide a feasible solution in a reasonable time (Lu et al., 2020). The approximate solutions can be achieved by using heuristics or metaheuristics methods. The heuristic methods require knowledge of the problem and are designed for a particular problem (Sörensen, Sevaux, & Glover, 2018). Therefore, these methods are not general enough to be effective. On the other hand, metaheuristics are general optimization methods applicable to solve different optimization problems (Sörensen et al., 2018; Stojanović, Brajević, Stanimirović, Kazakovtsev, & Zdravev, 2017). In contrast to exact and heuristic methods, metaheuristics apply a stochastic approach to find a feasible solution among randomly

generated variables. The goal is to escape from local optima (minima or maxima) to find better solutions for an optimization problem, where the number of local optima exponentially increase by increasing the dimensions, of an optimization problem.

Metaheuristics are simple to implement practically and they have proven their efficiency in solving optimization problems in different fields such as, operations research (Li et al., 2020), engineering (Dede, Grzywiński, & Venkata Rao, 2020; Sayed, Darwish, & Hassanien, 2018) and healthcare (Tsai, Chiang, Ksentini, & Chen, 2016). These algorithms are very flexible, and they have the ability to deal with problems with objective functions of different properties. The strong point of metaheuristics is that it does not require a detailed knowledge of the problem, one can represent it by a black-box carrying entries (the variables) and outputs (according to the objective functions) (Talbi, 2009; Tamura & Gallagher, 2019). This only requires manipulation of the inputs, reading the outputs, and manipulating the inputs again in order to improve the outputs.

Based on the number of solutions handled in the search space, metaheuristics can be classified into a single-solution (trajectory)-based and population-based metaheuristics (Blum & Roli, 2003a). Single-solution metaheuristics begin with a single initial solution and the search moves from a solution to a neighbouring solution, constructing a trajectory in the search space. Single solution-based methods include, essentially, the simulated annealing algorithm (Kirkpatrick, Gelatt, & Vecchi, 1983), tabu search (Glover, 1989), greedy randomized adaptive search procedure (Feo & Resende, 1989),

variable neighbourhood search (Hansen & Mladenović, 2018), iterated local search (Lourenço, Martin, & Stützle, 2019), and their variants .

Contrary to the single-solution-based algorithms, population-based algorithms start with a set of potential solutions, which improve with iterations. Under this category, there are two main classes, namely, evolutionary algorithms (EAs) which are derived from evaluation by natural selection theory (Darwin, 1859), such as genetic algorithms (GAs) (Holland, 1973) and swarm intelligence (SI) algorithms which are developed based on the SI theory (Beni & Wang, 1993). The algorithms under SI class are composed of simple agents (individuals), which interact with each other and with their environment following simple rules. In general, SI algorithms, in the same way as EAs, come from mimicking the biological or physical phenomena, so-called nature-inspired metaheuristics (Boussaïd, Lepagnot, & Siarry, 2013). Several SI-based metaheuristics have been proposed and they have shown superior skills in solving various optimization problems, such as, ant colony optimization (Dorigo, 1992), particle swarm optimization (PSO) (Kennedy & Eberhar, 1995), firefly algorithm (FA) (Yang, 2009), bat algorithm (BA) (Yang, 2010c), artificial bee colony (ABC) (Karaboga, 2005), and Grey wolf optimizer (GWO) (Mirjalili, Mirjalili, & Lewis, 2014).

The SI-based metaheuristics have been proposed to deal with single-objective problems (SOPs), where the goal is minimizing/maximizing a single criterion (objective), such as cost minimization. However, in real-world applications, usually

optimization problems tend to integrate several simultaneous criteria. This requires, finding a compromise. In this case, the optimal solution is no longer a single solution, but a set of non-dominated solutions. The problem with multiple objectives is known as the multi-objective optimization problem (MOP). Solving a MOP commonly called a multi-objective optimization (MOO), is to calculate a set of non-dominated solutions, known as the Pareto front.

To solve a MOP, several multi-objective SI-based (MOSI-based) metaheuristics have been proposed. In general, the MOSI-based algorithm extends a single objective optimization algorithm to solve MOPs. Most of the MOSI-based algorithms integrate a single objective optimization algorithm with a MOO approach to handle multiple objectives, such as MOPSO Coello and Lechuga (2002), Akbari, Hedayatzadeh, Ziarati, and Hassanizadeh (2012), Yang (2013), Yang (2012a) and Mirjalili, Saremi, Mirjalili, and Coelho (2016).

Metaheuristics, in principle, are based on the notion of exploitation and exploration. Exploration refers to the process of examining different regions in the search space and finding better potential solutions (Blum & Roli, 2003b). On the other hand, exploitation is the process of improving the best solutions found during the exploration process, with the aim of finding higher quality solutions (Blum & Roli, 2003b). One of the challenges in the implementation of metaheuristics is finding a balanced trade-off between exploration and exploitation which, otherwise, leads to poor convergence (Blum & Roli, 2003b).

In the population-based metaheuristics, there are three components that affect the convergence, exploration, and exploitation of an optimization algorithm. These components are generation of a new population of solutions, parameters configuration (Huang, Li, & Yao, 2019) and initial population of candidate solutions (Poles, Fu, & Rigoni, 2009; Talbi, 2013; Tu, Chen, & Liu, 2019).

The population updated strategy plays an important role in determining the quality of next generation (new candidate solutions). An inefficient positions update strategy will lead to the algorithm falling into local optima with a loss of population diversity, which leads to poor convergence toward the Pareto front (Tu et al., 2019).

In metaheuristics, parameter configuration directly affects the processes of exploring and exploiting the search space, which determines the quality of the final solution. Therefore, it is necessary to understand the most promising configuration that should be used (Eiben, Michalewicz, Schoenauer, & Smith, 2007; Karafotias, Hoogendoorn, & Eiben, 2015; Parpinelli, Plichoski, Da Silva, & Narloch, 2019). Improper parameters create the risk of being easily trapped in local optima, due to loss of population diversity, which leads to premature convergence (Barbosa & Senne, 2017; Emary, Zawbaa, & Grosan, 2017; Yan, Xu, & Yun, 2019; Yang, Deb, Hanne, & He, 2019).

The generation of the initial population also has an impact on the quality of the final solutions, in terms of convergence and diversity (Tu et al., 2019). If the initial

population is not well distributed in the search space, this will lead to a loss of population diversity and poor convergence toward the Pareto front (Tu et al., 2019).

Furthermore, the ultima goal of MOO algorithms is to help the decision maker (DM) to find the most satisfactory solutions, rather than all Pareto optimal solutions (Li, Chen, Min, & Yao, 2018a; Thiele, Miettinen, Korhonen, & Molina, 2009; Yang, Li, Deutz, Back, & Emmerich, 2016). However, most MOSI-based optimization algorithms try to approximate the whole Pareto front and return a set of non-dominated solutions, which are evenly distributed across the whole Pareto front. In real situations, different regions of the Pareto front could be more preferred than others, and some regions could not be interesting at all. Therefore, the main drawbacks of the approach used by these algorithms are spending time in exploring undesired solutions and difficulties for the DM in determining the most preferred solution among huge numbers of non-dominated solutions (Li et al., 2018a). To overcome these limitations, the preference of DM is combined with the search process, to guide the search to the region of greatest interest to the DM. This helps to improve optimization efficiency and reduce computational cost (Li et al., 2018a; Thiele et al., 2009; Yang et al., 2016).

1.1 Problem Statement

Several reference point-based MOSI-based algorithms have been proposed by integrating the preference information, represented by reference points with the search process (Allmendinger, Li, & Branke, 2008; Carrese & Li, 2015; DeBruyne & Kaur, 2016; Wickramasinghe & Li, 2008). The reference point is determined by the DM and

are used to guide the search process toward a preferred region (Li et al., 2018a; Thiele et al., 2009; Yang et al., 2016). Therefore, these algorithms inherit the advantages of preference approach in terms of effectively finding the most satisfactory solutions and this can help in reducing the computational cost (Liu, Wang, Feng, Huang, & Jiao, 2016; Mohammadi, Omidvar, & Li, 2012).

To the best of the author's knowledge, the current reference-point-based algorithms extended the standard PSO algorithm (Kennedy & Eberhar, 1995) to handle multiple objectives (Allmendinger et al., 2008; Carrese & Li, 2015; DeBruyne & Kaur, 2016; Wickramasinghe & Li, 2008). Furthermore, the reference point-based MOPSO algorithms inherit the disadvantages of PSO algorithm in terms of falling into local optima due to loss of population diversity which leads to premature convergence (Asih, Sopha, & Kriptaniadewa, 2017; Neumann, Gao, Doerr, Neumann, & Wagner, 2018; Ni & Deng, 2014; Xu, Wu, & Jiang, 2015).

In DeBruyne and Kaur (2016) another reference point-based multi-objective optimization algorithm has been proposed, which is called Harris's hawk multi-objective optimizer (HHMO). This algorithm is inspired by the hunting behaviour of the Harris's hawk in nature (Bednarz, 1988; Dawson, 1988). This algorithm has a few parameters and it inherits the advantages of preference-based approaches in terms of effectively finding the most satisfactory solutions and this can help in reducing the computational cost (DeBruyne & Kaur, 2016).

The HHMO algorithm was developed based on the mathematical model of the MOGWO and GWO algorithm (DeBruyne & Kaur, 2016). Therefore, it inherits its advantages and disadvantages. The GWO has several parameters, easy to use, flexible, scalable, and has a special capability to escape out of local optima (Faris, Aljarah, Al-Betar, & Mirjalili, 2018). The MOGWO has been successfully applied to solve different MOPs (Mirjalili & Dong, 2020) and has showed superior performance compare to other well-known MOSI-based algorithms, such as, MOPSO (Ni, Wang, Tang, & Wei, 2019; Xia et al., 2019; Zhao et al., 2020). However, in the MOGWO, the obtained solutions are evenly distributed across the Pareto front. Therefore, the HHMO can be considered as an improved version of MOGWO, in which it guides the search process toward a preferred region (Jakubovski Filho, Ferreira, & Vergilio, 2019; Li et al., 2018a). Based on that, the HHMO algorithm (DeBruyne & Kaur, 2016) is of special interest for this work.

The main problem that effects the performance of the MOSI-based metaheristics is the premature convergence, which caused by the improper balance between exploration and exploitation processes (Aguiar e Oliveira Junior, 2016). In general, this problem arises when every individual in the population is located at a sub-optimal area of the search space from which they cannot escape (Bhattacharya, 2016). After a local optimum is found, all individuals in the region that are attracted to it lose interest in optimization (Blum & Roli, 2003a). In other words, if the population is too similar, all individuals will be trapped at a Pareto local optimum set, which leads to premature

convergence (Cheng, Chen, Fleming, Yang, & Gan, 2017; Hancer, Xue, Zhang, Karaboga, & Akay, 2018; Khan & Li, 2017; Zhang, Zheng, Cheng, Qiu, & Jin, 2018).

Some of the studies on the MOSI-based algorithms, overcame the problem of premature convergence by proposing different position update and parameter adjustment strategies (Chen, Qian, Zhang, & Sun, 2019; Wei, Li, Fan, Sun, & Hu, 2018; Wei, Li, & Fan, 2019; Yang & Ji, 2016; Yu, Wang, & Xiao, 2020). However, these studies overlook the impact of initial population on the performance of the algorithm. They used the traditional RNG method to initialize the population of candidate solutions, which leads to the premature convergence problem.

Other studies overcame the problem of premature convergence, by proposing parameter adjustment strategies to control the exploration and exploitation and prevent premature convergence (Mohamed, El-Gaafary, Mohamed, & Hemeida, 2016; Zellagui, Hassan, & Abdelaziz, 2017). However, these studies used same position update strategies of the standard single objective optimization algorithm and traditional RNG method to initialize the population of solutions.

A number of studies have tried to prevent the premature convergence by improving the position update of strategy of the algorithm (Du, Wang, Hao, Niu, & Yang, 2019; Liu, Li, Kong, & Huang, 2019; Lv, Zhao, Wang, & Fan, 2019 ; Mellal & Zio, 2019). However, these studies did not take in consideration the parameters configuration and used RNG method to initialize the population.

Most of the MOSI-based algorithm used the traditional random number generator (RNG) to initialize the population of candidate solutions (Chen et al., 2019; DeBruyne & Kaur, 2016; J. Liu et al., 2019; Lv et al., 2019; Wei et al., 2018; Wei et al., 2019; Yang & Ji, 2016; Yu et al., 2020). However, with this method it is difficult to ensure the diversity of the initial population (Jana, Das, & Sil, 2018), which affects the search process of the algorithm to a certain extent (Altinoz, Yilmaz, & Weber, 2014) and leads to premature convergence (Digehsara, Chegini, Bagheri, & Roknsaraei, 2020; Jana et al., 2018; Poles et al., 2009).

Although HHMO algorithm inherits the advantages of preference approach and GWO algorithm, it has limitations that degrade its performance, especially in solving complex MOPs, resulting in poor approximation for the Pareto front of a MOP. In the HHMO algorithm, the positions of hawk represent candidate solutions for a MOPs. The new position of hawks is updated based on the average positions of the three leaders (DeBruyne & Kaur, 2016). These leaders represent the first, second and third best solutions in the search space and the positions of other hawks are not considered in updating the position of hawks in the search space (Guo, Wang, Zhu, Guo, & Xie, 2020; Long, Jiao, Liang, & Tang, 2018; Long et al., 2019). This indicates a lack of sharing the information between the hawks in the population.

In this case, the HHMO algorithm will not be able to escape from local optima of the Pareto front, due to the loss of population diversity, especially, in solving complex MOPs (Guo et al., 2020; Long et al., 2018; Long et al., 2019). This, in turn, leads to

poor convergence toward the Pareto front (Abdou & Bloch, 2020; Khan & Li, 2017; Niyomubyeyi, Sicuaio, González, Pilesjö, & Mansourian, 2020). Sharing the information by utilizing the experiences of all hawks during the search process is very important to accelerate and ensure the convergence and the diversity of the obtained solutions (Akbari & Ziarati, 2011; Wang & Tan, 2017). The cooperation between the individuals during the searching process is one of the main concepts of the SI system (Beni & Wang, 1993). In this context, this study aims to improve the population update strategy of hawks, by proposing a new population update strategy, which takes into consideration the contribution of all hawks in updating the population.

In the HHMO algorithm, half of the iterations are allocated for the exploration process while the other half deal with the exploitation (Aljarah, Mafarja, Heidari, Faris, & Mirjalili, 2020; Dadhich, Sharma, & Sharma, 2017; Dewangan, Shukla, & Godfrey, 2019; Joshi & Arora, 2017), overlooking the impact of the right balance between them (Joshi & Arora, 2017; Majeed & Patri, 2018; Padhy, Panda, & Mahapatra, 2017). This transition between exploration and exploitation is controlled by adjusting the convergence parameter (DeBruyne & Kaur, 2016). This parameter decreases linearly with the number of iterations (DeBruyne & Kaur, 2016).

In metaheuristics, the optimization process starts by exploring the search space to find more promising regions (Allawi, Ibraheem, & Humaidi, 2019; Moghaddam, Moghaddam, & Cheriet, 2012). Then, in the later stage of the optimization, the algorithm should be focused more on a particular region to find the best solutions

(Sahib, Abdulnabi, & Mohammed, 2018). In solving practical MOPs, the requirement for the ability to explore the algorithm is high due to the complexity of the search process (Yang, Deb, & Fong, 2014a). However, in the current adjustment strategy of the convergence parameter in HHMO algorithm, the exploration and exploitation are equally performed (Dewangan et al., 2019; Majeed & Patri, 2018), which is difficult to adapt to reflect the actual optimization process (Khanum et al., 2019; Long et al., 2018; Padhy et al., 2017). Therefore, it is not easy to reach the Pareto optimal solutions when solving an optimization problem (Barbosa & Senne, 2017; Yan et al., 2019; X.-S. Yang et al., 2019).

If the HHMO algorithm does not spend enough time on the exploration, it will fall into local optima because it will not be able to explore the search space for other promising regions to attainment the global optima (Hussain, Salleh, Cheng, & Shi, 2019; Sahib et al., 2018; Wang & Li, 2019). On the other hand, too much exploration will lead to a decrease in its efficiency and slow convergence toward the global optima solution (Hussain et al., 2019; Sahib et al., 2018; Wang & Li, 2019). Moreover, if the exploitation power becomes strengthened the population of the hawks loses its diversity (Hussain et al., 2019; Sahib et al., 2018; Wang & Li, 2019). Focusing increasingly on the already discovered regions in the search space will lead to population diversity loss and attainment of a homogeneous state (Vasuki, 2020; Wang & Li, 2019). This, in turn, will lead to premature convergence (Jia et al., 2019; Neumann et al., 2018; Ni & Deng, 2014). To avoid premature convergence and improve the convergence of the HHMO algorithm, the parameter adjustment strategy

of the convergence parameter needs to be improved (Barbosa & Senne, 2017; Long et al., 2018; Yan et al., 2019; X.-S. Yang et al., 2019). This can be achieved by integrating an effective adjustment strategy for the convergence parameter with the HHMO algorithm (Barbosa & Senne, 2017; Emary et al., 2017; Isiet & Gadala, 2019; Long et al., 2018; Yan et al., 2019; X.-S. Yang et al., 2019). This aims to coordinate the proportional relationship between the exploration and exploitation of the search space during the optimization process.

In the HHMO algorithm, the initial population of hawks is generated using a RNG method, within lower and upper bounds representing the limits for each decision variable in the search space (Talbi, 2009). The RNG methods produce points with the same probability on equal subintervals. These points are clustered in some regions and leave gaps in others (Digehsara et al., 2020; Jana et al., 2018; Maaranen, Miettinen, & Mäkelä, 2004). With this method, it is difficult to ensure the diversity of the initial population, which affects the search process of the algorithm to a certain extent (Altinoz et al., 2014). Since there is no prior knowledge of the global optimal solution to the optimization problem, the population of hawks should be distributed as evenly as possible in the search space (Altinoz et al., 2014; Kazemzadeh Azad, 2018). If an initial population is well generated, this will improve the convergence toward true Pareto front (Tu et al., 2019). Therefore, to improve the performance of the HHMO, it is important to ensure good distribution of the initial potential solutions (Talbi, 2013).

1.2 Research Questions

This research aims to develope and enhance the Harris's hawk multi-objective optimizer by answering the following questions:

i. How can the population update strategy of hawks be improved in the search space?

ii. How can the transition between the exploration and exploitation in the Harris's hawk multi-Objective optimizer be improved?

iii How can the distribution of initial population in the Harris's hawk multi-objective optimizer be improved?

iv Can the enhanced Harris's hawk multi-objective optimizer be used effectively to solve a MOO problem in the continuous domain?

1.3 Research Objectives

The main objective is to develop an enhanced Harris's hawk multi-objective optimizer algorithm for the continuous optimization problem. Specific objectives are:

i. To propose a population update strategy, which improves the movement of hawks in the search space.

ii. To propose a parameter adjusting strategy to control the transition between exploration and exploitation in the Harris's hawk multi-objective optimizer algorithm.

iii. To propose a population generator method to be used in generating the initial candidate solutions in the search space.

iv. To evaluate the performance of the enhanced Harris's hawk multi-objective optimizer algorithm.

1.4 Significance of the Research

This study is conducted to overcome the drawbacks of the HHMO algorithm by proposing a new population update strategy, parameter adjustment strategy and initial population generator method. The concept of the proposed population update and parameter adjustment strategies can be used to improve the search process of other MOSI-based algorithms. The initial population generator method can be used with other MOSI-based algorithm to initialize the population of candidate solutions.

The proposed enhanced HHMO algorithm can be applied to solve MOPs in various applications. These problems include, but are not restricted to, structural design optimization, tuning the proportional-integral-derivative (PID) controller, signal processing, communication and networking, power generation and controlling and biomedical problems.

1.5 Scope and Limitations of the Research

This study was conducted to enhance the HHMO algorithm, which is considered as a MOSI-based algorithm. Therefore, this study focuses on MOSI-based optimization algorithms and omits other algorithms under the EA class. Initially, the HHMO is developed to solve MOPs with continuous search space. Therefore, MOPs with discrete and mixed integer decision variables are not considered in this study.

The proposed enhanced HHMO algorithm has been evaluated using two different approaches. The first approach is by using well-known test problems. These problems have certain characteristics that provide a good basis for testing the performance of MOO metaheuristics. The second approach is evaluating the performance of the enhanced HHMO algorithm by solving different engineering MOPs.

Solving engineering MOPs requires a method to deal with the constraints of a problem. In this research, the penalty method is utilized to handle the constraints. Several constraints handling methods were proposed in the literature. However, most of these studies used the penalty method to solve constraint optimization problems. Finding out which one is the best is out of the scope of this study. The performance of the proposed enhanced HHMO algorithms were evaluated against other well-known algorithms, namely, HHMO, MOGWO (Mirjalili et al., 2016), multi-objective grasshopper optimization algorithm (MOGOA) (Mirjalili, Mirjalili, Saremi, Faris, & Aljarah, 2018), multi-objective dragonfly algorithm (MODA) (Mirjalili, 2016), multi-objective salp swarm algorithm (MOSSA) (Mirjalili et al., 2017), non-dominated sorting based multi-objective artificial bee colony (NSABC) algorithm (Kishor, Singh, & Prakash, 2016), non-dominated sorting gravitational search algorithm (NSGSA) (Zellagui et al., 2017) and reference point-based non-dominated sorting genetic algorithm (R-NSGA-II) (Deb & Sundar, 2006).

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter Two highlights the concept of optimization and the methods used to solve it and outlines the concept of SI and the state-of-the-art SI-based optimization algorithms. The chapter focuses on the MOO and the main concepts related to preference-based algorithms. The Harris's hawk MOO algorithm is described in this chapter together with the parameter adjustment strategies and the initial population generating methods. Chapter Three describes the methodology used to conduct this research. The high level of research framework is presented followed by describing the proposed population update and parameter adjustment strategies and two-step initial population generating method. This chapter includes descriptions of the multi-objective test problems and engineering applications used in the evaluation followed by the performance metrics. Chapter Four describes the proposed two-step enhanced Harris's hawk multi-objective optimizer (2S-ENDSHHMO) algorithm and implementation aspects. The chapter discusses in detail the population update and parameter adjustment strategies followed by describing the 2S-ENDSHHMO algorithm. Chapter Five presents the description of the experimental study performed as well as the results obtained and their discussion. Chapter Six presents the conclusions obtained, future work and contribution of this work.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter outlines the main approaches to solve MOPs via metaheuristics. In addition, the basic idea of major emerging SI-based metaheuristics and their extension to solve MOPs is considered. Section 2.2 of this chapter discusses the MOO and focuses on the reference point-based approach and metaheuristic methods. Section 2.3 presents the SI-based metaheuristics, focuses on MOSI-based algorithms and describes the Harris's hawk multi-objective optimizer algorithm. The discussion on the population update strategies in SI-based metaheuristics is presented in Section 2.4. The parameters settings approaches are discussed in Section 2.5 followed by the adaptive parameter control approaches in the SI-based metaheuristics in Section 2.6. In Section 2.7, the population generator methods are presented, followed by discussion on the population initialization approaches in the SI-based metaheuristics in Section 2.8. Finally, the chapter summary is presented in Section 2.9.

2.2 Multi-objective Optimization

According to the number of objective functions, optimization problems can be classified into SOPs and MOPs. The SOP consists of one objective. The goal of the single objective optimization process is to determine a solution with minimum/maximum cost for a given objective function. In this case, the input parameters (decision variables) of this function are manipulated within a predefined search space and provide only single best (optimal) solution. In the case of MOO, the

objective function returns multiple objectives related to a MOP. However, in the real-world, most optimization problems consist of two or more conflicting objectives, in which improving an objective leads to the degradation of others.

Compared to SOPs, solving a MOP is not a trivial task. The MOO can be defined as the process of simultaneously optimizing two or more objectives, subject to a set of constraints (Datta & Regis, 2016). Since these objectives conflict with each other, solving a MOP, generally, requires finding trade-off solutions. The trade-off solution is better in terms of one objective, and sacrifice on at least one other objective. The trade-off solutions, also called non-dominated solutions, form the Pareto front. In MOO, the solution *A* dominates *B* if *A* is not worse than *B* in all objectives and *A* is better than *B* in at least one objective (Emmerich & Deutz, 2018). In the set of non-dominated solutions, no single solution is better than others for all the objectives. Therefore, the DM is responsible for choosing a solution that addresses the overall objectives of the problem.

2.2.1 Classification of Multi-Objective Optimization Methods

According to their way of dealing with objective functions, the MOO methods, can be divided into four main categories, namely, scalarization-based, Pareto-based, decomposition-based and indicator-based methods (Emmerich & Deutz, 2018), as shown in Figure 2.1

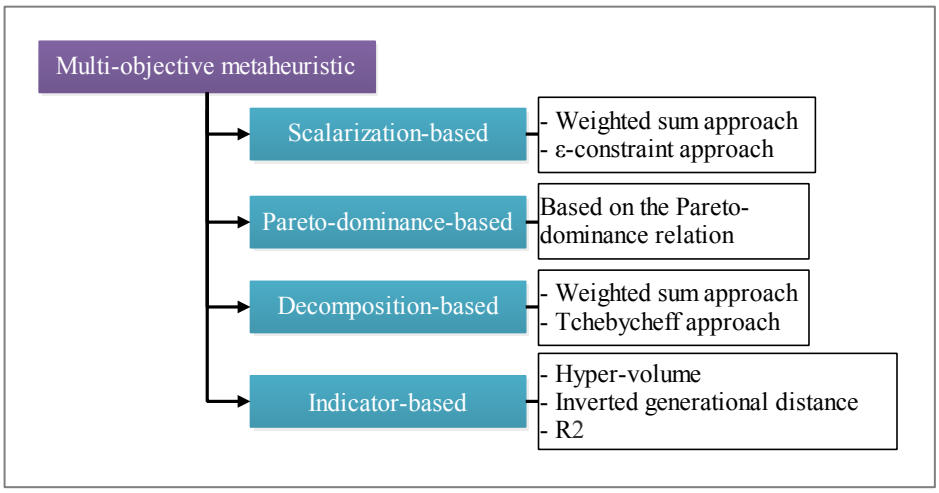


Figure 2.1. Classification of multi-objective optimization approaches

Scalarization-based approaches transform a MOP problem into one or a set of SOP problems by grouping the criteria to be optimized into a single objective function. The simplest aggregation methods use a scaling function for each criterion, to be able to add (additive model) or multiply them (multiplicative model).

The best-known and most widely used scalarization method is the weighted sum method. This method consists of adding all the objectives by assigning weight coefficients. These coefficients reflect the relative importance that the DM attributes to the objective (Emmerich & Deutz, 2018). The weighted-sum method is an approach developed to solve convex problems. However, the solutions obtained through this method are strongly dependent on the values of weights of each criterion. The choice of weights does not necessarily guarantee that the final solution will be accepted. In other words, these weights may not reflect, proportionally, the relative importance of the objectives and the problem with new weights will need to be resolved. Figure 2.2 illustrates the concept of the WS method (Jakob & Blume, 2014).

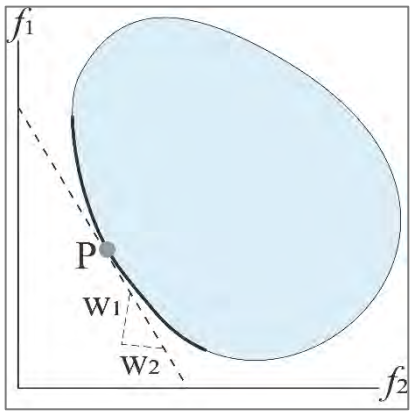


Figure 2.2. Geometric interpretation of the weighted-sum method with convex MOP

In Figure 2.2, the continuous and dashed lines represent the convex Pareto front of a MOP and weights vector, of each objective respectively. In solving convex MOPs, the point *P* can be reached by using the weights w_1 and w_2. However, in solving problems with non-convex Pareto front, some solutions may not be accessible using the weighted sum method (Brück, Faßbender, & Waffenschmidt, 2018; Jakob & Blume, 2014). Therefore, there is no guarantee that the Pareto curve will be well distributed, as shown in Figure 2.3 (Jakob & Blume, 2014).

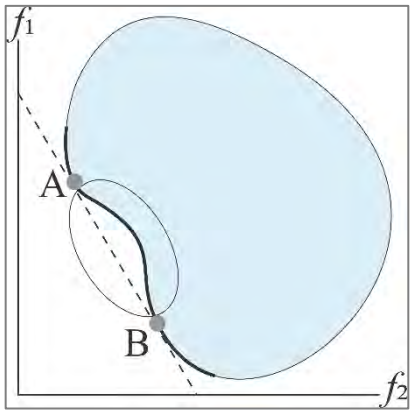


Figure 2.3. Non-convex MOP

In Figure 2.3, the weighted sum method can find points A and B while the solutions between points *A* and *B* cannot be evaluated by this method. That is, there are no

weight coefficients that can locate the points in that region (Hunter et al., 2019; Jakob & Blume, 2014; Tawhid & Savsani, 2018a).

Another scalarization-based method to solve MOPs was proposed by Vira and Haimes (1983), called the epsilon constraint (ε-constraint) method. In this method one objective is minimized while all other objective functions are used to form additional constraints. Solutions belonging to the Pareto boundary are obtained through a systematic variation of the boundary of these constraints. Compared to the weighted sum method, the ε-constraint method is able to handle a MOP with non-convex Pareto front. However, the main drawback of this method is determining the value of ε, which requires a priori knowledge about the true Pareto front of a MOP, which represents the real trade-offs between the objectives (also known as real Pareto front). Improper selection of ε can result in a formulation with no feasible solution. Furthermore, its computational complexity exponentially increases with the number of objectives (Jakob & Blume, 2014; Laumanns, Thiele, & Zitzler, 2006; van der Plas, Tervonen, & Dekker, 2012).

Pareto-dominance approach uses some form of Pareto dominance relationship among solutions to compare and assign them a score or select solutions. Based on the Pareto dominance relationship, a solution p is said to dominate q, if a solution p is better than q in at least one objective, and p is better than or equal to q in all objective functions (Emmerich & Deutz, 2018). In general, multi-objective optimization algorithms affect a score to a solution depending on whether it is dominated by other solutions of the

current population and possibly if it dominates other solutions. The main components of algorithms under this class are fitness assignment and diversity methods, such as sorting method, crowding distance and clustering methods (L. Li et al., 2016).

The Pareto dominance is the most popular approach in the field of MOO. This approach has become the main approach in solving MOPs because of their ability to find a potentially effective set of research studies through a population of solutions. However, the algorithm is developed based on the Pareto-dominance may face the loss of a selection pressure (Ochoa, Harvey, & Buxton, 2000) towards the Pareto front in solving many-objective optimization problems (Coello, Brambila, Gamboa, Tapia, & Gómez, 2019; J. Liu et al., 2019).

A decomposition-based approach decomposes the MOP into single objective optimization problems. Then, it uses a single-objective optimization algorithm to solve each sub-problem. The objective value of each sub-problem is a scalarization function of each objective. Each sub-problem corresponds to a weight vector (Dai & Lei, 2018; Tan, Lu, Liu, Wang, & Zhang, 2019). Many algorithms have been developed based on the decomposition approach. However, the decomposition-based algorithms still have challenges in obtaining a uniformly distributed solutions set (Coello et al., 2019). The decomposition-based approaches are strongly affected by the scalarization function that they adopt and are further sensitive to the method used to generate weights. If the weight vectors are not set properly, the algorithm will not be able to converge to the Pareto front (Weiszer, Chen, Stewart, & Zhang, 2018; Zhang, Zeng, Li, & Li, 2018).

Furthermore, the number of weight vectors grow exponentially with the number of objectives (Emmerich & Deutz, 2018).

Indicator-based approach was first proposed as a general framework by Zitzler and Künzli (2004). This approach uses performance indicators, such as, the hypervolume (HV) (Zitzler & Thiele, 1999) to score solutions. The principle is to define the multi-objective problem as a problem with the aim of maximizing the value of the indicator associated with the approximation (Emmerich & Deutz, 2018). Recently, the researchers started to get interested in indicator-based approach. However, it is not clear what were their advantages other than providing an alternative mechanism for selecting solutions (Coello et al., 2019). Based on the preference information provided by the DM, the MOO methods can be divided into three categories, namely a priori, progressive (interactive) and posteriori methods.

A priori methods require the DM to represent preferences information before starting the process of finding a solution. In other words, initialization of the optimization process, where the user assigns weights to the criteria or at least ranks the objectives. However, the DM does not have prior knowledge about the behaviour of the problem. Thus, it is difficult for the DM to define the preferences (define its goals), before starting the search process (Xin et al., 2018).

In the posteriori method class, the methods do not require the DM to provide preferences before, or during the optimization process. Instead, they are limited to

finding an approximation of the Pareto front, by modifying some of its parameters. Then, the DM chooses the most satisfied solution, according to its priorities. However, these methods are computationally expensive, because several runs are required before making a decision. Furthermore, the number of non-dominated solutions increases exponentially with the number of objective functions. This make it difficult for the DM to choose the most preferred solution (Xin et al., 2018).

In the interactive methods, the preference information of the DM is specified interactively during the run of the algorithm. This allows the DM to adjust the preferences and guide the search towards the preferred regions. In each iteration, the DM can learn from the solution process. Thus, the DM does not need to compare many non-dominated solutions, and the computational complexity will be reduced. Therefore, methods under the interactive class overcome the limitations of both a priori and posteriori methods, because, the DM does not need a global preference structure and only Pareto-optimal solutions are generated that are interesting to the DM (Xin et al., 2018). The preference information is usually provided in the form of reference points, reference areas and reference directions amongst others (L. Li et al., 2016).

2.2.2 Reference Point Approach

The reference point approach was first proposed by Wierzbicki (1980) and belongs to the family of interactive methods (Deb & Sundar, 2006). The reference point method uses the achievement scalarizing function (Thiele et al., 2009) to obtain solutions that fall closer to the True Pareto front in the preferred region of the search space. The

reference point, also called the aspiration level or goal vector, is determined by the DM, which includes the location of the reference point to guide the search process to a particular region in the True Pareto front and preference for different goals (Deb & Sundar, 2006). Figure 2.4 illustrates the concept of reference point method (Filatovas, Kurasova, Redondo, & Fernández, 2019).

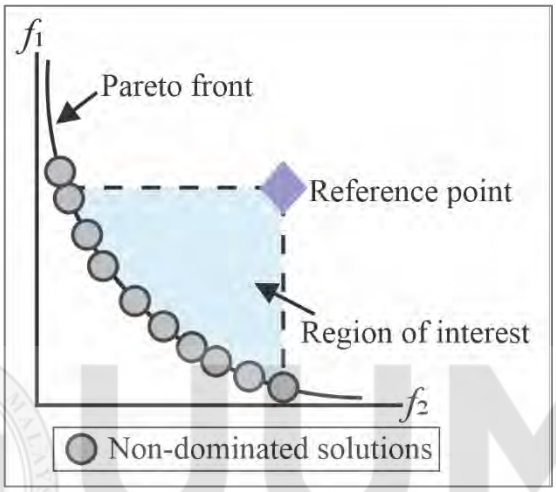


Figure 2.4. Reference point approach

In Figure 2.4, the reference point is determined by the DM in the objective space that represents the DM's aspiration level for each objective. The preferred region comprises a set of non-dominated solutions obtained by a reference point-based MOO algorithm.

Solving a MOP by searching for non-dominated solutions is not a trivial task. This is mainly due to the high complexity of a problem, which generates a large number of possible solutions, making it impossible to enumerate all. Given this difficulty, there is a growing interest in using the metaheuristic methods combined with MOO approaches, to solve MOPs. This is due to the fact that metaheuristics reach feasible solutions without having to list all possibilities. This class of MOP solving methods

uses metaheuristics to generate different solutions, as well as to determine the approximations of the Pareto front.

2.2.3 Metaheuristic Method

Metaheuristics are grouped under several classes; each class is the result of a specific point of view. The most intuitive way to classify metaheuristics is based on the origins of inspiration of the algorithm, the number of initial solutions or the method of usage of the objective function (Blum & Roli, 2003a). Figure 2.5 shows the taxonomy of metaheuristics.

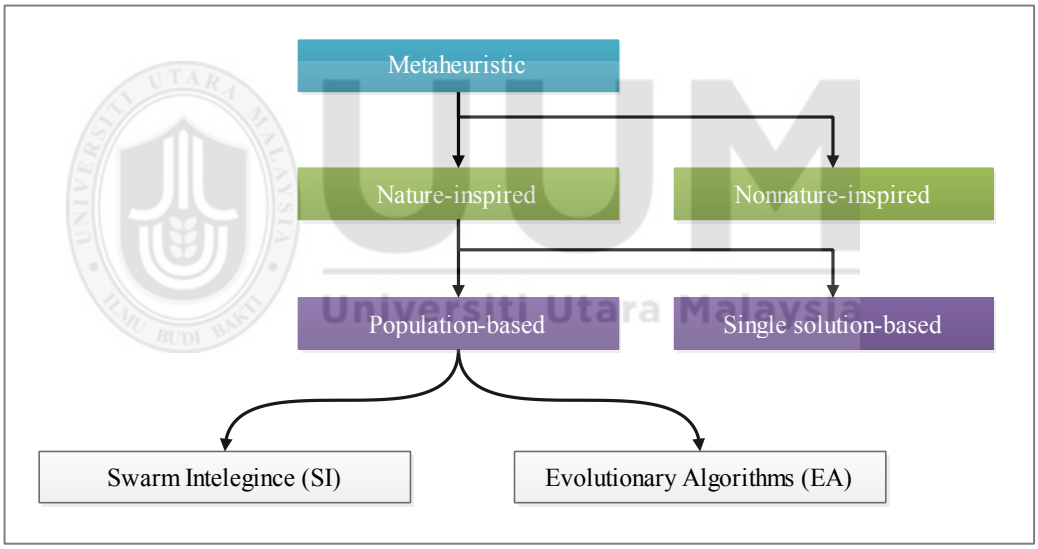


Figure 2.5. Classification of metaheuristics

Nature is a powerful source of inspiration for developing metaheuristics as it shows extremely diverse phenomena in biological and physical or chemical systems (Boussaïd et al., 2013; Dréo, Pétrowski, Siarry, & Taillard, 2006; Fister Jr, Yang, Fister, Brest, & Fister, 2013; Yang, 2010b). Nature-inspired metaheuristics can be sub-divided into population-based and trajectory-based, also known as single-solution-based metaheuristics (Blum & Roli, 2003a). These classes are based on the number of

solutions used in the search space. Basically, both single-solution-based and population-based metaheuristics aim to locate the global optimum in the search space through random moves (Adekanmbi & Green, 2015). However, single-solution-based metaheuristics start with a single solution and try iteratively to improve this solution by moving toward a local neighbourhood until reaching the best solution for the problem. These methods include but are not restricted to simulated annealing, tabu search, variable neighbourhood search (Mladenovic, 1995) guided local search (Voudouris, 1997) and iterated local search (Stützle, 1998). Population-based metaheuristics start from a set of initial potential solutions (initial population) and try to find a best solution by modifying solutions of that population (Blum & Roli, 2003a).

Population-based are more popular than single-solution-based metaheuristics and they perform better than single-solution-based methods (Prugel-Bennett, 2010; Saremi, Mirjalili, & Lewis, 2017). Population-based metaheuristics, as their name indicates, work on a population of solutions, which improve during optimization. If a solution is trapped into local optima, it can be assisted by other solutions in the population, to escape from the local optima (Saremi et al., 2017). Furthermore, population-based metaheuristics have more ability to explore the search space than single-solution-based metaheuristics (Ganchev, Garcia, Dobre, Mavromoustakis, & Goleva, 2019; Saremi et al., 2017). Therefore, there is a high probability of finding the global optimum solution for an optimization problem (Saremi et al., 2017). Examples of population-based metaheuristics include PSO (Kennedy & Eberhar, 1995) genetic algorithm (GA) (Goldberg & Holland, 1988; Holland, 1992), and ant colony optimization (Dorigo,

1992). Therefore, in this study, population-based metaheuristics will be studied. Each of these metaheuristics is characterized by its own way of solving problems and its source of inspiration. Basically, population-based metaheuristics, can be sub-divided into two main categories: Evolutionary Algorithms (EAs) (Back, 1996) and SI system (Blum & Li, 2008).

Evolutionary algorithm metaheuristics are inspired in principle by Darwin's evolution theory (Darwin, 1859). Based on this theory, individuals in the population are characterized by their respective genotypes, which give them, via their phenotype, a certain adaptation to the domain considered. In order to promote adaptation to the domain, the most adapted individuals have more chances (in the probabilistic sense) to reproduce over generations (the iterations of the algorithm), in order to transmit their genotype within the population. In the literature, various evolutionary algorithms have been proposed, such as, GA (Holland, 1992), differential evolution (Huang, Wang, & He, 2007), evolutionary strategy (ES) (Rechenberg, 1989) and biogeography-based optimization (Simon, 2008).

Most SI-based metaheuristics are inspired by the collective behaviour of groups in the biological systems, such as, fish schooling, bird flocking, ant colonies and animal herding (Yang, 2012c). However, not all SI-based metaheuristics are developed this way. Other algorithms have been developed using the inspiration of physical system, such as gravitational search algorithm (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009). Several SI-based algorithms have been proposed and successfully

applied to solve different optimization problems (Gogna & Tayal, 2013; Kar, 2016; Slowik & Kwasnicka, 2018; Yang, 2010a). In general, SI-based algorithms have earned more popularity over EAs (Del Ser et al., 2019) because, they provide effective strategies for solving complex optimization problems.

2.3 Swarm Intelligence-based Metaheuristic

The swarm intelligence is an artificial intelligence technique which refers to the local interactions between agents or environment by following some simple rules. This collective behaviour between agents often leads to the emergence of a global self-organized system, which can be considered as a kind of collective intelligence. The self-organizing properties of SI rely primarily on the interactions between its components. Many self-organized behaviours are based on direct interactions between members of a group. Probably the most obvious example is that of schools of fish and the flight of birds in which thousands or even millions of individuals move in a coherent way and change their direction synchronously (Hemelrijk, 2005; Pavlov & Kasumyan, 2000). Several SI-based metaheuristics were proposed. Figure 2.6 depicts the chronology of SI-based metaheuristics that have been proposed since 2010 until 2019.

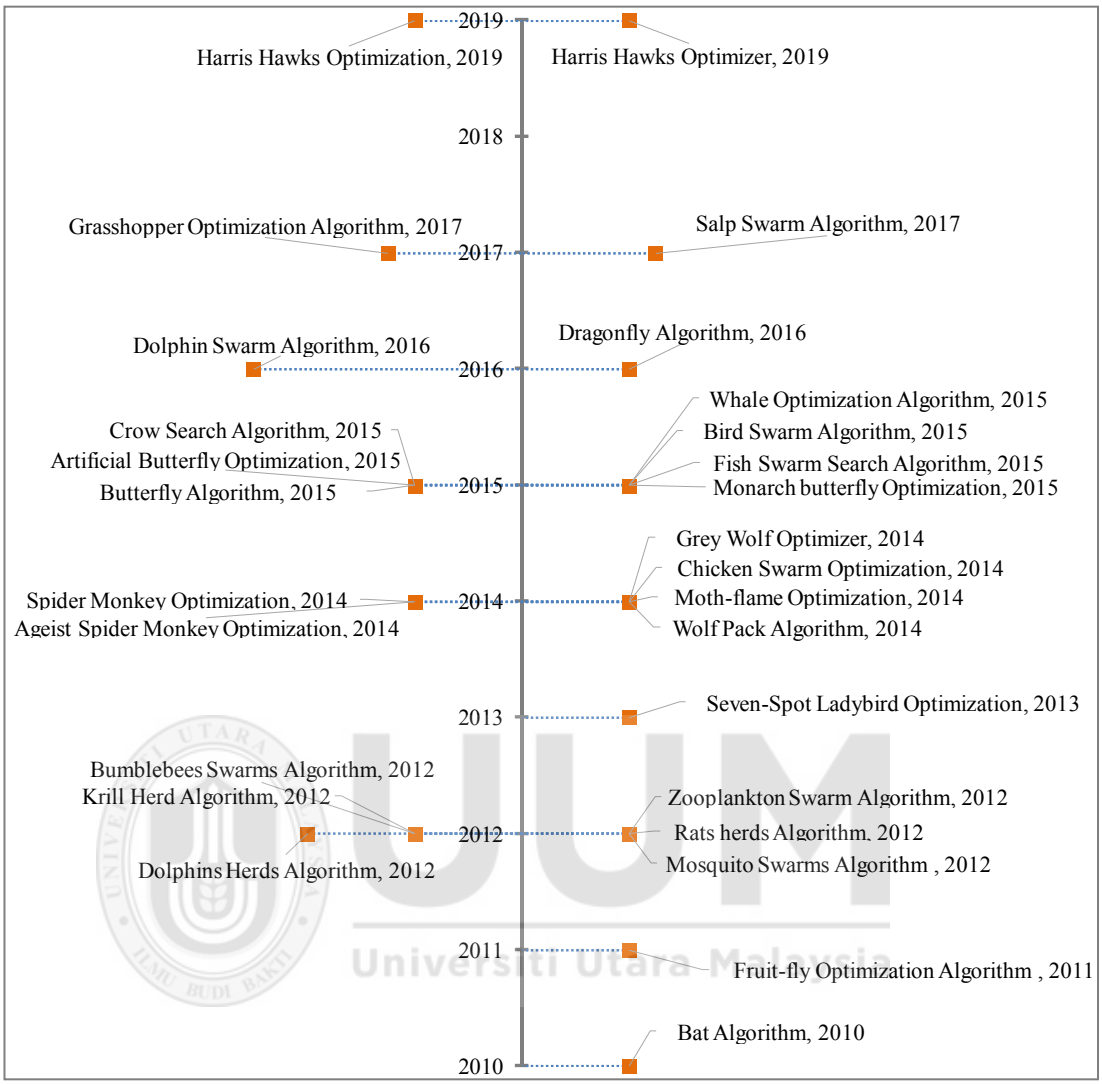


Figure 2.6. Chronology of SI-based metaheuristics

Figure 2.6 shows that most proposed SI-based metaheuristics imitate the behaviour of creatures in nature. Some are inspired by the same creature. However, the key difference between the metaheuristics is in the moving strategies of individuals in the search space, therefore, developing reliable strategies is continually evolving to produce effective metaheuristics (Adekanmbi & Green, 2015). The above presented SI-based metaheuristics are not exhaustive because the purpose of this section is in understanding the methods and techniques used in the process of inspiration of

phenomena and natural systems. Furthermore, some have not attracted much attention in the research community.

One of the most popular examples of the SI-based metaheuristic is the PSO algorithm. This algorithm was first proposed by Kennedy and Eberhar (1995), and quickly became one of the best metaheuristics in the field of optimization. The PSO was originally inspired by the social behaviour of swarming animals, such as schools of fish and flocks of birds. Since being proposed, the PSO algorithm has been applied in various applications in different fields (Babazadeh, Poorzahedy, & Nikoosokhan, 2011; Kuo & Huang, 2009; Ma, Yu, & Hu, 2013; Nenortaite & Butleris, 2008; Pluhacek, Senkerik, Viktorin, Kadavy, & Zelinka, 2017). However, the main drawback of the PSO algorithm is that it becomes easily trapped in local optima which leads to premature convergence, (Larsen, Jouffroy, & Lassen, 2016; Xu et al., 2015), especially in solving complex high-dimensional optimization problems (Xu et al., 2015). The premature convergence occurs due to the decrease of diversity in the swarm, where it cannot escape from a local optima (Nezami, Bahrampour, & Jamshidlou, 2013).

Cat swarm optimization has been proposed by Chu, Tsai, and Pan (2006) when observing the behaviour of cats. The cat swarm optimization algorithm, applied in various fields, provides satisfactory results compared to other algorithms, such as PSO, GSA (Rashedi et al., 2009) and GA (Kumar & Sahoo, 2016; Panda, Pradhan, & Majhi, 2011; Santosa & Ningrum, 2009). The performance of cat swarm optimization

algorithm has been compared with PSO and PSO with weighting factor (PSO-WF), using six test functions (Chu et al., 2006). However, the paper does not provide any details regarding the parameters used in the test, such as the dimension, population size and the number of evaluations for each algorithm. According to the authors, the cat swarm optimization algorithm outperforms PSO and PSO-WF. However, based on the convergence graph provided in the paper, cat swarm optimization algorithm is prone to lose diversity, which leads to premature convergence. The cat swarm optimization algorithm uses velocity and the update formula of the PSO algorithm to move the individuals closer to the optimal solution. As in PSO, by increasing the number of iterations, the algorithm performance degrades significantly. This may lead to premature convergence, especially when dealing with high-dimensional multimodal functions (Nie, Wang, & Nie, 2017).

Another popular SI-based algorithm is the artificial bee colony algorithm (Karaboga, 2005) inspired by the behaviour of bee colonies. The artificial bee colony algorithm has advantages; it is robust and simple in concept, easy to use and has few control parameters. Moreover, the artificial bee colony algorithm has been successfully used in different applications (Kaswan, Choudhary, & Sharma, 2015). In Karaboga and Akay (2009), the performance of the artificial bee colony algorithm was compared with PSO, GA and differential evolution algorithms. As stated in Karaboga and Akay (2009), the artificial bee colony algorithm provided a superior performance in solving most test functions. However, the artificial bee colony algorithm has some

disadvantages, such as poor population diversity, which makes it prone to premature convergence (Ng, Lee, Zhang, Wu, & Ho, 2017; Panniem & Puphasuk, 2018).

In Yang (2009), the firefly algorithm was proposed. This algorithm was inspired by the light-emitting behaviour of fireflies. These insects use special organs to produce light inside their bodies. This light production is a form of chemical reaction called bioluminescence (Stanger-Hall, Lloyd, & Hillis, 2007). The basic functions of these flashes are to attract mating partners (communication), and a prey. In addition, blinking can also serve as a protection alert mechanism. The attraction between fireflies is proportional to brightness, and this decreases as the distance grows. So, for any two shining fireflies, the one of lesser intensity will move toward the greater one. If there is no brightness difference, the movement occurs at random. The main advantages of the FA are simple structure and less adjustment parameters. Therefore, the application of the algorithm is quite extensive. According to the author, the FA has superior performance compared to PSO and GA algorithms, in terms of both efficiency and success rate. However, FA not only has these advantages, but also has its own defects, such as it is easy to fall into local optima in high-dimensional, complex optimization problems and premature convergence (Kasdirin, Yahya, Aras, & Tokhi, 2017).

The gravitational search algorithm proposed by Rashedi et al. (2009), was inspired by the physical theory of Newton. In the gravitational search algorithm, the searcher agents consist of a collection of masses, and their interactions are based on Newton's

laws of gravity and motion (Rashedi et al., 2009). The gravitational search algorithm has been evaluated using standard benchmark functions. The results were compared with real genetic algorithm and PSO. According to Rashedi et al. (2009), gravitational search algorithm outperforms other tested algorithms. However, gravitational search algorithm is prone to premature convergence problems due to the loss of diversity where all agents in the population converge to the same single point in the search space (Jiang, Wang, & Ji, 2014; Li & Dong, 2017).

The bat algorithm (Yang, 2010c) mimics the echolocation behaviour of microbats, which allows them to efficiently locate and hunt their prey even in complete darkness (Jacobs & Bastian, 2017). BA has been applied in different applications (Gandomi, Talatahari, Yang, & Deb, 2013; Yang & He, 2013) and produces reliable results. The performance of BA was tested and compared with the performance of PSO and GA. According to the author, BA outperforms other algorithms in terms of accuracy and efficiency.

Grey wolf optimizer (GWO) (Mirjalili et al., 2014) is one of the recently proposed SI-based optimization algorithms which mimics the hunting behaviour of the grey wolf. The GWO algorithm has the advantages of simple operation and less adjustment parameters. Furthermore, GWO has been used in different applications, and the results show that the GWO outperforms other well-known algorithms such as PSO (Precup, David, Szedlak-Stinean, Petriu, & Dragan, 2017), ABC (Mustaffa, Sulaiman, & Yusof, 2015) and GA (Siavash, Pfeifer, Rahiminejad, & Vahidi, 2017). In Mirjalili et

al. (2014) various benchmark functions were used to evaluate the performance of the GWO algorithm and compare it with PSO, GSA, differential evolution and fast evolutionary programing (Yao, Liu, & Lin, 1999). According to the authors, GWO outperforms other algorithms in terms of exploration, exploitation, local optima avoidance and convergence.

Moth-flame optimization algorithm (Mirjalili, 2015) is one of the most recent SI-based metaheuristics. This algorithm inspired by the navigation behaviour of moths in nature. Moth-flame optimization was tested using a set of benchmark functions and the performance of the algorithm was compared with PSO, GSA, BA, and flower pollination algorithm (Yang, 2012b), state of matter search (Cuevas, Echavarría, & Ramírez-Ortegón, 2014), FA, and GA with a population size of 30 and 1,000 iterations for each algorithm. The results show that the moth-flame optimization algorithm has the ability to provide competitive results compared to other algorithms (Mirjalili, 2015). However, this algorithm is prone to becoming trapped in local optima, especially in solving multimodal functions (Li, Zhou, Zhang, & Song, 2016).

The monarch butterfly optimization (Wang, Deb, & Cui, 2015) algorithm simulates the migration of monarch butterflies. The main operators in the monarch butterfly optimization algorithm are adjusting and migration operators. These operators determine the search movement direction of the individuals in the search space. The main advantage of monarch butterfly optimization is that it is easy to implement because it only needs to fine-tune migration and adjusting operators. The performance

of monarch butterfly optimization is evaluated using various benchmark functions and compared with five optimization algorithms, namely ABC, biogeography-based optimization, ant colony optimization, stud GA (SGA) (Khatib & Fleming, 1998) and differential evolution. According to the authors, monarch butterfly optimization outperforms other tested algorithms in solving most benchmarks functions.

The salp swarm algorithm is one of the most recent metaheuristics inspired by the swarming and foraging behaviour of salps (Mirjalili et al., 2017). Salps construct a swarm called a salp chain and move from one place to another using foraging and rapid coordinated changes, to achieve better locomotion (Anderson & Bone, 1980). The salp swarm is divided into a leader with follower groups. The followers follow each other and are guided by the leader. The salp swarm algorithm has been tested using a set of different benchmark functions. Furthermore, the performance of the algorithm has been compared with other algorithms, namely, PSO, flower pollination algorithm, state of matter search, GSA, FA, BA and GA. According to the authors, salp swarm algorithm is able to determine the global optima for most benchmark functions and outperform other optimization algorithms. However, the salp swarm algorithm is prone to fall into local optima and premature convergence (Sayed, Khoriba, & Haggag, 2018).

Another recent metaheuristic is the grasshopper optimization algorithm (GOA). This algorithm simulates the repulsion and attraction forces between grasshopper swarms (Saremi et al., 2017). The performance of the GOA has been compared with PSO, GA,

differential evolution, BA, FA and flower pollination algorithm using a set of benchmark functions and engineering problems (Saremi et al., 2017).

2.3.1 MOSI-based Metaheuristics

Several MOSI-based algorithms have been proposed. These algorithms were deduced by various existing single-objective algorithms. This section provides a review for the multi-objective PSO, ABC, FA, BA, GSA, GWO, BFO and moth-flame optimization algorithms. These algorithms are the most popular SI-based metaheuristics that have been successfully applied in solving different optimization problems in various fields (Lones, 2020). These algorithms are mainly developed to solve single objective optimization problems. Therefore, to solve MOPs, they are integrated with a MOO approach to handle multiple objectives.

Scalarization-based approach

In Yang (2012a), the proposed MOBA extends the BA algorithm to solve a MOP. This algorithm was developed based on the weighted sum approach. However, this approach cannot provide an efficient performance in solving complex and non-convex MOPs (refer Figure 2.3). In Yang (2013) the same author of MOBA proposed the multi-objective firefly algorithm (MOFA), which improves the firefly movement formula by using Lévy flights, which is used to maintain population diversity. The multi-objective firefly, as in MOBA, was developed based on the weighted sum approach. Therefore, it is not suitable for solving non-convex MOPs. Mellal and Zio (2019) proposed a MOPSO based on the weighted sum method. In the proposed algorithm, the Lévy flight is used to maintain the population diversity. The

performance of the algorithm is tested using a set of constraint numerical benchmark MOPs and compared with the PSO algorithm.

Pareto-dominance-based approach

In Coello and Lechuga (2002), the Pareto dominance approach was employed in the PSO algorithm (Kennedy & Eberhar, 1995) to make it capable of dealing with MOPs and guide the particles in the search space. Two approaches were introduced to the PSO, namely an external memory called repository to identify a leader that guides the particle and a geographically-based approach to maintain diversity (Coello & Lechuga, 2002). Later on, Coello, Pulido, and Lechuga (2004) extended MOPSO (Coello & Lechuga, 2002) by integrating a mutation operator to improve the exploration ability of the algorithm. Several studies proposed modifications for the Pareto dominance-based MOPSO by integrating different methods to maintain the population diversity and select the leaders (personal and global best solutions) (Coello et al., 2004; Man-Im, Ongsakul, Singh, & Boonchuay, 2015; Sierra & Coello, 2005)

Many others MOSI algorithms have been proposed based on the Pareto dominance approach. These algorithms used an external archive (Coello & Lechuga, 2002) to save non-dominated solutions and proposed different strategies to maintain the population diversity (Akbari et al., 2012; Bhowmik & Chakraborty, 2015; Hassanzadeh & Rouhani, 2010; Kumawat, Nanda, & Maddila, 2017; Mahmoodabadi & Shahangian, 2019; Mirjalili et al., 2016; Mohamed et al., 2016; Niu, Wang, Wang, & Tan, 2013; Yang & Ji, 2016; Zhou & Yao, 2017). In many studies, the Pareto dominance-based MOSI algorithms employed the non-dominated sorting approach and crowding

distance to select the next generation and maintain the population diversity (Chen et al., 2019; Jangir & Jangir, 2018; Kishor et al., 2016; Lv et al., 2019; Prakash, Trivedi, & Ramteke, 2016; Savsani & Tawhid, 2017; Tsai, Huang, & Chiang, 2014; Zellagui et al., 2017)

Decomposition-based approach

Peng and Zhang (2008) proposed the first MOPSO algorithm based on the decomposition approach. They followed the same concept as the multi-objective evolutionary algorithm based on decomposition (Zhang & Li, 2007) and replaced the genetic algorithm with a standard PSO algorithm. In the proposed algorithm an external archive was used to store the non-dominated solutions found during the search process. Sapre and Mini (2020) followed the same concept and replace the GA with a moth flame optimization algorithm. Dai, Wang, and Ye (2015) propose a MOPSO based on the decomposition approach. In the proposed algorithm, the Pareto optimal solution was generated for each sub-region in the objective space. The elitist strategy was employed to maintain the population diversity. Furthermore, the crowding distance was used to improve the convergence of the proposed algorithm. Others studies proposed a decomposition-based MOSI algorithms by utilizing a penalty boundary intersection approach (Bai & Liu, 2016; Zapotecas Martínez & Coello Coello, 2011). The penalty boundary intersection approach calculates the weighted aggregate for a given weight vector and reference vector.

Indicator-based MOMH

Compared to the Pareto dominance and decomposition-based approaches, the indicator-based approach has received relatively little attention in the area of the MOSI-based algorithms. García, Coello, and Arias-Montano (2014) proposed a MOPSO based on the HV (Zitzler & Thiele, 1999) indicator. The proposed algorithm uses of the HV contribution value to select the global and personal best solution from an external archive and as a mechanism for updating the external archive during the optimization process.

Other studies followed the same concept by using the R2 indicator instead of the HV. Díaz-Manríquez, Toscano, Barron-Zambrano, and Tello-Leal (2016) proposed a MOPSO algorithm based on the R2-indicator. In the proposed algorithm, instead of Pareto dominance and the external archive, the proposed algorithm used the R2-indicator to guide the search process. Wei et al. (2018) proposed a MOPSO algorithm based on R2 indicator. In the algorithm, an external archive is used to store non-dominated solutions. The R2 indicator contribution value is used to select individual from the external archive instead of the crowding distance. The diversity of swarm in the archive is maintained through polynomial mutation (Deb, Pratap, Agarwal, & Meyarivan, 2002a).

Others MOSI-based algorithms combined two or more MOO approaches to handle the multiple objectives. The combination of Pareto dominance and decomposition -based approaches was first proposed by Al Moubayed, Petrovski, and McCall (2014). Al

Moubayed et al. (2014) proposed a MOPSO by integrating both Pareto-dominance and decomposition approaches based on the penalty boundary intersection to transform a MOP into a set of SOPs. They introduce a mechanism to select the particle leaders and an archive technique to collect the non-dominance solutions based on the crowding-distance. Lin, Li, Du, Chen, and Ming (2015), followed same concept and proposed a MOPSO by combining the Pareto-dominance and decomposition approaches.

Some studies combined the R2-indicator and decomposition approaches. Li, Liu, Tan, and Yu (2015) proposed a MOPSO algorithm based on the R2-indicator and decomposition approaches. The R2-indicator contribution value is used to select the global best solution from the archive, whereas the personal best position is updated based on the scalarization approach, such as weighted sum, Tchebycheff and penalty boundary intersection. Inspired by R2-MOPSO which was earlier proposed in Li et al. (2015), J. Liu et al. (2019) proposed a MOPSO based on the R2-indicator and decomposition-based archiving pruning strategy. In this algorithm, the global-best leader is selected based on the R2 indicator and the personal-best leader is selected based on the Pareto dominance approach.

The epsilon indicator and Pareto dominance approaches were used to extend the SI-based single objective optimization algorithm to the MOO algorithm. Luo et al. (2017) proposed MOABC. In the proposed algorithm, the solutions are evaluated based on the epsilon indicator and the Pareto dominance approach is used to compare the solutions and an external archive to save the non-dominated solutions. Table 2.1

summarises the MOO approaches applied in some of the well-known SI-based metaheuristics.

Table 2.1
Summary of the MOSI-based metaheuristics

Algorithm Reference	Algorithm	Approach	Archive	Application domain
Coello and Lechuga (2002)	MOPSO	Pareto-dominance	✓	-
Coello et al. (2004)	MOPSO	Pareto-dominance	✓	-
Janga Reddy and Nagesh Kumar (2007)	MOPSO	Pareto-dominance	✓	Engineering design
Wickramasinghe and Li (2008)	MOPSO	Pareto-dominance	-	-
Hassanzadeh and Rouhani (2010)	MOGSA	Pareto-dominance	✓	-
Akbari et al. (2012)	MOABC	Pareto-dominance	✓	-
Niu et al. (2013)	MOBFO	Pareto-dominance		-
Yi, Huang, Fu, He, and Li (2015)	MOBFO	Pareto-dominance	✓	Aluminium Electrolysis Production Process
Carrese and Li (2015)	MOPSO	Pareto-dominance	✓	Airfoil design
Man-Im et al. (2015)	MOPSO	Pareto-dominance	-	-
Yang and Ji (2016)	MOBFO	Pareto-dominance	✓	-
Prakash et al. (2016)	MOBA	Pareto-dominance	✓	-
Bhowmik and Chakraborty (2015)	MOGSA	Pareto-dominance	✓	Optimal power flow problem
Mirjalili et al. (2016)	MOGWO	Pareto-dominance	✓	-
Mohamed et al. (2016)	MOGWO	Pareto-dominance	✓	Optimal power flow problem

Mirjalili (2016)	MODA	Pareto-dominance	✓	
Kishor et al. (2016)	MOABC	Pareto-dominance	✓	-
Savsani and Tawhid (2017)	MOMFO	Pareto-dominance	-	-
Kumawat et al. (2017)	MOWOA	Pareto-dominance	✓	-
Zellagui et al. (2017)	MOGSA	Pareto-dominance	✓	Power transformer design
Jangir and Jangir (2018)	MOGWO	Pareto-dominance	✓	-
Sun and Gao (2019)	MOPSO	Pareto-dominance	✓	-
Lv et al. (2019)	MOFA	Pareto-dominance	✓	-
Chen et al. (2019)	MOIBA	Pareto-dominance		Optimal power flow problem
Yang (2012a)	MOBA	Scalarization	-	-
Yang (2013)	MOFA	Scalarization	-	-
Mellal and Zio (2019)	MOPSO	Scalarization	-	System reliability optimization
DeBruyne and Kaur (2016)	HHMO	Scalarization	-	-
Peng and Zhang (2008)	MOPSO	Decomposition	✓	-
Zapotecas Martínez and Coello Coello (2011)	MOPSO	Decomposition	-	-
Dai et al. (2015)	MOPSO	Decomposition	-	-
Bai and Liu (2016)	MOABC	Decomposition	-	-
Sapre and Mini (2020)	MOMFO	Decomposition	✓	Placement of relay nodes in WSNs
Sierra and Coello (2005)	MOPSO	Pareto- & ε-dominance	✓	-
D^2MOPSO Al Moubayed et al. (2014)	MOPSO	Pareto-dominance & decomposition	✓	-

Lin et al. (2015)	MOPSO	Pareto-dominance & decomposition	✓	-
García et al. (2014)	MOPSO	Indicator-based	✓	-
Díaz-Manríquez et al. (2016)	MOPSO	Indicator-based	-	-
Wei et al. (2018)	MOPSO	Indicator-based	✓	-
Luo et al. (2017)	MOABC	Indicator-based & Pareto dominance	✓	-
Li et al. (2015)	MOPSO	R2 indicator, scalarizing approach & decomposition	-	-
J. Liu et al. (2019)	MOPSO	R2 indicator, Pareto dominance & decomposition	✓	-

It can be concluded that most of the previous multi-objective SI-based algorithms were developed based on the Pareto dominance or decomposition approach. The non-dominated sorting approach has been used with the Pareto-dominance approach in numerus algorithms to select the non-dominated solutions. In most of these algorithms, the crowding distance (Deb, 2014) is used with non-dominated sorting approach to estimate the density and maintain the diversity of population. However, in some cases, the crowding distance approach cannot be used to select appropriate solutions, which may affect the diversity of solutions (Vachhani, Dabhi, & Prajapati, 2016). Most of MOSI-based algorithms follow the approach of MOPSO (Coello et al., 2004) in which, they used an external archive (external memory) to store the non-dominance solutions.

2.3.2 Reference Point-based MOSI Optimization Algorithm

The main goal of MOO is to support a DM to find his/her most preferred solution. However, most existing MOSI metaheuristics focus on approximating the whole Pareto front, as shown in Figure 2.7 (a). Approximating the whole Pareto front not only increases the computational cost (Qi, Li, Yu, & Miao, 2019), but also leads to difficulties in distinguishing a promising solution that is preferred by the DM, which increases exponentially with the number of objectives. To facilitate the DM procedure, preference information provided by the DM can be integrated with MOSI algorithms to guide the search process toward a preferred region, also called region of interest, as shown in Figure 2.7 (b). This leads to reduce the computational cost, improvement in the performance of an algorithm in solving MOPs and it helps the DM to make the final decision. Solutions outside the region of interest can be considered as noisy data and there is no guarantee of finding the preferred solutions when solving complex MOPs (Li et al., 2018a; Qi et al., 2019; Weiszer et al., 2018; Zhu, Gao, Du, Cheng, & Xu, 2018) .

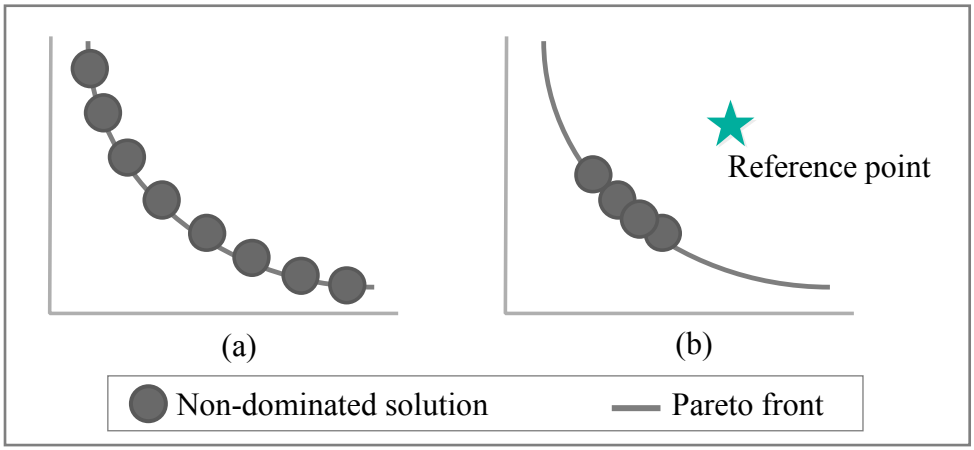


Figure 2.7. Non-dominated objective vectors: (a) evenly distributed across the whole Pareto front; (b) clustered near a reference point (near the region of interest) (Adapted from Filatovas, Kurasova, and Sindhya (2015))

By combining preference information from the DM with the optimization process, the search process is directed towards the region of interest. Among the approaches described to express DM preferences, this study highlights MOSI-based algorithms that make use of the reference point method.

In the context of finding a set of non-dominated solutions within a region of interest, instead of the entire Pareto-optimal set, several MOSI-based algorithms have been proposed. In Wickramasinghe and Li (2008) proposed two MOPSOs based on the reference point approach. One of the algorithms performed the non-dominated sorting with the crowding distance to maintain the population diversity. In the second MOPSO algorithm, the fitness of a particle is calculated using the maximum of the minimum values between the particle and all other particles in the population. The ε-clearing strategy is included in both MPSOs to control the diversity of non-dominated solutions.

Allmendinger et al. (2008) proposed a MOPSO based on the reference point approach. The algorithm uses reference information to select the leader and steady-state approach to compare the offspring with its parents. In the same paper, the author proposes an extended replacement strategy which allows the offspring to replace a particle that dominates the offspring and has a larger Euclidean distance to the reference point. The ε-clearing strategy (Deb & Sundar, 2006) is also used in the algorithm to maintain population diversity. According to Allmendinger et al. (2008), the proposed algorithm

showed good results in solving MOPs with two and three objectives. However, there were difficulties when solving highly multimodal problems test problem.

In Carrese and Li (2015), the proposed MOPSO algorithm, the PSO algorithm is integrated with the reference point method. The non-dominated sorting approach with crowding distance is applied to maintain the population diversity. The external archive is used to save the non-dominated solutions that have minimum distance to the reference point. This algorithm was developed to solve a particular problem, and it was not tested in solving different MOPs with different characteristics.

In DeBruyne and Kaur (2016), the HHMO algorithm was developed based on the reference point approach. The mathematical model of this algorithm is developed based on the GWO (Mirjalili et al., 2014) and MOGWO (Mirjalili et al., 2016) algorithms. Therefore, it has same advantages in terms of few parameters, easy to use, flexible, scalable, and has a special capability to escape out of local optima (Faris et al., 2018). However, the HHMO algorithm has limitations that degrade its effectiveness, especially in solving complex MOPs, resulting in poor approximation for the Pareto front of a MOP.

In the literature, there are other SI-based algorithm that have been developed based on the behaviour of Harris's hawk in nature (Bednarz, 1988). In Heidari et al. (2019), a single objective Harris's hawk optimization algorithm mimics the cooperative behaviour and chasing style of Harris's hawks in nature. The performance of the

algorithm has been compared with other algorithms, namely, PSO, differential evolution (Simon, 2008), FA (Gandomi, Yang, & Alavi, 2011), CS (Gandomi, Yang, & Alavi, 2013), biogeography-based optimization (Simon, 2008), BA/BAT (Mohan, Sivaraj, & Priya, 2016), GWO (Mirjalili et al., 2014), teaching learning-based optimization (Rao, Savsani, & Vakharia, 2012), flower pollination algorithm (X. S. Yang, M. Karamanoglu, & X. He, 2014b) and moth-flame optimization (Mirjalili, 2015) algorithms.

In Bairathi and Gopalani (2020) another single objective Harris's hawk optimization algorithm was proposed based on the cooperative hunting behaviour of Harris' hawks in nature. This algorithm includes two main phases, namely a non-hunting phase to perform the exploration process and a hunting phase to perform the exploitation process, which is further divided into local search and global attack phases. As in the PSO algorithm, the hawk in Harris's hawk optimization algorithm has two main components, namely personal best and global best position. Updating the position of hawks mainly depends on these components, in addition to three controlling variables, which are used to control the exploration and exploitation. According to Bairathi and Gopalani (2020), the Harris's hawk optimization algorithm showed superior performance compared to other algorithms, namely PSO, differential evolution, GWO and whale optimization algorithm (Mirjalili & Lewis, 2016) in solving a set of benchmark functions.

These algorithms by Heidari et al. (2019) and Bairathi and Gopalani (2020)

were initially developed to solve SOPs. However, as discussed in Section 2.2, there are fundamental differences between a SOP and MOP. Therefore, the basic metaheuristics cannot be used directly to solve the MOP. They require some modification/adaptation to handle MOPs. In Du et al. (2019) the multi-objective Harris's hawks optimization algorithm extends the single objective Harris's hawk optimization algorithm proposed by Heidari et al. (2019), to solve MOPs. The multi-objective Harris's hawk optimization algorithm was developed based on the Pareto dominance approach and the external archive mechanism is used to store the non-dominated solutions, as well as the roulette wheel selection which is utilized to carry out the optimization process. The performance of the multi-objective Harris hawks optimization algorithm is compared with MOGOA (Mirjalili et al., 2018), MOPSO (Coello & Lechuga, 2002) and MOSSA in solving four test problems with linear front. According to Du et al. (2019), the multi-objective Harris's hawk optimization algorithm showed superior performance compared to other algorithms. However, the test did not include MOPs with non-uniform or multimodal Pareto front, and it restricted the solution of MOPs with only two objectives. Furthermore, as is the case in most MOSI-based optimization algorithms, the obtained non-dominated solutions are evenly distributed across the true Pareto front (Du et al., 2019).

It is worth to mention that, the multi-objective Harris's hawk optimization algorithm (Du et al., 2019) is not related to the HHMO proposed by DeBruyne and Kaur (2016), where, each algorithm has a different mathematical model. Furthermore, the new metaheuristics, like Heidari et al. (2019), Bairathi and Gopalani (2020) and (Du et al.,

2019) algorithms, have not been applied in many applications, because they remain under experiment. According to the no-free-lunch theorem, there is no single metaheuristic that can be used to solve all optimization problems. The HHMO proposed by DeBruyne and Kaur (2016) will be discussed in detail in Section 2.3.3.

2.3.3 Reference point-based Harris's Hawk Multi-objective optimizer

Harris's hawk (Parabuteo unicinctus) is a raptor (bird of prey) that belongs to the Accipitridae family, a name given by the famous American ornithologist, James Audubon, in honour of the ornithologist Edward Harris, a friend and financial partner of various campaigns carried out together in North American. The Harris's hawk is a medium-sized raptor with exceptional flying ability and dexterity (Bednarz, 1988).

Harris's hawks attacks are quite coordinated. In contrast to other raptors, Harris's hawks tend to live in groups (Bednarz, 1988). They aggregate at one perch site and, once assembled, the hunting members split into two groups (DeBruyne & Kaur, 2016). The first group consists of lookout hawks, which fly out on a series of short flights, then land on relatively high perches. Harris's hawks have the ability to capture any prey from a long distance by homing in on it before the hunt. This is due to their excellent vision, possibly eight times greater than that of humans. This behaviour is most intense when the hawks seem to be actively searching for prey. Subsequently, the lookout hawks coordinate with a second group, which consists of ground hawks, to direct it toward the potential prey. Females will usually take the role of making the

final move to kill the prey, after the males have done all the work to flush out and tire the prey; the females will also be allowed to feed first and will even perch higher than the males to show their dominance.

The HHMO (DeBruyne & Kaur, 2016) was developed based on the reference point approach. Instead of returning non-dominated solutions evenly distributed along the true Pareto front, as is the case in most classical MOSI-based metaheuristics, the HHMO algorithm returns clusters of solutions sets near the provided reference points (Deb & Sundar, 2006). This helps the DM in making a better and more reliable decision. Another advantage of HHMO is it requires low computational cost to return clusters of non-dominated solutions, based on reference points determined by the DM (DeBruyne & Kaur, 2016).

In the HHMO, the hawks are subdivided into two distinct groups, based on their distance to the reference points. The first group consists of lookout hawks, which represents the reference points, while the second group consists of ground hawks, which represents the search for optimal solution sets. The best three hawks (alpha, beta, and delta) in each group are selected based on their distance to the reference points. These hawks represent the three best non-dominated solutions. However, the HHMO algorithm has some limitations which affect its performance.

In the HHMO, the positions of hawks are updated based on the average position of first three best positions of hawks (leaders). This obliges other hawks to update their

positions according to the position of the first three best hawks in the search space (DeBruyne & Kaur, 2016). Therefore, the rest of the hawks do not play any role in updating the position of leaders, which leads to the algorithm becoming trapped in local optima.

Furthermore, in the HHMO algorithm, if a convergence parameter is not set properly, the quality of the solution will become very poor and the HHMO algorithm requires a large amount of computation and takes a long time to solve the problem. The convergence parameter ensures that the algorithm performs the exploration process in the early period, and exploitation in the later period. However, in the MOO, the requirement for the exploration ability of the algorithm is high. Therefore, it is necessary to improve the adjustment strategy of a convergence parameter to improve the performance of the algorithm.

Moreover, the existing HHMO algorithm uses the RNG method to initialize a population; however, this method does not guarantee that the initial population can be evenly distributed in the search space of the problem. This will affect the utilization of the initial population by the optimization process. Furthermore, in DeBruyne and Kaur (2016) the authors do not provide any test that shows the performance of HHMO in solving three or more objectives with high-dimensional MOPs. This algorithm is relatively new and, according to the no-free-lunch theorem, there is no single algorithm that can be efficiently used to solve all optimization problems. If the algorithm provides an efficient performance in solving a particular problem, this does not mean

it will be able to provide the same performance in solving other problems. This study will focus on addressing these limitations and finding proper methods to overcome them and improve the performance of the HHMO algorithm.

2.4 Population Update Strategy in MOSI-based Metaheuristics

In MOPSOs, the problem of premature convergence is still one of the main issue that affect the performance of MOPSO (Ünal & Kayakutlu, 2020). The premature convergence in the earlier stage is due to loss of population diversity which leads to the algorithm trapped in local optima and cause poor convergence toward the true Pareto front (Ünal & Kayakutlu, 2020). The population update strategy used by an algorithm plays an important role in determining the performance of the algorithm, in terms of producing new solutions that helps in maintaining the population diversity and prevent the premature convergence.

Coello et al. (2004) extended the MOPSO proposed by Coello and Lechuga (2002) by integrating a mutation operator to improve the exploration ability of the algorithm. Sierra and Coello (2005) used uniform and non-uniform mutation schemes to preserve the population diversity in MOPSO. Peng and Zhang (2008) employed the polynomial mutation after calculating the new positions to maintain the population diversity in MOPSO. However, this algorithm is prone to fall in local optima due to the mutation operator (Sedarous, El-Gokhy, & Sallam, 2018). Mahmoodabadi, Bagheri, Nariman-Zadeh, and Jamali (2012) integrated the MOPSO with convergence and divergence operators. The divergence operator is used as a simple controlled mutation. Leung, Ng,

Cheung, and Lui (2014) enhanced the exploration ability of the proposed MOPSO by using mutation operator. In Fan, Chang, and Chuang (2015) and Wei et al. (2019) the polynomial mutation was used to preserve the diversity of the swarm in MOPSO during the search process.

Hassanzadeh and Rouhani (2010) integrated mutation operator with the proposed MOGSA to maintain the population diversity. In Bhowmik and Chakraborty (2015), the Opposition-based learning (OBL) approach introduced by (Tizhoosh, 2005) has been integrated with the proposed MOGSA to update the gravitational acceleration to improve the convergence of the solutions. To maintain the population diversity and prevent the algorithm from trapping in local optima, two mutation operators, namely, sign and reordering mutation has been used. In Zellagui et al. (2017), the same two mutation operators proposed in Bhowmik and Chakraborty (2015) have been used with the proposed MOGSA to maintain the population diversity. And the OBL approach has been omitted from the algorithm.

In Al Moubayed, Petrovski, and McCall (2010), instead of mutation operator, they used an information exchange method with the MOPSO to help the algorithm escape from local optima and avoiding premature convergence. In Zhu et al. (2017) an immune-based evolutionary strategy is used to maintain the population diversity and improve the convergence of the algorithm. In Luo, Huang, Li, and Gao (2019) a personal best and multi-global best selection mechanism were used to exploration ability and balance convergence and diversity of the proposed MOPSO. In Pan, Wang,

Guo, and Wu (2018) the velocities of particles were analyzed during the optimization process. In the proposed MOPSO, the diversity of swarm was preserved by performing a diversity enhancing process based on the velocity analysis. In Mellal and Zio (2019), the Lévy flight was used with the MOPSO to maintain the population diversity. In Ünal and Kayakutlu (2020), instead of the mutation operator, the random immigrants method was used with the MOPSO to maintain the population diversity.

Akbari et al. (2012) used a grid-based approach in the proposed MOABC to maintain the diversity of the population in the external archive. In Zhong, Xiang, and Liu (2014) a self-adaptive grid was used to preserve the population maintain diversity in the archive. The Lévy flight was used to improve the local search ability of the algorithm. In Bai and Liu (2016) the Boltzmann selection mechanism was used with the MOABC to adjust dynamically the probability of unemployed bees following (selecting) the employed bees. According to authors, this mechanism helps the algorithm to escape from local optima. In Mahmoodabadi and Shahangian (2019), the Euclidean distance was used to maintain the population diversity in the external archive. In the proposed MOABC, if the Euclidean distance between two individuals is less than a neighborhood radius, then one of them will be omitted.

In Tsai et al. (2014), the MOFA has been integrated with the non-dominated sorting approach and an improved crowding distance to select best non-dominated solutions and maintain population diversity. In Kishor et al. (2016), the MOABC algorithm was proposed. The algorithm uses the non-dominated sorting approach (Deb et al., 2002a)

to maintain population diversity. In Zhou and Yao (2017) the non-dominated sorting approach (Deb et al., 2002a) with crowding distance was used with the MOABC to maintain the diversity of solutions in the external archive. Jangir and Jangir (2018) proposed a MOGWO and used the non-dominated sorting approach and crowding distance to maintain the population diversity. Chen et al. (2019) proposed a MOBA with the non-dominated sorting approach and crowding-distance to maintain the population diversity and guide the search process. Table 2.2 summarizes the methods that have been integrated with the population update strategy of a MOSI-based algorithm.

Table 2.2
Summary of the population update strategy in MOSI-based algorithms

Reference	Algorithm	Diversity preservation method
Coello et al. (2004)	MOPSO	Mutation
Sierra and Coello (2005)	MOPSO	Mutation
Peng and Zhang (2008)	MOPSO	Mutation
Mahmoodabadi et al. (2012)	MOPSO	Mutation
Leung et al. (2014)	MOPSO	Mutation
Fan et al. (2015)	MOPSO	Mutation
Wei et al. (2019)	MOPSO	Mutation
Hassanzadeh and Rouhani (2010)	MOGSA	Mutation
Bhowmik and Chakraborty (2015)	MOGSA	Mutation
Zellagui et al. (2017)	MOGSA	Mutation
Tsai et al. (2014)	MOFA	Non-dominated sorting approach with crowding distance
Kishor et al. (2016)	MOABC	Non-dominated sorting approach with crowding distance
Zhou and Yao (2017)	MOABC	Non-dominated sorting approach with crowding distance
Jangir and Jangir (2018)	MOGWO	Non-dominated sorting approach with crowding distance
Chen et al. (2019)	MOBA	Non-dominated sorting approach with crowding distance
Al Moubayed et al. (2010)	MOPSO	information exchange method
Zhu et al. (2017)	MOPSO	Immune-based evolutionary strategy

Luo et al. (2019)	MOPSO	personal best and multi-global best selection mechanism
Pan et al. (2018)	MOPSO	Diversity enhancing process based on the velocity analysis
Mellal and Zio (2019)	MOPSO	Lévy flight
Ünal and Kayakutlu (2020)	MOPSO	Random immigrants
Akbari et al. (2012)	MOABC	Grid-based approach
Zhong et al. (2014)	MOABC	Self-adaptive grid
Bai and Liu (2016)	MOABC	Boltzmann selection mechanism
Mahmoodabadi and Shahangian (2019)	MOABC	Euclidean distance
Coello and Lechuga (2002)	MOPSO	-
Yang (2012a)	MOBA	-
Yang (2013)	MOFA	-
Mirjalili (2016)	MODA	-
Mirjalili et al. (2016)	MOGWO	-
Mohamed et al. (2016)	MOGWO	-
Mirjalili et al. (2017)	MOSSA	-
Kumawat et al. (2017)	MOWOA	-

Most of the studies improved the population update strategy of a MOSI-based algorithm by integrating a diversity preservation method, such as perturbation (mutation or disturbance) (Hu & Yen, 2013; Ünal & Kayakutlu, 2020) and non-dominated sorting approach with crowding distance. This integration aims to maintain the diversity of the population during the search process and overcome the premature convergence (Hu & Yen, 2013; Ünal & Kayakutlu, 2020). However, according to Al Moubayed et al. (2010), the usage of mutation operator leads to a very high convergence speed. However, such convergence speed could be a disadvantage in solving MOPs, because it may lead to a false Pareto front due to falling into local optima (Coello et al., 2004).

Other MOSI-based algorithms integrate a single objective optimization algorithm with a MOO approach to handle multiple objectives, without any modification on the

original mathematical model, such as MOABC (Akbari et al., 2012), MOFA (Yang, 2013), MOBA (Yang, 2012a), MODA (Mirjalili, 2016), MOGOA (Mirjalili et al., 2018) and MOGWO (Mirjalili et al., 2016). Therefore, these algorithms inherit the disadvantages of a single objective optimization algorithm in terms of falling into local optima due to loss of population diversity which leads to premature convergence (Asih et al., 2017; Neumann et al., 2018; Ni & Deng, 2014; Xu et al., 2015). In this study, the Pareto dominance and non-dominated sorting approaches will be used in the proposed population update strategy to handle multiple objectives and preserve the population diversity.

2.5 Classification of parameters adjustment strategies

In any metaheuristic, the adjustment of its parameters plays an important role in controlling the search process. According to Eiben et al. (2007) parameter configuration directly affects the quality of the final solution. Therefore, it is necessary to determine the most promising parameters configuration of the algorithm being used. The proper parameters configuration helps the algorithm to avoid a risk of being trapped in local optima and the premature convergence (Huang et al., 2019; X.-S. Yang et al., 2019).

In Eiben et al. (2007) a taxonomy of parameter adjustment is proposed. The parameter setting is divided into two main categories, namely, offline adjustment (parameter tuning) and online adjustment (parameter control), as shown in Figure 2.8.

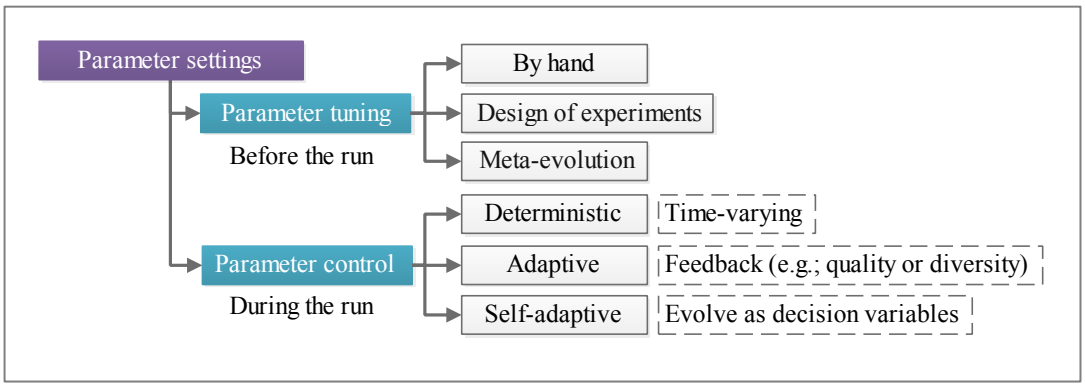


Figure 2.8. Global taxonomy of parameter setting (Adapted from Kramer (2010))

Offline adjustment is characterized by strategies that use static and previously adjusted parameter values. This category includes manual tuning, design of experiments and meta-evolution tuning (Kramer, 2010). Manual adjustment depends on user experience, where it modifies the parameter values before each execution of the algorithm. In the design of experiments, a set of parameter values are tried and through analysis of the results obtained by combining the parameter values, the best configuration to be used is defined. However, the trial and error method to adjust the parameters and choose the best configuration to be used requires time and knowledge on both the optimization method and the problem addressed. The meta-evolution consists of applying an algorithm to evolve the parameter values. The goal is to obtain standard values for the parameters that can be recommended for future executions. However, such recommendations cannot be generalized to all types of problems, as the best parameters for one optimization problem may not necessarily be the best parameters for another optimization problem. Furthermore, these techniques require too many computing resources due to the large number of combinations (Marinakis, Marinaki, & Migdalas, 2015).

In online control, an important feature is that the value of a parameter varies according to the search stage. Having online control during the process of optimization can significantly improve the final result (Huang et al., 2019). In online control, the value of the parameters is modified during the execution of the algorithm. Online control is divided into three categories: deterministic, adaptive and self-adaptive.

Deterministic control is performed through predetermined deterministic rules that update the value of the parameter during execution, without using any feedback from the search process. These methods depend on the time step, as an example for these methods, time varying strategies, such as random and linear decreasing strategies. In time-varying strategies, the value of a parameter changes constantly within a certain range and according to certain rules, during the whole iteration process, and thus presents different values. However, deterministic parameter control has similar limitations to parameter tuning where the rule of a control parameter has to be defined a priori and does not take any notion of the actual progress of the search. Furthermore, these methods lead to the loss of a lot of information gained during the search process.

In the self-adaptive approach, the parameters are coded together with the solution vector. In other words, the parameters evolved at run time by the algorithm itself during the optimization process (Eiben et al., 2007; Stützle et al., 2012). However, including a parameter values in the search space will lead to an increase in the size of the search space; thus causing an increase in the complexity of the optimization problem and slow convergence toward the global optima (Eiben et al., 2007). Another

drawback of self-adaptive control is that the search for optimal parameter values may be subject to premature convergence (Aleti & Moser, 2016).

In the adaptive parameter control approach, parameters are adjusted according to the state of the search. This approach uses feedback from the search process to determine the value of parameters. These feedbacks include but are not limited to the fitness value, diversity of population and distance to the relative position (Zhang et al., 2012). The field of adaptive parameter control has become an active research area where different techniques have been proposed. The goal is to free the user from the task of selecting control parameters and, instead, entrust this responsibility to the algorithm itself, in which it can decide upon and adapt the appropriate values of the control parameters without user assistance. Therefore, many studies have been conducted to adjust the parameters of metaheuristics, based on this method (Parpinelli et al., 2019).

2.6 Parameters adjustment strategies in MOSI-based metaheuristics

In general, a MOSI-based algorithm extends a single objective optimization algorithm to simultaneously optimize multiple objectives (Alhammadi & Romagnoli, 2004). Therefore, they used same parameters to control the search process. Although the parameters configuration has a great impact on the performance of an algorithms, to the best knowledge of the author, it has received little attention in the MOSI-based algorithm literature.

Mohamed et al. (2016) proposed a nonlinear parameter adjustment strategy to control the value of parameter *a* in the MOGWO algorithm. However, the proposed strategy also depends on other parameters, which were used to control the convergence of the algorithm. The value of these parameters is fixed during the optimization process. In Wei et al. (2019) an improved non-linear cosine-adjusted inertia weight was proposed. The proposed method is used to control the exploitation and exploration of MOPSO algorithm. This helps in preventing the algorithm from falling in local optima. Chen et al. (2019) proposed a nonlinear strategy to adjust the value of inertia weight coefficient to improve the convergence of the MOPSO algorithm. In Zellagui et al. (2017), a linear strategy to calculate the value of initial gravitational constant is proposed. The original GSA algorithm uses a fixed initial gravitational constant (Rashedi et al., 2009), which would lead to poor convergence. This parameter is used to calculate the gravitational constant, which play an important rule in controlling the exploration capabilities of the algorithm (de Moura Oliveira, Oliveira, & Cunha, 2017). However, the proposed strategy depends on two parameters, namely the initial value of inertial coefficient and final value of inertial coefficient, which were set as constant values.

Yang and Ji (2016) proposed a method to dynamically update the step size along the swimming direction of the bacterium in multi-objective bacterial foraging optimization. According to Yang and Ji (2016), the proposed algorithm helps in improving the convergence of the algorithm. Wei et al. (2018) proposed a method to adaptively adjust the inertia weight and acceleration coefficients based on the *Sigmoid* function. These parameters are used to control the exploration and exploitation process

in MOPSO (Wei et al., 2018). According to Wei et al. (2018) this aims to improve the velocity update formula, which leads to improve the exploitation and exploration of the algorithm. Yu et al. (2020) proposed an adaptive parameter method to dynamically adjust the values of inertia weight and learning parameters in MOPSO algorithm which helps the algorithm to escape from local optima and avoid the premature convergence.

In the adaptive parameter control approach, the parameters values are adjusted according to feedback represents the state of the search space (Eiben et al., 2007). However, in Wei et al. (2018); Yu et al. (2020), the value of parameters were adjust based on a linear decreasing strategy, without receiving any feedback from the search space. Therefore, it cannot be considered as an adaptive strategy. Table 2.3 shows the parameters adjustment strategies used by the MOSI-based algorithms.

Table 2.3
Parameters adjustment strategies in MOSI-based algorithms

Reference	Algorithm	Parameter adjustment strategy
Coello et al. (2004)	MOPSO	x
Sierra and Coello (2005)	MOPSO	x
Peng and Zhang (2008)	MOPSO	x
Mahmoodabadi et al. (2012)	MOPSO	x
Leung et al. (2014)	MOPSO	x
Fan et al. (2015)	MOPSO	x
Wei et al. (2018)	MOPSO	Nonlinear cosine-adjusted inertia weight
Wei et al. (2019)		
Hassanzadeh and Rouhani (2010)	MOGSA	x
Bhowmik and Chakraborty (2015)	MOGSA	x
Zellagui et al. (2017)	MOGSA	Linear
Tsai et al. (2014)	MOFA	x
Kishor et al. (2016)	MOABC	x
Zhou and Yao (2017)	MOABC	x
Jangir and Jangir (2018)	MOGWO	x
Chen et al. (2019)	MOBA	Nonlinear

Al Moubayed et al. (2010)	MOPSO	x
Zhu et al. (2017)	MOPSO	x
Luo et al. (2019)	MOPSO	x
Pan et al. (2018)	MOPSO	x
Mellal and Zio (2019)	MOPSO	x
Ünal and Kayakutlu (2020)	MOPSO	x
Akbari et al. (2012)	MOABC	x
Zhong et al. (2014)	MOABC	x
Bai and Liu (2016)	MOABC	x
Mahmoodabadi and Shahangian (2019)	MOABC	x
Coello and Lechuga (2002)	MOPSO	x
Mirjalili et al. (2016)	MOGWO	x
Mohamed et al. (2016)	MOGWO	Nonlinear
Kumawat et al. (2017)	MOWOA	x
Yang (2012a)	MOBA	x
Mirjalili (2016)	MODA	x
Yang and Ji (2016)	bacterium in multi-objective bacterial foraging optimization	Dynamic
Yu et al. (2020)		Linear decrease

In Table 2.3 the cross sign (X) indicates that the MOSI-based algorithm used the original parameter adjustment strategy of the single objective optimization algorithm. Most of the MOSI-based algorithms, used the same parameters adjustment strategy of the single objective optimization algorithm to change the value of parameters.

In the MOSI-based algorithms, most of the proposed parameters adjustment strategy did not take in consideration the state of search space during the optimization process. Instead, a linear and non-linear decreasing strategy to adjust the values of parameters is used. However, these strategies may not reflect the requirements of actual optimization process, which leads to poor convergence toward the true Pareto front. This study, aims to improve the performance of HHMO algorithm by proposing an adaptive parameter adjustment strategy for the convergence parameter, *a*. The

proposed strategy aims to adaptively coordinate the transition between the exploration and exploitation of the search space during the optimization process.

2.7 Classification of initial population generator methods

All SI-based metaheuristics are population-based, which start by initialization of a set of search agents (potential solutions) for an optimization problem. In SI-based metaheuristics, the initial population affects the convergence toward global optima and the quality of the solution (Altinoz et al., 2014; Kazemzadeh Azad, 2018; Tu et al., 2019). Therefore, the population initialization method is important in improving the performance of optimization algorithms (Altinoz et al., 2014; Kazemzadeh Azad, 2018; Tu et al., 2019). The initialization method can be classified into three main classes, namely, randomness, compositionality and generality (Kazimipour, Li, & Qin, 2014), as shown in Figure **Error! Reference source not found.**.

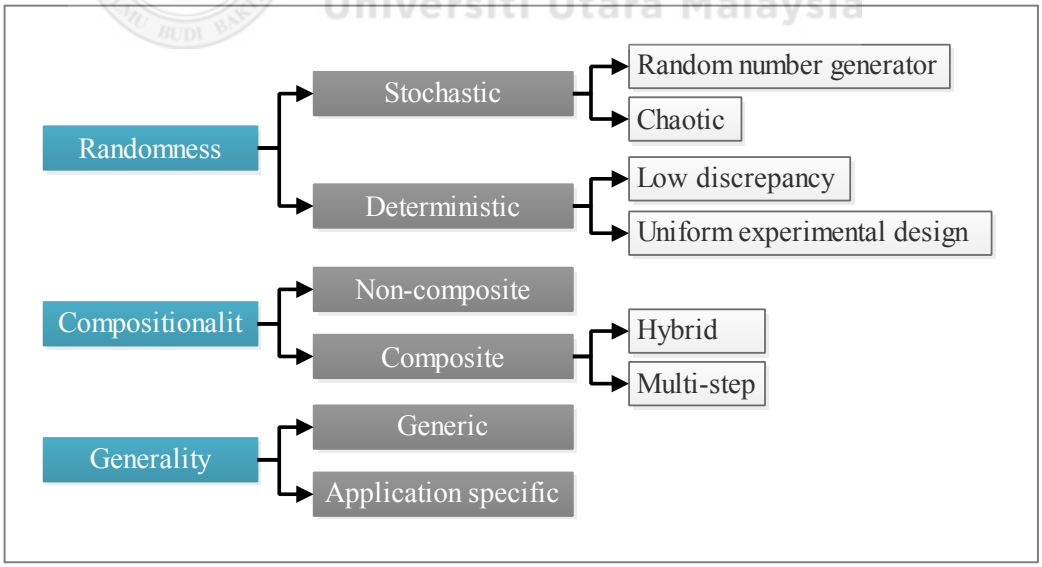


Figure 2.9. Taxonomy of population initialization methods (Adapted from Kazimipour et al. (2014))

Based on the more specific criteria, each category is further divided into several subcategories. These criteria are determined based on two facts. The first criterion is generality of a technique in which a technique can be applied to all types of optimization problems or is suitable for solving a specific problem. Second, a unique and independent aspect, such as whether a technique is random or not.

2.7.1 Randomness

Based on the degree of dependence on the initial seed, the random initialization techniques can be classified into stochastic and deterministic techniques. The stochastic techniques can be divided into two subgroups, namely pseudo random number generator and chaotic number generator (CNG).

The pseudo random number generator is widely used in applications to generate random numbers due to implementation simplicity in any programming language. However, PRNGs suffer from the curse of dimensionality because they cannot produce a perfect uniform distribution of points, especially with an increase in size of the search space. This will affect the quality of the generated points (Kazimipour, Li, & Qin, 2013).

The CNG techniques mimic the behaviour of the dynamic system to produce unpredicted points in the search space and they are sensitive to initial conditions. ergodicity, randomness and regularity are the main characteristics of chaotic systems. The chaotic number generator techniques are configured through a chaotic map, built

by a set of user-defined parameters (Kazimipour et al., 2014). The main disadvantage of chaotic map techniques is they a parameter-depended and very sensitive to its initial conditions (Tian, 2017).

In deterministic techniques, the initial method always produces the same population and ignores the initial seeds. Deterministic techniques, also known as low-discrepancy techniques, are designed to provide evenly distributed points in the entire search space. These techniques attract more attention due to uniformity and problem-independent characteristics, which leads to better convergence. In general, deterministic techniques are divided into two categories, namely, uniform experimental design and quasi-random generator (QRNG) (Kazimipour et al., 2014).

The uniform experimental design technique looks for points to be evenly distributed in a given interval. uniform experimental design is widely used in industrial and computer simulation design. The QRNGs are designed to produce low-discrepancy sequences with high levels of uniformity in high-dimensional space (Levy, 2002). Compared to PRNGs, the sequences generated by QRNGs are more evenly distributed within hypercube than sequences generated by PRNGs. Furthermore, the low-discrepancy techniques are superior, in terms of discrepancy, over PRNGs. This is due to the high correlation between the generated points, where the next point "knows" where the previous points are. Examples for QRNG techniques include Sobol (Sobol', 1967), Halton (Halton, 1960), Hammersley (Owen, 2019) and Kronecker (Larcher & Niederreiter, 1993) sequences.

Broadly, low-discrepancy techniques generalise the one-dimensional Van der Corput sequences, also known as radical inverse function (Faure, Kritzer, & Pillichshammer, 2015). In a Halton sequence, each dimension is a b_n Van der Corput sequence, where (b_1, ... b_n) is a prime number. The Hammersley sequence is very similar to Halton, except, the first dimension is changed to $\frac{i}{N}$, where i and N are the index and total number of sample points, respectively (Pharr, Jakob, & Humphreys, 2016). Discrepancy of Hammersley is slightly lower than Halton. However, Halton can generate an unlimited number of points without restriction, whereas the Hammersley point set can only generate a fixed number of points (Pharr et al., 2016). Unlike Halton and Hammersley, each dimension of the Sobol sequence consists of a radical inverse with a base of 2. The Sobol sequence can generate unlimited samples as needed, which is very suitable for progressive sampling.

In Roberts (2018), another low-discrepancy quasi-random sequence, called R-sequence, has been proposed. The R-sequence is developed based on extensions of the golden ratio (Slater, 2019). R-sequence is a method of constructing (infinite) sequence points in a deterministic manner that reduces the likelihood of clustering (discrepancy) whilst still ensuring that the entire space is uniformly covered. According to Roberts (2018), this sequence generates more evenly distributed points than any of the other state-of-the-art techniques, such as, Sobol, Halton and Kronecker sequences, as shown in Figure 2.10.

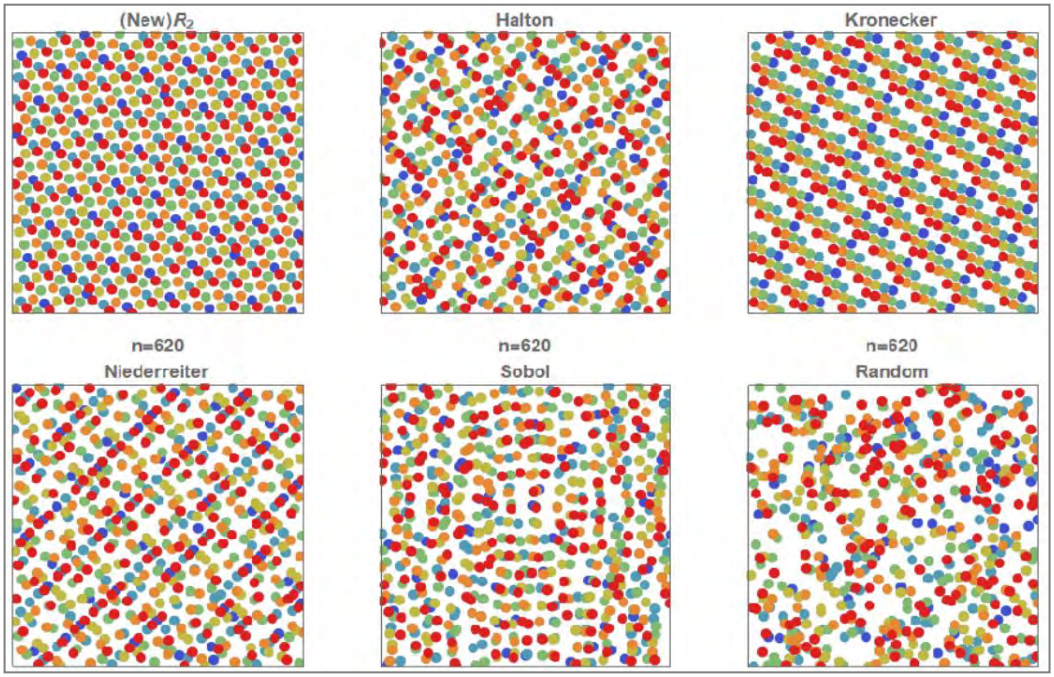


Figure 2.10. Comparison of various low discrepancy quasi-random sequences (Adapted from Roberts (2018))

As shown in Figure 2.10, the distributions of two-dimensional R-sequence and other QRNG are far more uniform than RNG. The discrepancy of R-sequences is lower than Halton, Sobol, Niederreiter, and Kronecker sequences. Furthermore, in contrast to other QRNG, the R-sequence does not require any selection of basis parameters. Improper selection of these parameters can lead to degeneracy (Roberts, 2018).

2.7.2 Compositionality

Compositionality techniques involve a number of standalone procedures. The population initialization techniques can be divided into composite and non-composite techniques. Non-composition techniques produce a population in a single step. The use of this technique can be stochastic, deterministic, generic or application-specific. The composition techniques produce populations in more than one step. These

techniques, in turn, can be divided into two categories, namely, multi-step and hybrid techniques.

Multi-step techniques are composed of two or more techniques in which at least one of them cannot be used autonomously. These techniques generally refine the previously generated population in the subsequent steps. The two-step initialization technique includes two phases. In the first phase, an algorithm generates the initial population and then uses some criteria (such as the fitness function value) to improve it in the second phase. The hybrid technique is generally a combination of some basic techniques, where each procedure can be applied separately, as a non-compositional technique. However, in general, hybrid techniques theoretically inherit the advantages and disadvantages of the basic techniques from which they are adopted.

2.7.3 Generality

These population initialization techniques can be applied directly to various optimization problems. The population initialization techniques are again divided into two categories, namely, generic and application-specific techniques.

The generic techniques assume that the given optimization problem is a black-box. Therefore, they can be applied in all types of optimization problems. In the absence of such prior knowledge about the problem, generic population initialization techniques can be used easily and effectively. All previously described techniques belong to the generic category. On the other hand, application-specific techniques are specifically

designed for particular real-world problems. This help in producing promising results and increasing convergence speed. However, they are not effective and applicable in other areas. Furthermore, these techniques require high levels of experience in the application domain (Kazimipour et al., 2014).

2.8 Initial Population Generator methods in MOSI-based Metaheuristics

In the area of MOSI-based metaheuristics, most of the studies have focused on introducing MOO algorithms based on the existing single-objective algorithms. In some of these algorithms, different MOO approaches have been used to handle multiple objectives. Other studies improved the performance of algorithms by integrating different methods that help in maintaining the population diversity during the optimization process. Initial population of candidate solutions can affect the performance of algorithm, in terms of convergence and the quality of the final solution (Tu et al., 2019). To the best of the author's knowledge, small effort has been given to study this field. Orouskhani, Teshnehlab, and Nekoui (2018) proposed the OBL (Tizhoosh, 2005) method to initialize the population of the multi-objective cat optimization algorithm. In Wang, Li, Wang, and Li (2019b), the initial population of whales in the MOWOA was generated using OBL method. In the proposed method, the population is generated using RNG and divided into two halves. The OBL is applied on the second half. Then, the two subpopulations are combined to generate population with a specific size.

The OBL method by (Tizhoosh, 2005) has also been widely used with the RNG method to initialize the population of single objective optimization algorithms. Rashid and Baig (2010) integrated the OBL with the PSO algorithm to generate the population of a swarm. The study showed that the OBL-based PSO can deliver a better performance as compared to the standard PSO. In Farooq, Ahmad, and Hameed (2017) the OBL approach was used to initialize the population of a swarm in a PSO algorithm. According to Rashid and Baig (2010), the proposed algorithm significantly improved the performance of PSO and it became more robust when compared to other contending algorithms. Verma, Aggarwal, and Patodi (2016) incorporated OBL initialization to generate a candidate solution for the firefly algorithm. Wen (2016) utilized an OBL approach to initialize the population of GWO. The proposed algorithm is able to provide very competitive results compared to other algorithms. Bao, Jia, and Lang (2019) employed OBL to generate the initial population of dragonflies in the dragonfly algorithm. The results showed that the proposed method has superior performance compared with other algorithms. In Jain and Saxena (2019), the OBL approach has been used in moth-flame optimization to initialize half of the population whereas the other half is initialized by using a classical RNG method.

In other studies, the OBL and chaotic map have been used to initialize the population of single objective optimization algorithms. Gao, Liu, and Huang (2012) employed a chaotic OBL population initialization instead of a traditional random initialization for PSO to improve its performance. According to authors, the proposed method, to some extent can achieve certain success compared to the PSO algorithm with usual random

initialization under the same conditions. Kuang, Jin, Xu, and Zhang (2014) employed the tent chaos map and the OBL method to generate an initial population for the ABC algorithm. The results showed that the proposed method improved the performance of ABC, in terms of increasing the population diversity and accelerating the convergence rate. In Gao, Huang, Wang, Liu, and Qin (2016) a chaotic OBL approach is employed to generate a population of bees in ABC algorithm. Ibrahim, Elaziz, and Lu (2018) employed chaotic logistic map and the OBL to initialize the candidate solutions of a GWO algorithm. The results showed that the proposed approach provides an effective performance in solving optimization problems.

Some studies used different types of chaotic map to initialize the population of algorithms. Tian (2017) utilized two types of chaotic map, namely tent map and logistic map to generate uniformly distributed particles in PSO. The proposed method provides good performance in terms of its stability, the quality of the final solutions and the convergence speed.

The quasi-random sequences also have been used instead of the RNG method. In Weerasinghe, Chi, and Cao (2016), the population of swarm in PSO algorithm was initialized using scrambled optimal Halton sequence. The proposed method was compared with the Halton sequence and the scrambled optimal Halton sequence after dropping the first 5000 points. According to the authors, all sequences showed similar results. Bangyal, Ahmad, Rauf, and Pervaiz (2018) proposed an improved bat algorithm in which the population of bats is initialized using the torus quasi-random

sequence. According to Bangyal et al. (2018), the proposed algorithm outperforms the original BA algorithm.

Other different methods have been employed to initialize the population of candidate solutions. In Zhou, Dai, Fang, Chen, and Tan (2008), the orthogonal design was used to generate the initial population of the proposed MOPSO. Cazzaniga, Nobile, and Besozzi (2015) proposed a strategy to initialize swarm of particles based on the logarithmic and lognormal distributions. According to the authors, the proposed strategy improves the performance of PSO, the results showed that lognormal and logarithmic distribution outperform other distributions, namely, uniform and normal distributions, which are commonly used in standard PSO. In Su et al. (2017), an orthogonal Latin squares approach is utilized to initialize a population of bees in a comprehensive learning artificial bee colony algorithm. Table 2.4 summarises the initial population generator methods applied in the MOSI and single objective SI-based metaheuristics.

Table 2.4
Summary of initial population generator methods in MOSI and single objective-based metaheuristics

Reference	Algorithm	Method
Rashid and Baig (2010)	PSO	OBL
Verma et al. (2016)	FA	OBL
Wen (2016)	GWO	OBL
Farooq et al. (2017)	PSO	OBL
Ibrahim et al. (2018)	GWO	OBL
Orouskhani et al. (2018)	Multi-objective cat swarm	OBL
Jain and Saxena (2019)	moth-flame optimization	OBL
Bao et al. (2019)	DA	OBL
W. L. Wang et al. (2019b)	Multi-objective whale	OBL

Gao et al. (2012)	PSO	Chaotic OBL
Kuang et al. (2014)	ABC	Chaos map & OBL
Gao et al. (2016)	ABC	Chaotic OBL
Tian (2017)	PSO	Chaotic map
Yang, Gao, Liu, and Song (2015)	PSO	QRNG
Weerasinghe et al. (2016)	PSO	QRNG
Bangyal et al. (2018)	BA	QRNG
Cazzaniga et al. (2015)	PSO	Logarithmic and lognormal distributions
Zhou et al. (2008)	MOPSO	Orthogonal design
Su et al. (2017)	ABC	Orthogonal Latin squares
Coello and Lechuga (2002)	MOPSO	RNG
Coello et al. (2004)	MOPSO	RNG
Sierra and Coello (2005)	MOPSO	RNG
Peng and Zhang (2008)	MOPSO	RNG
Hassanzadeh and Rouhani (2010)	MOGSA	RNG
Al Moubayed et al. (2010)	MOPSO	RNG
Mahmoodabadi et al. (2012)	MOPSO	RNG
Akbari et al. (2012)	MOABC	RNG
Yang (2012a)	MOBA	RNG
Zhong et al. (2014)	MOABC	RNG
Leung et al. (2014)	MOPSO	RNG
Tsai et al. (2014)	MOFA	RNG
Fan et al. (2015)	MOPSO	RNG
Bhowmik and Chakraborty (2015)	MOGSA	RNG
Bai and Liu (2016)	MOABC	RNG
Mirjalili (2016)	MODA	RNG
Mirjalili et al. (2016)	MOGWO	RNG
Kishor et al. (2016)	MOABC	RNG
Zellagui et al. (2017)	MOGSA	RNG
S/MOGWO Mohamed et al. (2016)	MOGWO	RNG
Zhu et al. (2017)	MOPSO	RNG
(Kumawat et al., 2017)	MOWOA	RNG
Zhou and Yao (2017)	MOABC	RNG
Jangir and Jangir (2018)	MOGWO	RNG
Pan et al. (2018)	MOPSO	RNG
Wei et al. (2019)	MOPSO	RNG
Mahmoodabadi and Shahangian (2019)	MOABC	RNG
Luo et al. (2019)	MOPSO	RNG
Mellal and Zio (2019)	MOPSO	RNG
Got, Moussaoui, and Zouache (2020)	MOWOA	RNG

Chen et al. (2019)	MOBA	RNG
Ünal and Kayakutlu (2020)	MOPSO	RNG

Most of the MOSI-based algorithms used the RNG method to initialize the population of candidate solutions. However, with this method the generated candidate solutions are not well distributed in the search space, which affect the performance of an algorithm (Digehsara et al., 2020; Jana et al., 2018; Maaranen et al., 2004). Since there is no a priori knowledge about the location and the number of local optima in the fitness space, the generated solutions should be distributed as much as possible in the search space. If the initial population is well distributed, then the possibility of finding a good solution will be increased (Hamdan & Qudah, 2015; Talbi, 2013).

In the area of MOSI-based metaheuristics, several methods have been employed instead of the RNG, namely OBL and orthogonal design. However, the orthogonal design is a parameter-depended method (Kazimipour et al., 2014). The number of levels plays a very important role in determining the performance of the method (Kazimipour et al., 2014). The OBL method has been used with the RNG method to initialize the population of candidate solutions. However, the main disadvantage of this method is the computational complexity, which increases with the number of dimensions (Liu, Xu, Ding, & Li, 2015).

Most of the single objective SI-based metaheuristics employed the RNG method with OBL method. The partial OBL has been used instead of full OBL to initialize the population. The partial OBL is a new form of OBL which was first proposed by Hu, Bao, and Xiong (2014) to improve the performance of differential evolution algorithm.

In the proposed algorithm, the partial OBL was used during the evolution process. However, according to Tizhoosh (2005), the usage of OBL is recommended only in the early stages of optimization because as the learning continues it becomes disadvantageous. In Si and Dutta (2019), the partial OBL was used to initialize the swarm of particles for PSO algorithm. The position of particles is first, initialize using RNG method. Then, the partial OBL is applied to calculate the opposites of position and velocity.

Initialization methods based on the chaos map highly depends on the initial parameters, which significantly affect the performance of an algorithm. The QRNG method was employed in several algorithms to generate the population of solutions which showed superior performance compared to the algorithms with a classical method (Bangyal et al., 2018) Weerasinghe et al. (2016) Yang et al. (2015).

The initializing methods in Table 2.4 were used to initialize the population of single objective optimization algorithm and these methods did not take into consideration the characteristics of the MOO. In general, several studies have been conducted in the field of initial population for MOO (Hamdan & Qudah, 2015; Poles et al., 2009; Wang, Li, & Wang, 2019a; Yang, You, Zhao, Dou, & Guo, 2019). In most MOSI-based algorithms, the population of potential solutions is initialized by using the RNG method (Akbari et al., 2012; Coello & Lechuga, 2002; Du et al., 2019; Hassanzadeh & Rouhani, 2010; Kumawat et al., 2017; Mirjalili et al., 2016; Yang, 2012a, 2013).

In this context, this thesis aims to improve the performance of HHMO algorithm by proposing an initial population generator method. The proposed method will be developed based the two-step initialization technique. In the first step, instead of the RNG method, the R-sequence introduced by Roberts (2018); (Slater, 2019) will be used to generate a low discrepancy sequence. In the second step, a partial OBL method will be used to preserve the initial population diversity. The proposed initial population generator method also will take in consideration the characteristics of the MOO to ensure good convergence toward true Pareto front.

2.9 Multi-objective Benchmark Optimization Problems

Benchmark optimization problems with different features have been wildly used in the literature to evaluate the performance of MOO algorithms. These benchmark problems include test functions and real-world problems.

2.9.1 Test functions

The test functions are normally used in the literature to validate the performance of a MOO algorithm or to compare two or more algorithms. Compared to the real-world problems, the test functions has advantages of their true Pareto front is known, their difficulty degree can be controlled, and in most problems, the number of objectives and decision variables can also be controlled (Tanabe & Ishibuchi, 2020). Several test problems have been used in the literature over the years. The most widely used are the ZDT and DTLZ families (Elarbi, Bechikh, Ben Said, & Datta, 2017).

The ZDT family were developed by Zitzler, Deb, and Thiele (2000) and their names are derived from the initials of the authors' names. This family includes ZDT1-ZDT6 problems and their main features include, convexity, non-convexity, disconnected, and multi-modality boundaries. The problem of this family cannot be scalaraized to more than two objective functions (Meneghini, Alves, Gaspar-Cunha, & Guimarães, 2020). Another widely used test problem is the Deb, Thiele, Laumanns and Zitzler (DTLZ) (Deb, Thiele, Laumanns, & Zitzler, 2005) family, which includes the test problems names DTLZ1-DTLZ7. These test problems encapsulate special characteristics, such as multi-modality, non-convexity and discontinuity which are known to generally cause difficulties to most multi-objective metaheuristics (Coello, Lamont, & Van Veldhuizen, 2007; Radulescu, López-Ibánez, & Stützle, 2013). Compared to the ZDT family, the DTLZ family can be scaled to any number of objectives and decision variables and the desired Pareto fronts are known which can be analytically determined (Meneghini et al., 2020). Therefore, they represent a good basis for testing the credibility and efficiency of optimization algorithms in different situations (Coello et al., 2007).

The Walking-Fish-Group (WFG) family is one of the test problems that is attracting the attention of the community (Huband, Hingston, Barone, & While, 2006). This family includes WFG1-WFG9 problems. The characteristics of these problems include connected, non-degenerate, degenerate, and disconnected Pareto front (Zapotecas-Martínez, Coello, Aguirre, & Tanaka, 2018). The UF series proposed by Zhang et al. (2009) has been used in evaluating the performance of MOO algorithms. This family

includes UF1-UF10 MOPs that have convex, concave, disconnected, multi-modal and linear Pareto front characteristics. The UF1-UF7 involves two objectives and UF8-UF10 are three objectives problems (Ergul & Eminoglu, 2014).

These test problems have been widely used in in evaluating the performance of MOSI-based algorithms, as shown in Table 2.5. Most of the studies used ZDT, DTLZ and UF test problems in evaluating the performance of the MOSI-based optimization algorithms. The study presented in this thesis uses these problems to evaluate the performance of the proposed enhanced HHMO algorithm, and in the comparison with other algorithms.

Table 2.5
Test functions used in evaluating the performance of MOSI-based algorithms

Reference	Algorithm	Test Function
Zhou et al. (2008)	MOPSO	ZDT
Mahmoodabadi et al. (2012)	MOPSO	ZDT
Leung et al. (2014)	MOPSO	ZDT
Mirjalili (2016)	MODA	ZDT
Orouskhani et al. (2018)	Multi-objective cat swarm	ZDT
Fan, Wang, Cheng, Li, and Gu (2017)	MOPSO	ZDT
Acı and Gülcan (2019)	MODA	ZDT
Luo et al. (2019)	MOPSO	DTLZ
Allmendinger et al. (2008)	MOPSO	ZDT& DTLZ
Qiao, Zhou, and Yang (2018)	MOPSO	ZDT& DTLZ
Han, Sun, and Ling (2018)	MOPSO	ZDT& DTLZ
Jia and Zhu (2017)	MOPSO	ZDT& DTLZ
Yu and Zhang (2017)	MOPSO	ZDT& UF
Mirjalili et al. (2018)	MOGOA	ZDT& UF
Liu, Jiao, Ma, Ma, and Shang (2016)	MOPSO	ZDT, DTLZ& UF

Díaz-Manríquez et al. (2016)	MOPSO	ZDT, DTLZ& UF
Zhou and Yao (2017)	MOABC	ZDT, DTLZ& UF
W. L. Wang et al. (2019b)	Multi-objective whale	ZDT, DTLZ& UF
Wei et al. (2019)	MOPSO	ZDT, DTLZ& UF
AbdelAziz, Soliman, Ghany, and Sewisy (2019)	Multi-objective whale	ZDT, DTLZ& UF
Fan et al. (2015)	MOPSO	ZDT, DTLZ& WFG
Zhu et al. (2017)	MOPSO	ZDT, DTLZ, UF & WFG
Pan et al. (2018)	MOPSO	DTLZ & WFG
X. Zhang et al. (2018)	MOPSO	ZDT, DTLZ & WFG
Luo et al. (2017)	MOABC	DTLZ & UF
Kishor et al. (2016)	MOABC	UF
Bai and Liu (2016)	MOABC	UF
Mirjalili et al. (2016)	MOGWO	UF

2.9.2 Real-world problem

The real-word problems have been used by many researchers to determine the performance optimization algorithms. Most of the problems in the continuous domain are engineering problems (Stewart et al., 2008; Tanabe & Ishibuchi, 2020). Table 2.6 shows different MOPs in various application domains.

Table 2.6
Summary of engineering applications

Specific Applications	Reference	Variable
Beam design	(Deb & Sundar, 2006)	Continuous
	(Coello Coello & Christiansen, 1999)	Continuous
Truss design	(Coello et al., 2007; Keller, 2019; Stadler & Dauer, 1993)	Continuous
		Continuous
Disk brake design	(Tawhid & Savsani, 2018b)	Mixed
Optimal power flow (OPF)	(Berrouk et al., 2018; Frank & Rebennack, 2016)	Continuous

Gear train problem	(Deb & Srinivasan, 2006)	Integer
Helical compression spring	(Lampinen & Zelinka)	Mixed
Pressure vessel design	(Kannan & Kramer, 1994)	Mixed
Speed reducer design	(Coello et al., 2007; Farhang-Mehr & Azarm, 2002)	Mixed

As shown in Table 2.6, the real-world applications include various optimization problems. The beam design includes problems such as welded beam (Deb & Sundar, 2006) and I-beam (Coello Coello & Christiansen, 1999). Compared to the I-beam, the welded beam has been widely used in evaluation the performance of MOSI-based algorithms (Got et al., 2020; Janga Reddy & Nagesh Kumar, 2007; Kotinis, 2010; Kulkarni, 2017; Tawhid & Savsani, 2018b; X.-S. Yang, M. Karamanoglu, & X. He, 2014).

The truss design includes several problems (Coello et al., 2007; Keller, 2019; Stadler & Dauer, 1993), among them the two-bar and four-bat truss (Stadler & Dauer, 1993) have been used by many studies (Coello & Pulido, 2005; Got et al., 2020; Janga Reddy & Nagesh Kumar, 2007; Jangir & Trivedi, 2018; Kotinis, 2010; Tawhid & Savsani, 2018b) in evaluation the performance of MOSI-based algorithm.

The OPF is considered as one of the most important optimization problems in the area of power engineering (Frank & Rebennack, 2016). However, a few MOSI-based algorithms have been used to solve this problem (Mohamed et al., 2016).

The previous MOPS have continuous design variables, while other problems such as the pressure vessel, speed reducer (Coello et al., 2007; Farhang-Mehr & Azarm, 2002) and disk brake (Tawhid & Savsani, 2018b) design have integer and mixed variables.

2.10 Performance Metric for MOO

In general, the performance metrics in MOO is used to measure two criteria, namely the convergence and diversity of non-dominated solutions (Mohammadi, Omidvar, & Li, 2013). Several metrics have been proposed to evaluate the performance of MOO algorithms. According to Riquelme, Von Lücken, and Baran (2015), the most widely used are the HV (Zitzler & Thiele, 1999), the generational distance (GD), the epsilon indicator (Zitzler, Thiele, Laumanns, Fonseca, & Da Fonseca, 2003) and the inverted generational distance (IGD) (Coello & Cortés, 2005).

In Li, Deb, and Yao (2018b) the R-metric was proposed based on the multi-criterion decision making approach. The multi-criterion decision making approach is first used to pre-process the preferred efficient set, then the a regular performance metric, such as HV or IGD is used to evaluate the obtained solutions (Li et al., 2018b). Compared to regular performance metrics. the R-metric takes into consideration the preference information determined by the DM (Li et al., 2018b) and it has been used to evaluate the performance of both preference and non-preference based algorithms (Tang et al., 2020). It is worth to mention that, in the literature other preference-based performance metrics have been proposed (Mohammadi et al., 2013; Wickramasinghe, Carrese, & Li, 2010). However, these metrics have limitations that make them provide misleading

information to the DM (Li et al., 2018b). In this study, the R-IGD, R-HV, and epsilon metrics will be used to evaluate the performance of the proposed 2S-ENDSHHMO algorithm.

2.11 Summary

In the real-world, most optimization problems consist of two or more conflicting objectives in which improving an objective leads to the degradation of others. Compared to SOPs, solving a MOP is not a trivial task. This is mainly due to the high complexity of a problem, which generates many possible solutions. Given this difficulty, there is a growing interest in using the metaheuristic methods combined with MOP approaches to solve MOPs. This is due to that metaheuristics reach feasible solutions without having to list all possibilities. This class of MOP solving methods uses metaheuristics to generate and analyse different solutions as well as to determine the approximations of the Pareto front.

The HHMO algorithm is one of the recent MOSI-based algorithms, which have been developed based on a reference point approach. In this approach, instead of wasting time in exploring undesired regions, the algorithm focuses on a preferred region, determined by the DM. This helps reduce the computational cost and generate better solutions for a MOP. Although the HHMO algorithm can achieve good results for some MOP problems, its performance can be further improved. This can be achieved by enhancing the population update of hawks. The adaptive parameter control approach can be utilized to adjust the distance parameter control. Moreover, the

diversity of the initial population of hawks can be improved by employing another initial population generator method.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

The main goal of this study is to propose a two-step enhanced non-dominated sorting Harris's hawk multi-objective optimizer (2S-ENDSHHMO) algorithm, with the aim to overcome the limitations of HHMO and provide an effiecient MOO method. In this context, it is appropriate to introduce a framework to carry out this study. In this chapter, the main research processes and research framework that is used to conduct this study are described in Section 3.2. Stages 1, 2 and 3 of the framework are presented in Section 3.3, Section 3.4, Section 3.5 and Section 3.6, respectively. Section 3.7 describes the MOPs, followed by the application domain, which presented in Section 3.8. The performance measure is described in Section 3.9 followed by chapter summary in Section 3.10.

3.2 Research Framework

Developing the 2S-ENDSHHMO algorithm requires a careful understanding of the HHMO algorithm and identification of its limitations. Overcoming these limitations will lead to an effective MOO algorithm. To achieve this goal, this research was performed, based on four main processes, as shown in Figure 3.1.

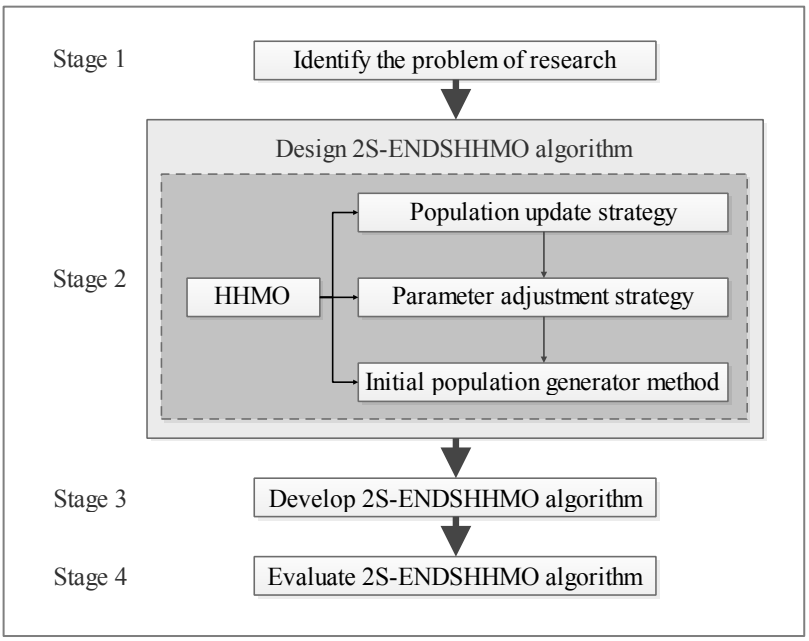


Figure 3.1. Main Research Processes

The framework organizes this study into four main stages. In the first stage the main problems that affect the performance of HHMO algorithm have been identified. The proposed population update strategy, parameter adjustment strategy and initial population generator method are formulated in the second stage. These strategies and method are integrated with the HHMO algorithm to produce the 2S-ENDSHHMO algorithm in the third stage. Finally, in the fourth stage, the performance of the 2S-ENDSHHMO algorithm has been evaluated. The research framework is illustrated in Figure 3.2.

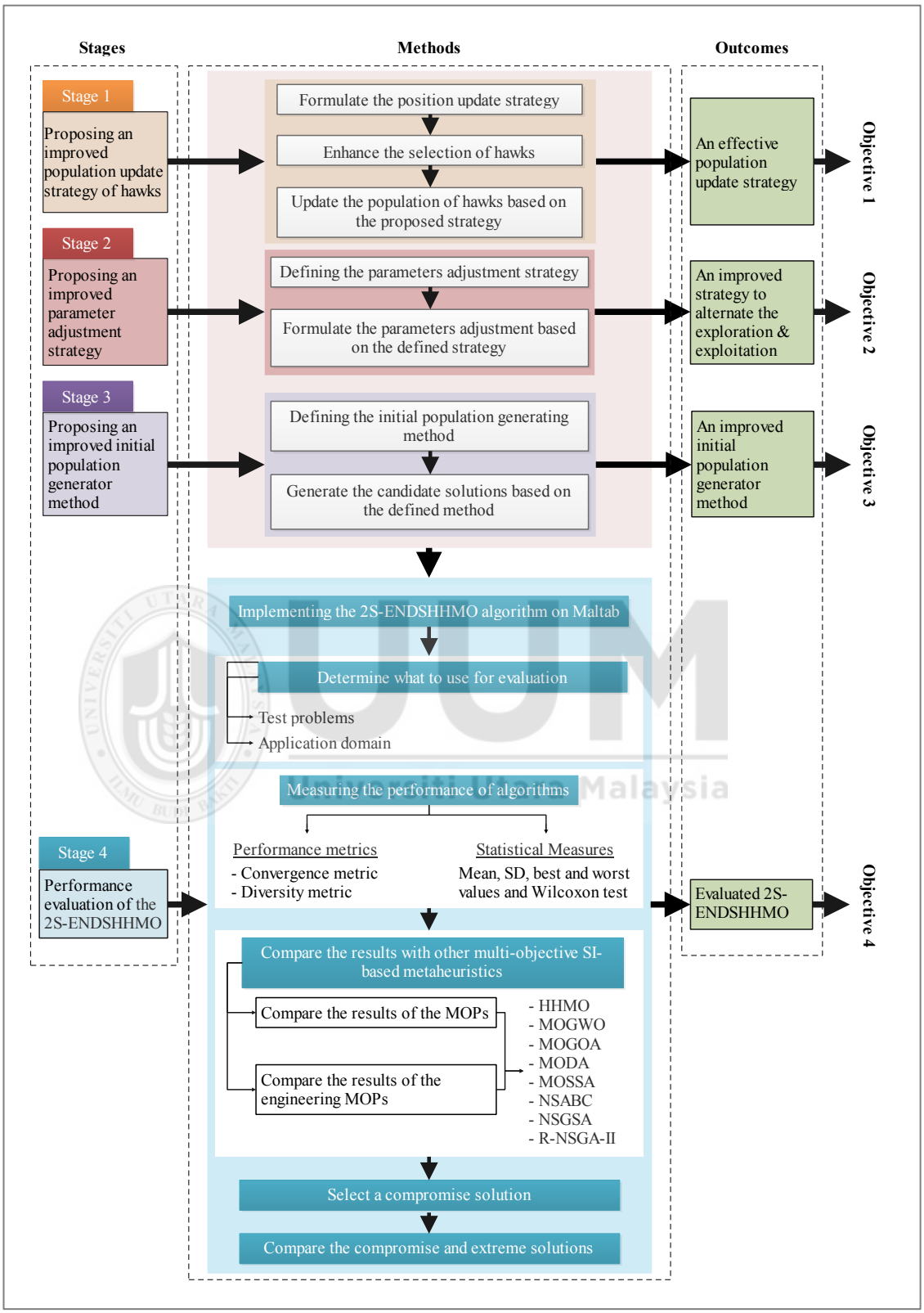


Figure 3.2. Research Framework

In the research framework, each stage involves methods, which are performed to achieve the research objectives. Stages 1, 2, 3 and 4 are designed to achieve objectives 1, 2, 3 and 4, respectively. The following sub-sections will elaborate further on each stage.

3.3 Proposing an Improved Population Update Strategy of Hawks

In this stage, the population update strategy is proposed to enhance sharing information between hawks in the population. This strategy involves updating the position of hawks with respect to the experience of all hawks and selecting the best hawks to produce the next generation. This can help in maintaining population diversity and reducing the probability of the algorithm falling into local optima.

3.4 Proposing an Improved Parameter Adjustment Strategy

There are two important components of metaheuristics: exploration and exploitation. To achieve a satisfactory performance, there should be a balance between these components. In the HHMO algorithm, the transit between exploration and exploitation are controlled by adjusting a convergence parameter, a. Therefore, to enhance these components, an adaptive parameter adjustment strategy was proposed to adjust the value of the convergence parameter. This strategy was used to alternate the exploration and exploitation during the optimization progress, with the aim to achieve an ideal balance between exploration and exploitation.

3.5 Proposing an Improved Initial Population Generator Method

The performance of the HHMO algorithm depends significantly on the distribution of the initial population in the search space (Kazemzadeh Azad, 2018; Tu et al., 2019). In this stage, a two-step initial population generator method was proposed to generate the initial potential solutions (hawks), to improve the convergence of the algorithm toward the true Pareto front.

3.6 Performance Evaluation of the 2S-ENDSHHMO Algorithm

This stage involves the implementation of the 2S-ENDSHHMO algorithm, followed by how its performance is evaluated. Generally, SI-based metaheuristics have unified procedures, namely, initialization and solution update. The initialization involves problem representation and generation of potential solutions. Problem representation is an approach to construct a problem in the form of an objective function. Based on problem characteristics the candidate solutions can be binary, real number or mixed representation. Solutions update to generate a new population (Talbi, 2009).

The improved population update strategy, adjustment strategy of the convergence parameter and two-step initial population generator method, proposed in Stages 1, 2 and 3, respectively, were integrated with the unified procedures to develop the 2S-ENDSHHMO algorithm.

The performance of the 2S-ENDSHHMO was evaluated using a set of MOPs and through applying the 2S-ENDSHHMO to solve a set of well-known multi-objective

engineering optimization problems. The evaluation stage involved measuring the performance of the 2S-ENDSHHMO using performance matrices and statistical measures in the first stage. In the second stage, the obtained results were compared with state-of-the-art multi-objective SI-based algorithms. Figure 3.3 shows the stage of performance evaluation.

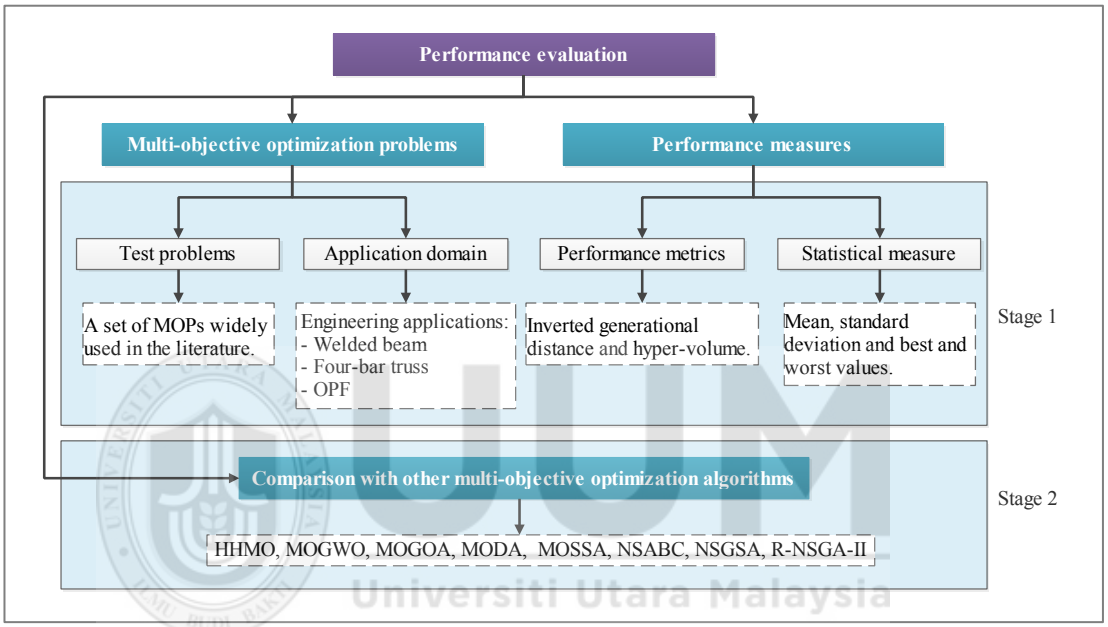


Figure 3.3. Performance evaluation stage

3.7 Multi-Objective Optimization Problem

To evaluate the performance of the 2S-ENDSHHMO algorithm, the test functions namely, the ZDT (Zitzler et al., 2000) and DTLZ (Deb, Thiele, Laumanns, & Zitzler, 2002b) have been used.

3.7.1 ZDT Test Problems

The ZDT MOPs (ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6). Each of the test functions is constituted by a minimization problem with two objectives. The ZDT family characteristics and formulations can be found in Appendix A.1.

The ZDT1 test problem has an optimal convex Pareto front and uniform distribution of solutions along the curve, as shown in Figure 3.4 (a). All variables are defined in the range [0,1]. The optimal region of Pareto corresponds to $x_1 \in (0, 1)$ and $x_i = 0$ (i = 2; …; $d = 30$), where d is the number of dimensions.

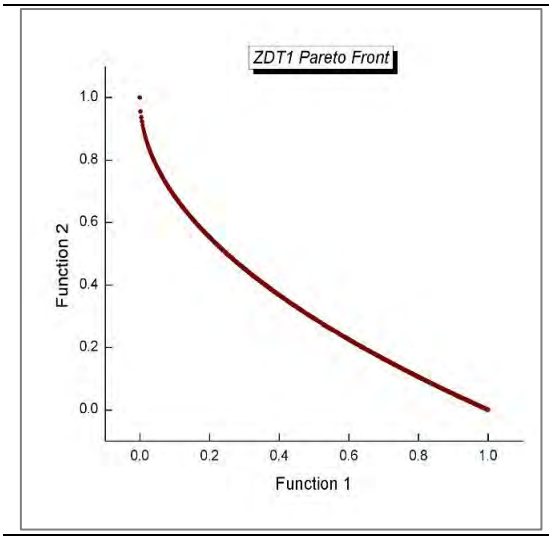
The ZDT2 problem has an optimal nonconvex Pareto front, as shown in Figure 3.4 (b). All variables are set in the range [0,1]. The optimal region of Pareto corresponds to $x_1 \in (0, 1)$ and $x_i = 0$ ($i = 2,\ldots, m = 30$).

The ZDT3 problem has a disconnected optimal Pareto boundary, caused by the presence of the sine function in $h,$, as shown in Figure 3.4 (c). The optimal region of Pareto corresponds to $x_1 \in (0, 1)$ and $x_i = 0$ ($i = 2,\ldots, m = 30$).

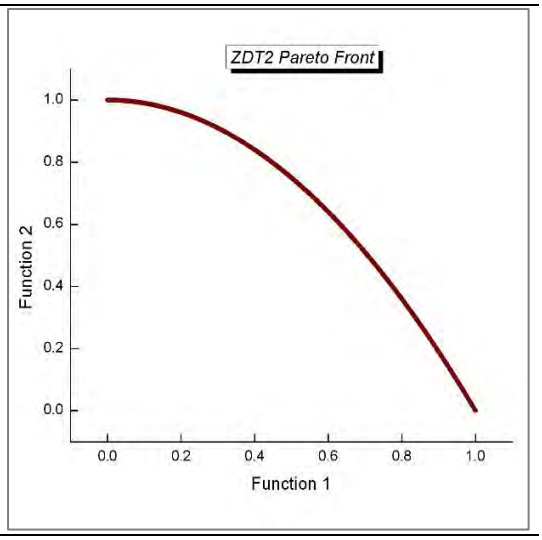
The ZDT4 function is composed of 21^9 Pareto fronts' local optima, which represents a major difficulty in the search for the global optimal. Therefore, it tests the ability of an algorithm to deal with multi-frontality. The optimal region of Pareto corresponds to $x_1 \in (0, 1)$ and $x_i = 0$ ($i = 2,\ldots, m = 10$). Figure 3.4 (d) shows the ZDT4 problem.

The ZDT6 function has two difficulties caused by the non-uniformity of the search space: Pareto optimal solutions are not evenly distributed across the Pareto global boundary and the densities of the solutions are smaller near Pareto's optimal border and larger in the region away from it (Coello et al., 2007). The optimal region of Pareto

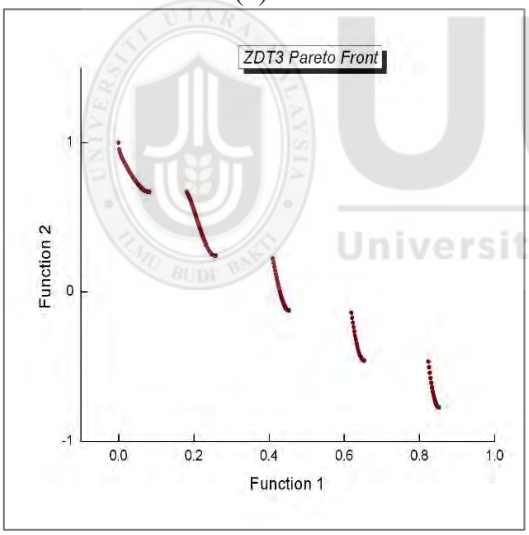
corresponds to $x_1 \in (0, 1)$ and $x_i = 0$ ($i = 2,\ldots, m = 10$). Figures 3.4 show the Pareto front of the ZDT problems.



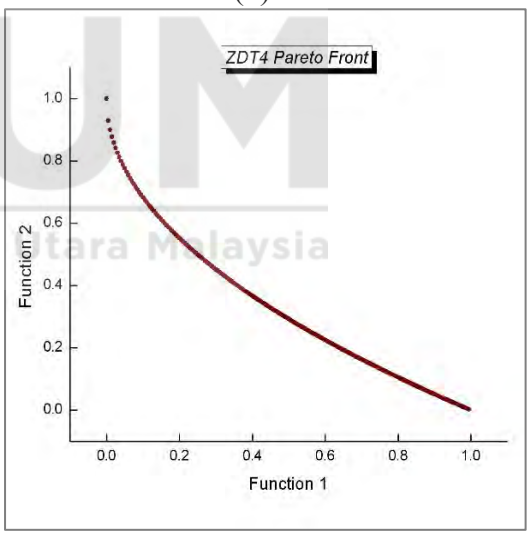
(a)



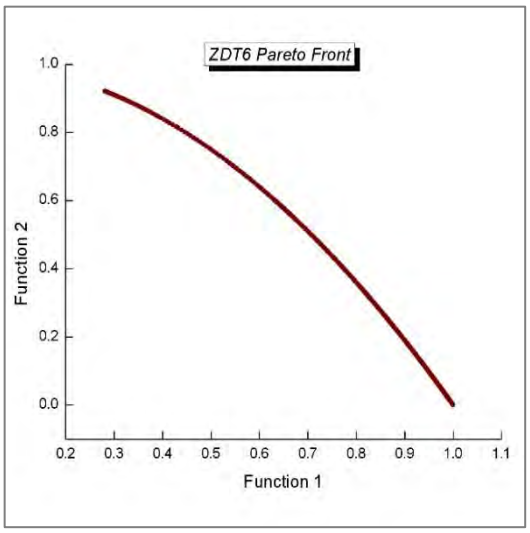
(b)



(c)



(d)



(e)

Figure 3.4.Two views of the true Pareto front of ZDT family (Adapted from Coello et al. (2007))

3.7.2 DTLZ Test Problems

The DTLZ family includes DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6 and DTLZ7. These problems test the ability of the multi-objective algorithm to converge to the true Pareto front (Coello et al., 2007). For all DTLZ problems, the total number of variables is $n = M + k - 1$. The formulations and characteristics of the DTLZ family can be found in Appendix A.2.

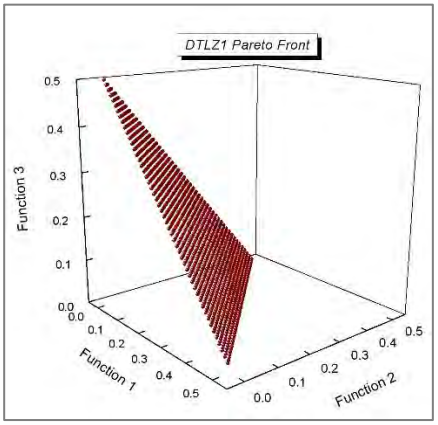
The DTLZ1 problem has a linear multimodal Pareto front. This problem can be used to test the ability of algorithms to convergence toward the true Pareto front (Coello et al., 2007). The DTLZ2 problem is a problem with a concave (spherical) and continuous Pareto-optimal front. The Pareto front of this problem represents only the positive octant of the hyper sphere, that is, the octant where the points forming the Pareto front have only positive values in their coordinates. This problem can be used to test the ability of algorithms to maintain their performance for a high number of objectives.

The optimal solutions of this problem correspond to $x_M^* = 0.5$ and its optimal Pareto boundary is defined with the objective values corresponding to $\sum_{i=1}^{M}(f_i^*)^2 = 1$.

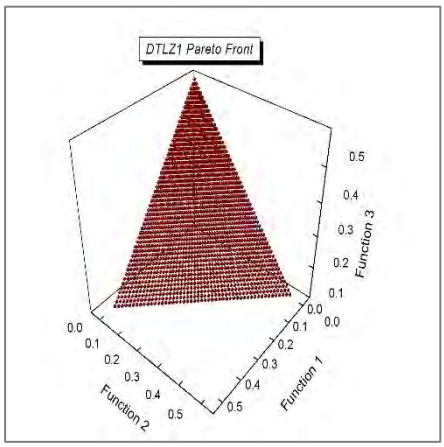
The DTLZ3 has a concave, scalable, multimodal true Pareto front. The DTLZ4 problem has a concave (spherical) Pareto front. This problem was created to evaluate the ability of algorithms to generate a well distributed set of solutions on the Pareto front. This problem has a natural tendency to attract solutions to a specific region of the Pareto front.

The DTLZ5 problem tests the ability of an algorithm to converge to a True Pareto front. The DTLZ6 problem is similar to the DTLZ5 problem in terms of the number of decision variables. However, the DTLZ6 problem incorporates greater convergence difficulties by using different *g* functions. This change makes an algorithm difficult to converge to the true Pareto front.

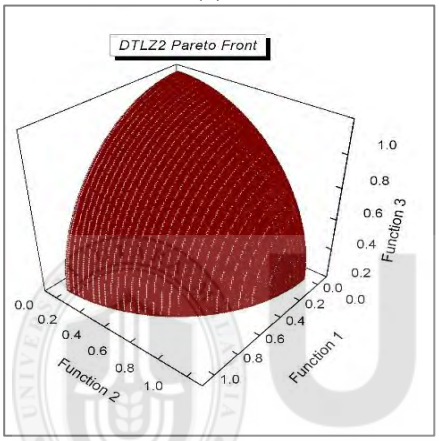
The DTLZ7 problem does not use the spherical coordinate system in *M*-dimensional space. The last objective $f_M(x)$ is the only one that depends on the other variables of the problem. This problem has $2M - 1$ disconnected regions on the Pareto Border. The DTLZ7 problem tests the ability of algorithms to keep individuals in different regions of the True Pareto front. Figures 3.5 show the Pareto front of the DTLZ problems (Coello et al., 2007).



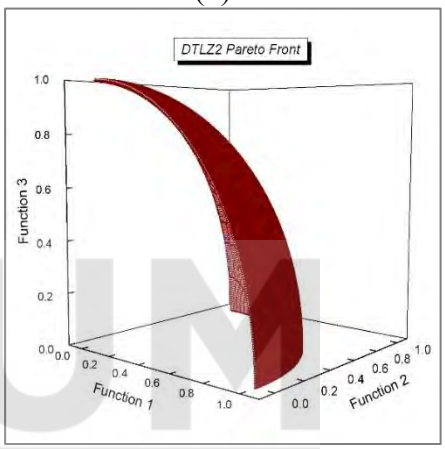
(a)



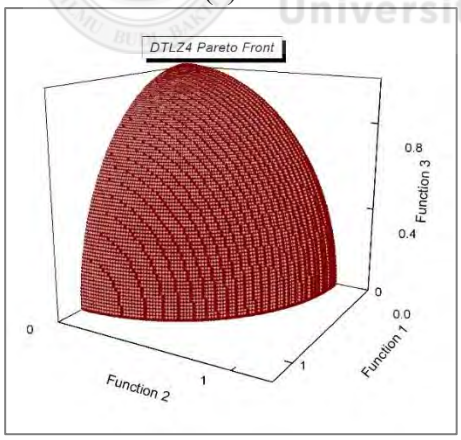
(b)



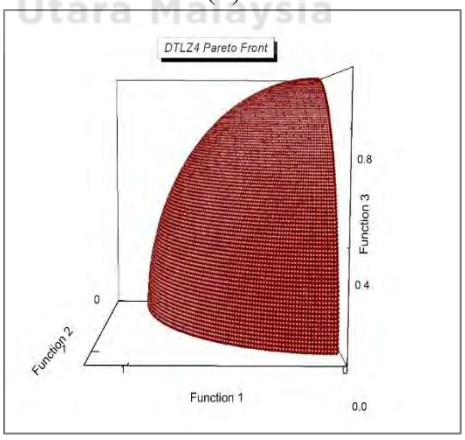
(c)



(d)



(e)



(f)

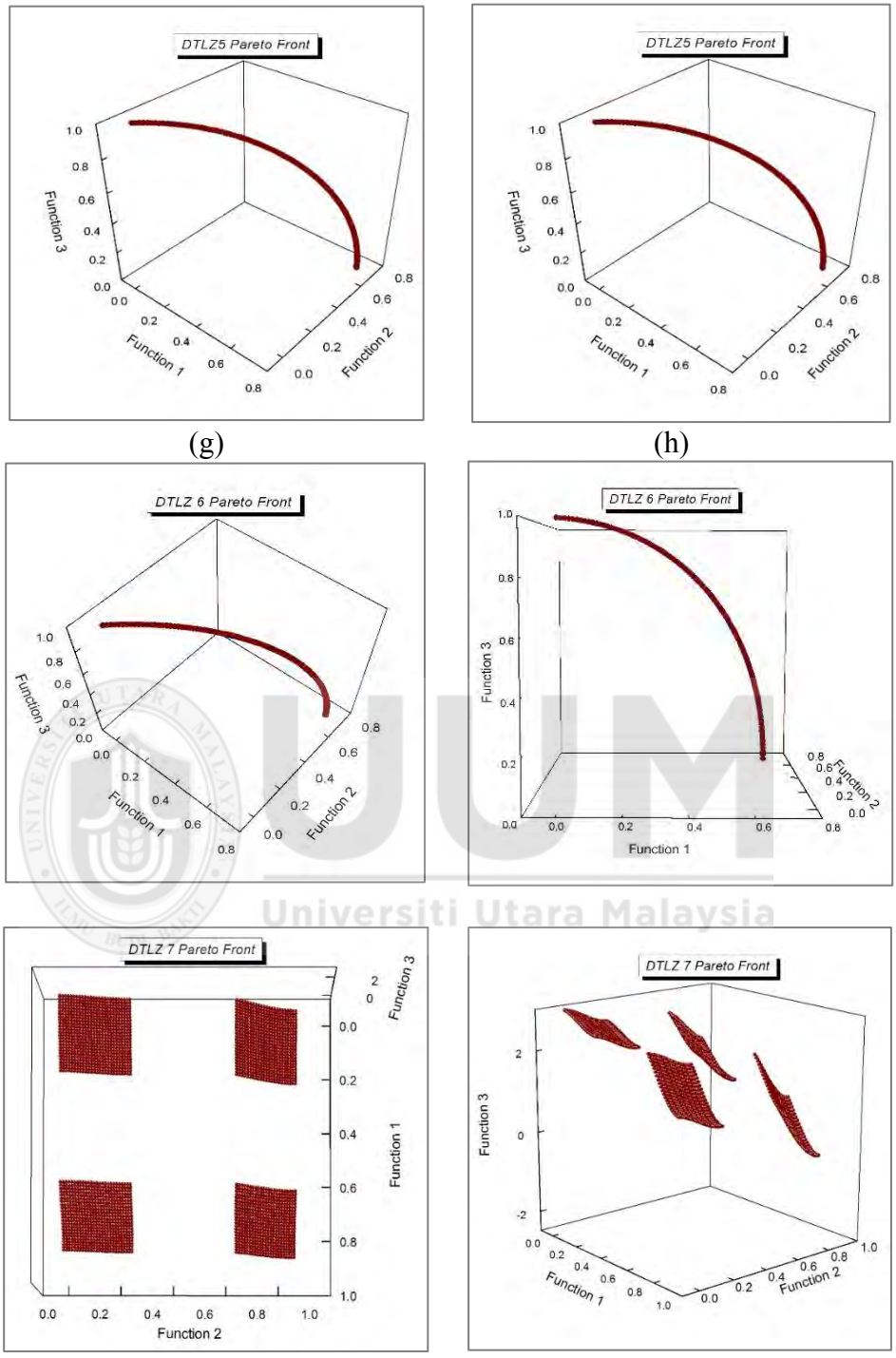


Figure 3.5.Two views of the true Pareto front of DTLZ family family (Adapted from Coello et al. (2007))

In this study, twelve test problems were used, namely, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 and DTLZ2 with three, five and 10 objectives, problems. These problems were chosen because they present the Pareto-optimal front continuum and the space of decision variables is continuous. These test problems were used in the comparison of the performance of 2S-ENDSHHMO with other algorithms, namely, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II.

In addition to ZDT and DTLZ2 MOPs, the UF family proposed by Zhang et al. (2009) has been used in evaluating the performance of 2S-ENDSHHMO. This set includes UF1-UF10 MOPs that have convex, concave, disconnected and linear Pareto front characteristics. The UF1-UF7 involves two objectives and UF8-UF10 has three objectives problems. These sets are used in the second phase of comparison where the performance of 2S-ENDSHHMO is compared with the results of each algorithm collected from the original publications.

3.8 Application Domain

The evaluation of the 2S-ENDSHHMO algorithm was extended by applying it in solving three well-known multi-objective engineering problems, namely, welded beam (Ragsdell & Phillips, 1976) , four-bar truss (Coello & Pulido, 2005) and OPF MOPs.

3.8.1 Significant of Engineering Applications

Structural engineering is a classic branch of civil engineering that responsible for economically designing buildings, bridges, warehouses, walls and other structures with sufficient strength to prevent collapse when they are loaded and protect them from extreme natural phenomena such as wind, snow, fire and earthquakes. They can also deal with the design and calculation of machinery, medical equipment, and vehicles where structural integrity affects functioning and safety. Any structure is essentially made up of only a small number of different types of elements, which include beams, trusses, columns, plates, arches, shells and catenaries.

Welded beams are those whose sections are composed of several steel sheets welded together to achieve the desired geometry. The economic benefit of the use of welded beams for the construction of structures and buildings allows construction companies to reduce the cost of labour, while ensuring the unique reliability of buildings. In structural engineering, trusses are structures formed by rigid elements composed of members or bars, whose ends are connected at joints or points known as nodes. In the trusses the loads (forces) are applied only to the nodes resulting in either a compression or tension force. They are commonly used in the design of bridges, overpasses and roofs. Trusses are an efficient way to cover long distances while minimizing the amount of material used, especially in the design of overpasses, as the force is distributed among a number of members.

Electrical engineering is a branch of engineering that is responsible for studying and applying everything related to electricity. Power Engineering is a core sub-field of electrical engineering that plans, analyses and develops systems for generation, transportation, transmission, distribution and use of energy. In power engineering, the power system is a network of electrical components. The main components of a power system are, namely, the generators, transformers, transmission and distribution (power lines). The analysis of the power flow consists of determining the flows of active and reactive powers. This helps in the best operating condition of an electrical system, minimizing possible losses in transmission, as well as determining the energy planning for expansion, thus reducing economic costs.

3.8.2 Engineering multi-objective optimization problem

The engineering MOPs are usually used in the literature to show the effectiveness of algorithms. These problems are the most challenging, due to their diverse characteristics, most of which are constrained. In general, in constrained optimization problems, the number of linear and nonlinear constraints, objective function and the decision variables are major factors in determining the difficulty of the problem (Tuba, Bacanin, & Stanarevic, 2012). *Table 3.1* summarises the characteristics of the welded beam, four-bar truss and OPF engineering MOPs.

Table 3.1
Characteristics of engineering MOPs

MOP	Number of constraints		Type of constraint
	Linear	Nonlinear	
Welded beam	2	4	- Four inequality constraints - Range of decision variables
Four-bar truss	4	-	Range of decision variables
OPF	6	2	Six inequality constraints Two equality constraints

In Table 3.1 *M* and *D* denote the number of objectives and the number of decision variables. The constrained optimization problem is a kind of mathematical programming problem that is often encountered in the field of engineering application. The general form of MOP can be written as shown in Equation (3.1).

$$\text{Minimize } F(\vec{X}) = \left(f_1(\vec{X}), f_2(\vec{X}), f_3(\vec{X}), \dots, f_M(\vec{X})\right);$$

$$\vec{X} = (x_1, x_2, x_3, \dots x_n) \in F \subseteq S \subseteq R^n$$

subject to: (3.1)

$$g_i(\vec{X}) \leq 0 \qquad i = 1,2,3, \dots, q$$

$$h_j(\vec{X}) = 0 \quad j = q+1, \dots, m$$

where $\vec{X}$ is the vector of solutions, *F* is the feasible region in the search space, *S*. There are q inequality and m-q equality constraints. $f(\vec{X})$ is the objective function. $\vec{X}$ that satisfies all the constraints is a feasible solution to the problem. All of the feasible solutions constitute the feasible region. Inequality constraints that satisfy $g_i(\vec{X}) = 0$ are called active at X. Using these definitions, the nonlinear programming problem is

to find a point $\check{X} \in$F such that $f(\check{X}) \leq f(\vec{X})$ for all $\vec{X} \in F$ (Bazaraa, Sherali, & Shetty, 2013).

The constrained optimization problem requires constraints handling method. The penalty function is a common method to deal with constrained optimization problems (Bassen, Vilkhovoy, Minot, Butcher, & Varner, 2017; Berrouk et al., 2018; bin Mohd Zain, Kanesan, Chuah, Dhanapal, & Kendall, 2018; Coello Coello, 2000; Ding et al., 2017; Farnad & Jafarian, 2018; Kaveh & Mahdavi, 2019; Kohli & Arora, 2018; Mellal & Zio, 2019; Meng, Shen, & Jiang, 2014; Parsopoulos & Vrahatis, 2005; Savsani, 2014; Tawhid & Savsani, 2018a; Tomassetti, 2010). The goal is to allow individuals in a group to violate the constraints to a certain extent, but the individual must be punished according to the degree of violation of the constraints to reduce its selection. The degree to which a probability individual violates a constraint is determined by a penalty function, and most methods construct a penalty function using the approach in Equation (3.2).

$$eval(\vec{X}) = \begin{cases} f(\vec{X}) & ; \text{ if } \vec{X} \in F \\ f(\vec{X}) + p(\vec{X}) & ; \text{ otherwise} \end{cases} \quad (3.2)$$

where p is a penalty term. If no violation occurs, p will be zero and positive otherwise. Under this conversion, the overall objective function now is $eval(\vec{X})$, which serves as an evaluation function in the algorithm. Figure 3.6 shows the main stages in solving a constraints MOP. Figure 3.6 shows the procedures of design MOPs.

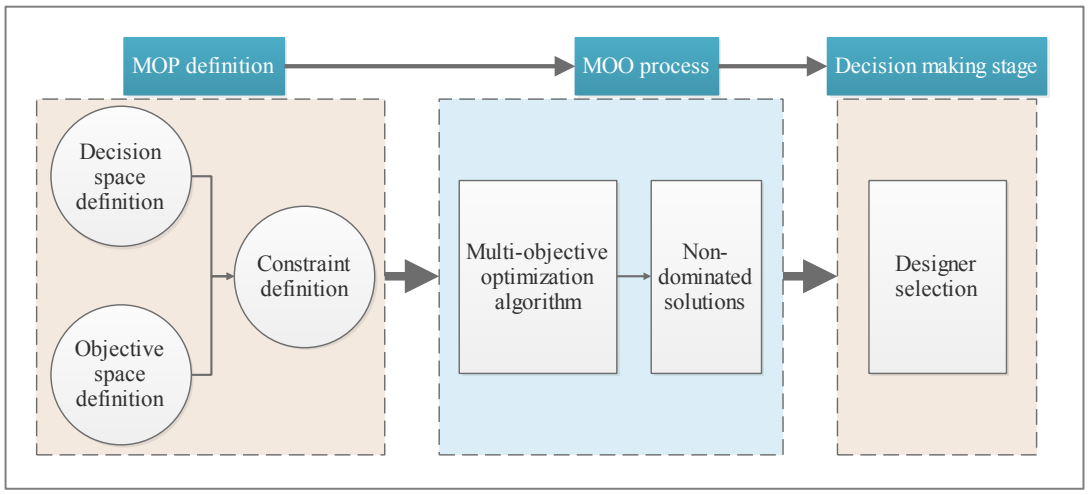


Figure 3.6. Multi-objective optimization procedures for a design MOP

As shown in Figure 3.6 solving a design MOP starts by defining the problem model. This includes defining the variables in the decision space, objective functions and the constraints of the MOP. The goal of optimization is to find the best combination of design variables that minimize (maximize) two or more objective functions, which is subject to constraints on its decision variables.

3.8.3 Welded Beam Design Problem

This optimization problem concerns the design and fabrication of welded joints. A beam is welded onto another beam and carries a certain load. The welded beam problem has been proposed as a SOP (Ragsdell & Phillips, 1976). For a more flexible design the welded beam problem has been transformed into a MOP (Deb & Sundar, 2006). In the SOP the goal is to minimize the fabrication cost ($f1(x)$) of the joint. In a MOP, one more objective has been included, which is to minimize the end deflection ($f2(x)$) of the welded beam. Figure 3.7 shows a schematic of the welded beam MOP.

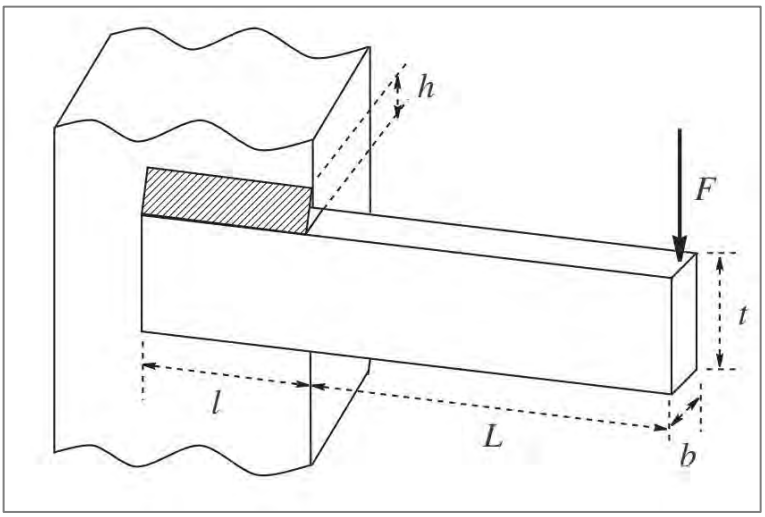


Figure 3.7. Schematic of welded beam MOP family (Adapted from Zheng and Zhou (2013))

The welded beam MOP has four continuous design variables, namely, weld thickness (h), weld length (l), beam height (t), and beam width (b) whose nonlinear constraints include normal and shear stress and a geometry constraint. The design variables can be formalized as: $x_1(h)$, $x_2(l)$, $x_3(t)$, and $x_4(b)$, while the objective function and constraints are as shown in Equations (3.3-3.6).

$$\min f_1(\vec{x}) = 1.10471{x_1}^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{3.3}$$

$$\min f_2(\vec{x}) = \frac{2.1952}{x_3 x_4} \tag{3.4}$$

Subject to:

$$\begin{aligned} g_1(\vec{x}) &= T(\vec{x}) - T_{max} \leq 0 \\ g_2(\vec{x}) &= \sigma(\vec{x}) - \sigma_{max} \leq 0 \\ g_3(\vec{x}) &= F - P_c(\vec{x}) \leq 0 \\ g_4(\vec{x}) &= x_1 - x_4 \leq 0 \\ 0.125 \leq x_1, x_4 &\leq 5 \end{aligned} \tag{3.5}$$

$$0.1 \leq x_2, x_3 \leq 10$$

$$T = (tau1^2) + (tau2^2) + (ltau1tau2)/\sqrt{0.25(l^2 + (h+t)^2)}$$

$$tau1 = \frac{6000}{\sqrt{2}hl}$$

$$tau2 = \frac{6000(14+0.5l)\sqrt{0.25(l^2+(x_1+t)^2)}}{2(0.707hl(\frac{l^2}{12+0.25(x_1+t)^2}))} \quad (3.6)$$

$$\sigma(\vec{x}) = \frac{504000}{t^2 b}$$

$$P_c(\vec{x}) = 64746.022(1 - 0.0282346t)tb^3$$

where F = 6000 lb, T_{max} = 13600 psi, $E = 3 \times 10^6$ psi, σ_{max} = 30000 psi, u_{max} = 0.25 in and L = 14 in. In Equation (3.3), the coefficients 1.10471 and 0.04811 are related to the material cost per unit volume. The first two constraints ensure that the shear stress, T and the normal stress, σ, developed along the beam support, are less than the allowable shear (T_{max}) and normal (σ_{max}) stresses of the material. The third constraint ensures that the tensile strength (along the t-direction) of the beam is greater than the applied load F. The fourth constraint ensures that the thickness of the beam is not less than the thickness of the weld.

3.8.4 Four-bar Truss Design Problem

The four-bar truss design problem was studied by (Stadler & Dauer, 1993). The goal is to design the truss with minimum volume and deflection of node *C*. Figure 3.8 shows the schematics of the four-bar truss structure.

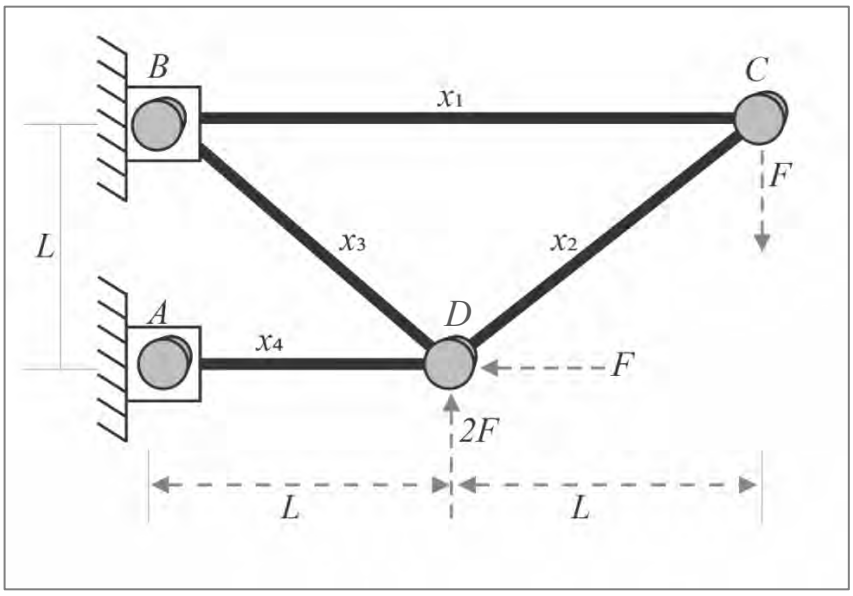


Figure 3.8. Schematic of the four-bar truss MOP family (Adapted from Stadler and Dauer (1993))

In Figure 3.8, x_1, x_2, x_3 and x_4 are the cross-sectional areas of each member in cm^2 and the objective function and constraints are as formulated in Equations (3.7 - 3.9).

$$\min f_1(x) = L \times 2x_1 + \sqrt{2x_2} + \sqrt{x_3} + x_4 \tag{3.7}$$

$$\min f_2(x) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} + \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right) \tag{3.8}$$

$$\begin{aligned} \frac{F}{\sigma} \le x_1 \le 3 \times \frac{F}{\sigma} \\ \frac{\sqrt{2 \times F}}{\sigma} \le x_2 \le 3 \times \frac{F}{\sigma} \\ \frac{\sqrt{2 \times F}}{\sigma} \le x_3 \le 3 \times \frac{F}{\sigma} \\ \frac{F}{\sigma} \le x_4 \le 3 \times \frac{F}{\sigma} \end{aligned} \tag{3.9}$$

where F=10 KN, E=2×10^6 KN/cm^2 is the Young's modulus, L=200 cm, σ=10 KN/cm^3

3.8.5 Optimal Power Flow

The OPF refers to the optimization process of objective function by adjusting the parameters of various control devices in the electrical power system, with respect to the constraints of network and equipment (Bouchekara, 2014; Chen, Yi, Zhang, & Lei, 2018; Ebeed, Kamel, & Jurado, 2018; Kahourzade, Mahmoudi, & Mokhlis, 2015). The OPF is a highly nonlinear, non-convex and complex problem with several constraints and decision variables.

In general, flow optimization is divided into two types: active and reactive power optimization. The objective function of active power optimization is the generating cost or power consumption, and the objective function of reactive power optimization is the transmission active power losses of the entire network. In this study, the objective functions of optimal power flow are formulated as shown in Equation (3.9).

$$\begin{aligned} \min \quad & F(f_{cost}(X), f_{loss}(X)) \\ s.t. \quad & g(x,u) \le 0 \\ & h(x,u) = 0 \end{aligned} \tag{3.10}$$

where f_{cost}(x,u) and $f_{\text{loss}}(x,u)$ are the objective functions to be optimized, which includes minimizing the operating cost of generators (minimize generation fuel cost) and transmission active power loss. The mathematical model is formulated as in Equation (3.11).

$$f_{cost}(X) = \sum_{i=1}^{NG} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2 \right) \tag{3.11}$$

where $f_{\text{cost}}(X)$ is the total fuel cost function ($/h). a_i, b_i, c_i are the cost coefficients of generator i and NG is the number of generators. $X = [P_G, V_G, T_{tap}, Q_C]$ is the control

variables. $P_G = [P_{G1}, P_{G2}, \ldots P_{G(NG-1)}]$ is the active power output of the generators. $V_G = [V_{G1}, V_{G2}, \ldots V_{G(NG)}]$ and $T_{tap} = [T_{trans1}, T_{trans1}, \ldots, T_{trans1(NT)}]$ are the generator voltage and the tap-settings of transformers, respectively. whereas NT is the number of tap transformers. $Qc = [Q_{C1}, Q_{C2}, \ldots, Q_{C(NC)}]$ is the reactive power compensation power. NC is the number of compensator units.

Minimizing the transmission active power loss of the system is to determine the active power output of each unit under various constraints. This can be achieved by adjusting the generator output and configuring reactive power compensation equipment. In this case, the voltage level can be effectively improved, and the system loss reduced. Therefore, the active power loss of the whole system can be expressed as shown in Equation (3.12).

$$P_{loss} = \sum_{i=1}^{NB} P_{Gi} - \sum_{i}^{NB} P_{Di} \tag{3.12}$$

where P_D is real load demand and NB denotes the total number of busses.

The control variable u, in Equation (3.10), can be expressed as in Equation (3.13)

$$u = [P_{G2}, P_{G3}, \ldots P_{Gm}, U_{G1}, U_{G2}, \ldots, U_{Gm}, T_1, T_2, \ldots T_{NT}, Q_{C1}, Q_{C2}, \ldots, Q_{C(NC)}] \tag{3.13}$$

where P_G is the active output of other generators except the balancer node. U_G is the terminal voltage at the bus connected to the generator. T is the adjustable transformer ratio. Q_C is reactive power compensation.

For the system, to operate in a safe and stable environment, the parameters of the system must also meet certain constraints, including inequality, g and equality, h

constraints. The equality constraint is the power flow equation of the power system, as formulated in Equations (3.14 and 3.15), respectively.

Real power constraints

$$P_{Gi} - P_{Di} - V_j \sum_{j=i}^{NB} [G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij})] = 0 \tag{3.14}$$

Reactive power constraints

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=i}^{NB} V_j [G_{ij} sin(\theta_{ij}) - B_{ij} \cos(\theta_{ij})] = 0 \tag{3.15}$$

where G_{ij} and B_{ij} are the conductance and susceptance, respectively, between bus *i* and bus *j*. $\theta_{ij} = \theta_i - \theta_j$ is the phase angle between θ_i and θ_j. The inequality constraints refer to controlling the adjustable variables to change within a certain allowable range to meet the safety of power system operation. The main inequality constraints include constraints on power and voltage, as described in Equations (3.16-3.21).

Generator voltage limits

$$V_{Gi}^{\min} \le V_{Gi} \le V_{Gi}^{\max}, \quad i = 1, \cdots, NG \tag{3.16}$$

Active power generated at slack bus

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max}, \quad i = 1, \cdots, NG \tag{3.17}$$

Generated reactive power

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}, \quad i = 1, \cdots, NG \tag{3.18}$$

Transformer constraints

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 1, \cdots, NT \tag{3.19}$$

Shunt VAR compensator constraints

$$Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max}, \quad i = 1, \cdots, NC \tag{3.20}$$

Load bus voltage

$$V_{Li}^{\min} \le V_{Li} \le V_{Li}^{\max}, \quad i = 1, \cdots, NL \tag{3.21}$$

To verify, the effectiveness of the proposed 2S-ENDSHHMO algorithm is used to solve the multi-objective OPF problem, and the obtained results are compared with other algorithms. The OPF calculation is performed for the IEEE 30-node system.

IEEE 30-bus system

The IEEE30 system consists of 30 buses, 41 transmission lines and four automatically controlled tap transformers, namely, 6–9, 6–10, 4–12 and 27–28. The 30 busses consist of six generator buses and 24 load buses. The generator buses, consist of thermal generators, which are connected at bus numbers 1, 2, 5, 8, 11 and 13. The 24 load buses have 283.4 MW demand power (Chakraborty & Chakrabarti, 2015; Hoque, Hoque, & Sinha, 2019). The one-line diagram of the IEEE 30-bus system is shown in Figure 3.9.

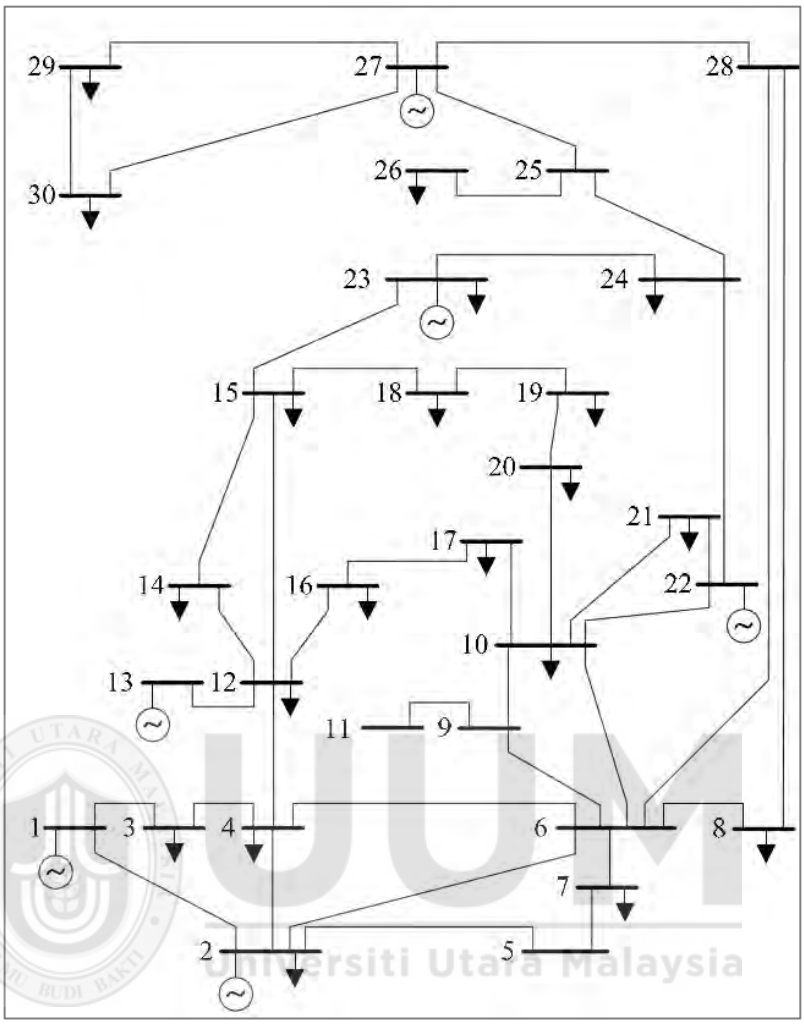


Figure 3.9. IEEE 30-bus system (Adapted from Christie (1993))

Bus data:										
Bus no	Bus code	V In p.u	$<\delta$ radian	P_a MW	Q_d MVar	P_g MW	Q_g MVar	Q_{min} MVar	Q_{max} MVar	Q MVar
1	1	1.06	0	0	0	0	0	0	0	0
2	2	1.043	0	21.7	12.7	40	0	–40	50	0
3	0	1	0	2.4	1.2	0	0 0	0	0	0
4	0	1.06	0	7.6	1.6	0	0	0	0	0
5	2	1.01	0	94.2	19	0	0	–40	40	0
6	0	1	0	0	0	0	0	0	0	0
7	0	1	0	22.8	10.9	0	0	0	0	0
8	2	1.01	0	30	30	0	0	–10	60	0
9	0	1	0	0	0	0	0	0	0	0
10	0	1	0	5.8	2	0	0	–6	24	19
11	2	1.082	0	0	0	0	0	0	0	0
12	0	1	0	11.2	7.5	0	0	0	0	0
13	2	1.071	0	0	0	0	0	–6	24	0

14	0	1	0	6.2	1.6	0	0	0	0	0
15	0	1	0	8.2	2.5	0	0	0	0	0
16	0	1	0	3.5	1.8	0	0	0	0	0
17	0	1	0	9	5.8	0	0	0	0	0
18	0	1	0	3.2	0.9	0	0	0	0	0
19	0	1	0	9.5	3.4	0	0	0	0	0
20	0	1	0	2.2	0.7	0	0	0	0	0
21	0	1	0	17.5	11.2	0	0	0	0	0
22	0	1	0	0	0	0	0	0	0	0
23	0	1	0	3.2	1.6	0	0	0	0	0
24	0	1	0	8.7	6.7	0	0	0	0	4.3
25	0	1	0	0	0	0	0	0	0	0
26	0	1	0	3.5	2.3	0	0	0	0	0
27	0	1	0	0	0	0	0	0	0	0
28	0	1	0	0	0	0	0	0	0	0
29	0	1	0	2.4	0.9	0	0	0	0	0
30	0	1	0	10.6	1.9	0	0	0	0	0

Bus code: 1=Slack bus, 2= Generator bus (PV) and 0= Load bus (PQ)
Base MVA = 100.

Line data:

Bus no	Bus no	Resistance (R.-in p.u)	Reactance (X-in p.u)	Line charging admittance (1/2 *B*)	Transformer tapping
1	2	0.0192	0.0575	0.0264	1
1	3	0.0452	0.1852	0.0204	1
2	4	0.057	0.1737	0.0184	1
3	4	0.0132	0.0379	0.0042	1
2	5	0.0472	0.1983	0.0209	1
2	6	0.0581	0.1763	0.0187	1
4	6	0.0119	0.0414	0.0045	1
5	7	0.046	0.116	0.0102	1
6	7	0.0267	0.082	0.0085	1
6	8	0.012	0.042	0.0045	1
6	9	0	0.208	0	0.987
6	10	0	0.556	0	0.969
9	11	0	0.208	0	1
9	10	0	0.11	0	1
4	12	0	0.256	0	0.932
12	13	0	0.14	0	1
12	14	0.1231	0.2559	0	1
12	15	0.0662	0.1304	0	1
12	16	0.0945	0.1987	0	1
14	15	0.221	0.1997	0	1
16	17	0.0824	0.1923	0	1

Line data:					
Bus no	Bus no	Resistance (R.-in p.u)	Reactance (X-in p.u)	Line charging admittance (1/2 *B*)	Transformer tapping
15	18	0.1073	0.2185	0	1
18	19	0.0639	0.1292	0	1
19	20	0.034	0.068	0	1
10	20	0.0936	0.209	0	1
10	17	0.0324	0.0845	0	1
10	21	0.0348	0.0749	0	1
10	22	0.0727	0.1499	0	1
21	22	0.0116	0.0236	0	1
15	23	0.1	0.202	0	1
22	24	0.115	0.179	0	1
23	24	0.132	0.27	0	1
24	25	0.1885	0.3292	0	1
25	26	0.2544	0.38	0	1
25	27	0.1093	0.2087	0	1
28	27	0	0.396	0	0.968
26	29	0.2198	0.4153	0	1
27	30	0.3202	0.6027	0	1
29	30	0.2399	0.4533	0	1
8	28	0.0636	0.2	0.0214	1
6	28	0.0169	0.0599	0.065	1

3.9 Performance Measure

In this section, the performance measures are presented to evaluate the performance of the 2S-ENDSHHMO algorithm. Measuring the performance of MOO is more complex than in the case of single objective optimization. In general, the main goal of all multi-objective metaheuristics involves minimizing the distance from the set of dominant solutions found to the true Pareto front and obtaining good diversity of the generated solutions (H. Chen et al., 2018; Dai & Lei, 2019; Li et al., 2018a; Monsef, Naghashzadegan, Jamali, & Farmani, 2019; Wang, Liu, Ma, Wong, & Li, 2019; Zou et al., 2019).

It should be noted that convergence and diversity can be conflicting goals, so using only one of the metrics does not fully evaluate the performance of an algorithm (Deb, 2001). Therefore, in this study, different performance metrics were utilized to measure the convergence and diversity of the obtained solutions. To statically measure the performance of multi-objective metaheuristics, statistical measures are normally used. The performance metrics and statistical measures will be discussed in detail in the next sections.

3.9.1 Performance Metrics

In this work, R-metrics proposed by Li et al. (2018b) were used to assess the performance of a 2S-ENDSHHMO algorithm. The general idea of this metric is to pre-process the preferred solutions according to a multi-criterion decision making approach before using a regular performance metric to evaluate the obtained solutions (Li et al., 2018b). In this study, the regular metrics that were utilized are the IGD and HV. Figure 3.10 shows the high level flowchart of the R-Metric calculation.

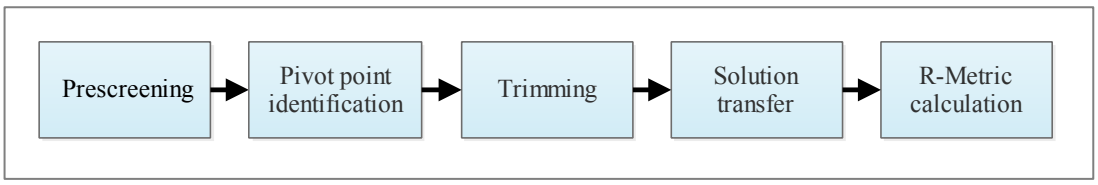


Figure 3.10. Flowchart of R-metric calculation (Adapted from Li et al. (2018b))

In Figure 3.10, five steps are performed for the calculation of the R-metric. The first step (Pre-screening) is to remove all dominated and duplicated solutions and keep only the non-dominated. The second step (Pivot point Identification) is responsible for identifying a representative point that reflects the overall satisfaction of the solutions in relation to the reference point provided by the DM, for which the closest solution is

used in relation to the reference point. In the third step (Trimming), the points outside the region of interest are eliminated to have a fair comparison between different sets. In other words, only solutions located in the approximate region of interest are valid for performance evaluation. The fourth stage (Solution transfer) is the core of the R-metric, where the cut points are transferred to a virtual position. Finally, the last step (Calculate R-metric) is to apply the regular a performance metric (the IGD and HV in this study) to the solutions processed by the R-metric.

The HV (Zitzler & Thiele, 1999) is used when the optimal Pareto solutions are unknown, where the larger value of HV indicates a better result. Mathematically, the HV is described by equation (3.22).

$$\mathrm{HV(A)} = \lambda(\cup_{a \in A} [f_1(a), r_1] \times [f_2(a), r_2] \times \ldots \times [f_k(a), r_k]) \qquad (3.22)$$

The HV is denoted as the hyper-volume of a space that is dominated by a set of solution A and is bounded by a reference point, $r = (r_1, r_2, r_3, \ldots r_k) \in R^k$. $\lambda(S)$ is the Lebesgue measure of a set S. $[f_1(a), r_1] \times [f_2(a), r_2] \times \ldots \times [f_k(a), r_k])$ includes all points that are dominated by the point a but not dominated by the reference point (Brockhoff, Friedrich, & Neumann, 2008). Figure 3.11 illustrates the HV metric.

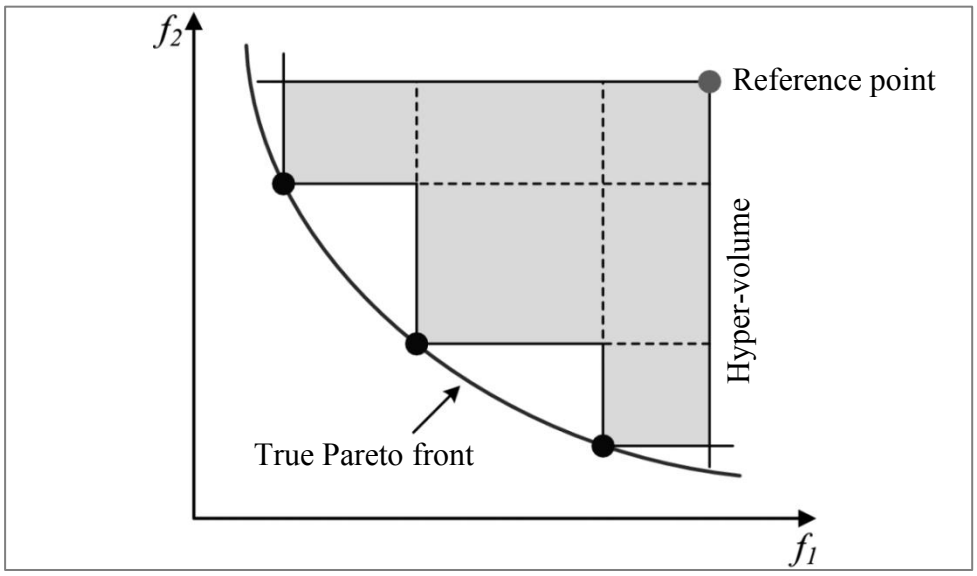


Figure 3.11. Hyper-volume indicator for a set of non-dominated solutions with respect to the reference point (Adapted from Ayari, Nikdast, Hafnaoui, Beltrame, and Nicolescu (2017))

The inverted generational distance (Coello & Cortés, 2005) measures the average distance between the solutions of the set Pareto front (Pareto Front approximation) obtained by the algorithm and the point closest to each one in the True Pareto front. The IGD allows to observe if the Pareto border converges to the Pareto optimal set. Equation (3.23) defines the IGD metric.

$$IGD = \frac{\sqrt[p]{\sum_{i=1}^{n} d_i^2}}{n} \tag{3.23}$$

where n is the number of solutions belonging to True Pareto front, $p = 2$ and d_i^2 is the minimum Euclidean distance between point i and the nearest point of the Pareto front. A smaller value for this indicator means that Approximate Pareto front is a better approximation of True Pareto front, IGD=0 means that, all the generated elements are in the True Pareto front of the problem (Coello & Cortés, 2005).

The epsilon metric (Zitzler et al., 2003) is one of the most-used metrics in measuring the performance of MOO algorithm. It determines which approximation set is better than another (Riquelme et al., 2015). This epsilon metric compares two sets of non-dominated solutions *A* and *B* by calculating the lowest translation value ε of *A* in such a way that *B* is fully dominated by *A* in at least one solution, as shown in Equation (3.24).

$$\varepsilon(A, B) = \min\{\varepsilon \in \mathrm{R} \mid \forall \mathrm{b} \in \mathrm{B} \exists \mathrm{a} \in A: \mathrm{a} \succ_{\varepsilon} b\} \tag{3.24}$$

where *A* is approximated Pareto front and True Pareto front. The smaller value for this metric indicates better approximation for the Pareto front of the problem.

3.9.2 Statistical Measure

For fair comparison, the statistical measures for 10 independent runs is calculated. These measures include, the mean, standard deviation (SD), best and worst values of the performance metrics. These measures were calculated for the 2S-ENDSHHMO and the other algorithms used in the comparison, namely, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II. To make the conclusions statistically reasonable and reliable, the Wilcoxon rank sum test (Wilcoxon, 1992) is used to check that the results obtained by the 2S-ENDSHHMO algorithm and other comparison algorithms have a significant difference at the significance level, α. If null hypothesis is rejected, when *p*-value < α, this indicates that a given algorithm outperforms the other one with the associated *p*-value (Derrac, García, Molina, & Herrera, 2011).

3.9.3 Comparison with Other Algorithms

The performance of a 2S-ENDSHHMO algorithm were determined by comparing it with other multi-objective SI-based metaheuristics, namely, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, in solving the same MOPs presented in Section 3.7. These algorithms were chosen because they are state-of-the-art algorithms for dealing with MOPs. The performance metrics presented in Section 3.9.1, are utilized in this comparison, under the same common parameters, namely, population size, dimension and maximum number of iterations.

The 2S-ENDSHHMO algorithm and the other algorithms were implemented using MATLAB v.2018b, which is a technical computing platform, produced by Mathworks Inc. The platform is used for high-performance numeric computation and visualization, where it integrates numerical analysis, matrix calculation, signal processing and graphics processes in one environment.

In these experiments, the True Pareto front for each test problem is used as the reference set. For HHMO and 2S-ENDSHHMO algorithms, a set of reference points are used as an inspiration level determined by the DM. For HV calculation, the reference points are set as the nadir vectors taken from the reference set of each test problems (Li et al., 2018b).

3.10 Summary

The experimental method has been adopted in conducting this research to achieve the research objectives. The main goal of this study is to develop a 2S-ENDSHHMO algorithm with effective exploration and exploitation to improve the convergence toward the True Pareto front. To achieve this goal, the research framework started by determining the main limitations of the HHMO algorithm. To overcome these limitations, several improvements were proposed including proposing an improved population update strategy in Stage 1. Then, in Stage 2, an improved parameter adjustment strategy was proposed. This was followed by Stage 3, the where two-step initial population generator method was proposed, with the aim of more improvement in the search ability of the HHMO algorithm. The methods and strategies, proposed in Stages 1, 2 and 3, were integrated with other processes to design the 2S-ENDSHHMO algorithm in Stage 4. The 2S-ENDSHHMO was implemented using the MATLAB2018b platform. To check the effectiveness of the solutions generated by the 2S-ENDSHHMO algorithm, the ZDT, DTLZ and structural design engineering MOPs were used as a test basis, and the results of these algorithms compared with those of the multi-objective SI-based MHs, namely HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II. The performance of the algorithms was evaluated in terms of convergence toward the True Pareto front and diversity of the non-dominated solutions obtained by the algorithms. This was achieved by using two well-known performance metrics, namely epsilon, IGD and HV integrated with R-metrics approach.

CHAPTER FOUR
DEVELOPMENT OF TWO-STEP ENHANCED HARRIS'S HAWK MULTI-OBJECTIVE OPTIMIZER

4.1 Introduction

This chapter presents the development steps of the proposed enhanced Harris's hawk MOO algorithm. The HHMO algorithm is improved by integrating several methods and strategies. In this context, this chapter is organized as follows: Section 4.2 presents the original HHMO algorithm which has been used as the base for the proposed algorithm. This is followed by introducing the two-step enhanced non-dominated sorting Harris's hawk multi-objective optimizer (2S-ENDSHHMO) algorithm in Section 4.3. Section 4.4 describes the population update strategy. Section 4.5 presents the parameter adjustment strategy. Section 4.6 describes the proposed two-step initial population generator method. and the summary is presented in Section 4.7.

4.2 Harris's Hawk Multi-Objective Optimizer Algorithm

The HHMO algorithm (DeBruyne & Kaur, 2016) is a kind of SI-based optimization algorithm proposed by mimicking the social hierarchy and hunting behaviour of the Harris's hawk predator in nature (Bednarz, 1988). In the social hierarchy of the Harris's hawk, there are four social ranks, from high to low, alpha (α), beta (β), delta (δ) and gamma (γ) hawks. The hunting process is divided into two main stages, namely encircling and attacking, which closely resemble the encircling and attacking behaviour of grey wolves (DeBruyne & Kaur, 2016).

In the HHMO algorithm there are two types of hawks, namely lookouts and ground hawks. The former represents the reference points and the latter represents the potential solutions in the search space (search agents). The population of hawks is divided into sub-groups according to the number of reference points. In each group, the leaders X_α, X_β and X_δ are the hawks that have the three shortest distances to a reference point and they represent the first best three non-dominated solutions, respectively. The remaining hawks are represented by $X\gamma$. The main steps of HHMO algorithm are illustrated in Figure 4.1.

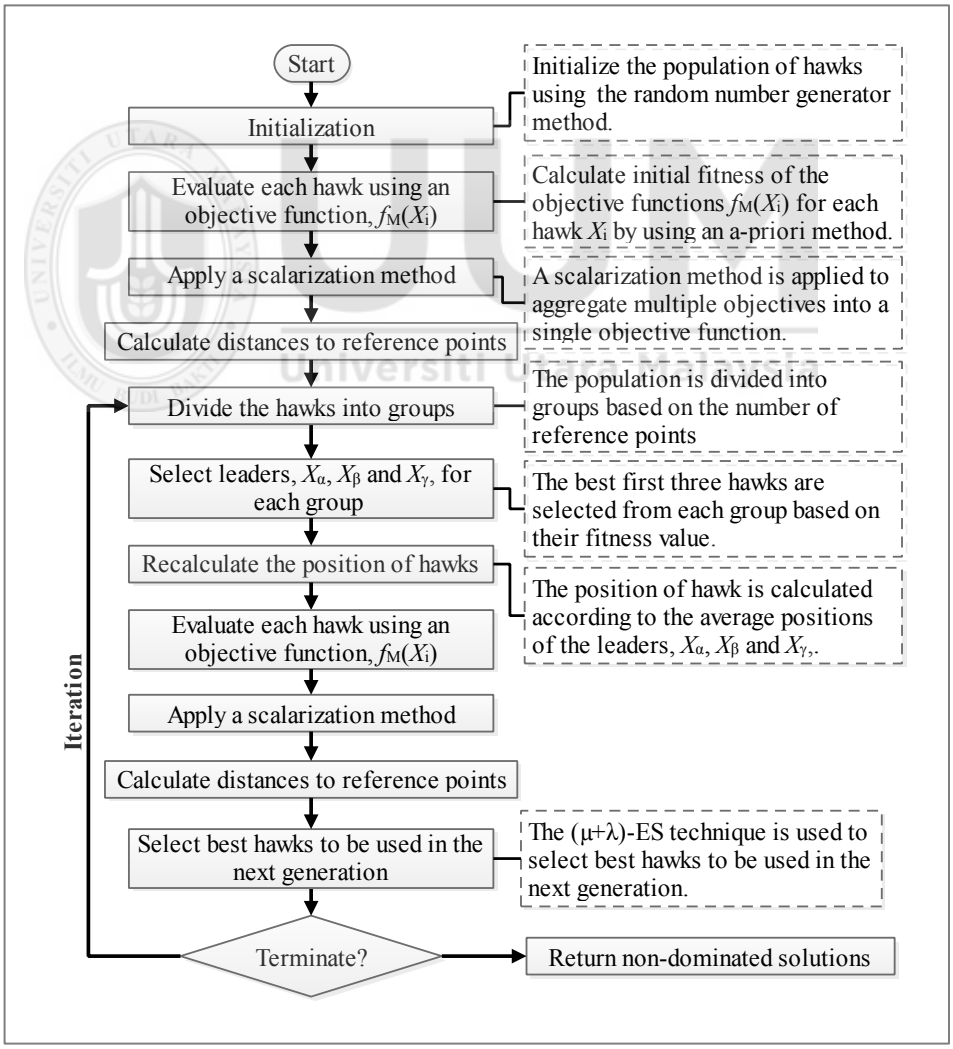


Figure 4.1. Flowchart of HHMO algorithm (Adapted from DeBruyne and Kaur (2016))

The HHMO algorithm starts by defining reference points for the sub-group and their search radius *r*. The population of hawks in each sub-group is initialized using the RNG method. The position of each hawk is evaluated by using multiple objective functions, $F(X_i) = [f_1(X_i), f_2(X_i), \dots, f_M(X_i)]$. In the HHMO algorithm, a scalarization method, such as the weighted sum and ε-constraint methods has been used to calculate the fitness values of each hawk (DeBruyne, 2018). Next, the Euclidean distances to the reference points are calculated for all $f(X)$. During the optimization process, the position of the hawks is updated based on the position of leaders X_α, X_β and X_δ. For each hawk, the objective functions and fitness values are calculated at each iteration. The best solutions are selected using the (μ+λ)-ES technique (DeBruyne & Kaur, 2016; DeBruyne, 2018) to be used in the next generation. The new leaders, X_α, X_β and X_δ, are selected for each sub-group and the hawks are relocated based on their distances to the reference points. These processes are iterated until they meet the termination condition. According to DeBruyne (2018) the main advantage of the HHMO algorithm consists of dividing the populations of hawks into sub-groups, which helps in improving the exploration and decreases the computational time and the complexity of the algorithm (DeBruyne, 2018). However, the performance of the HHMO algorithm can be improved further. This will be discussed in the next sections.

4.3 Two-step Enhanced Non-dominated Sorting Harris's Hawk Multi-objective Optimizer

The 2S-ENDSHHMO algorithm is developed based on the initial HHMO algorithm (DeBruyne & Kaur, 2016) with several improvements as follows:

a) Improved population update strategy of hawks.

b) Improved convergence parameter adjustment strategy.

c) Improved initial population generator method.

The 2S-ENDSHHMO algorithm is as depicted in Figure 4.2. The highlighted parts are the enhancement of HHMO algorithm.

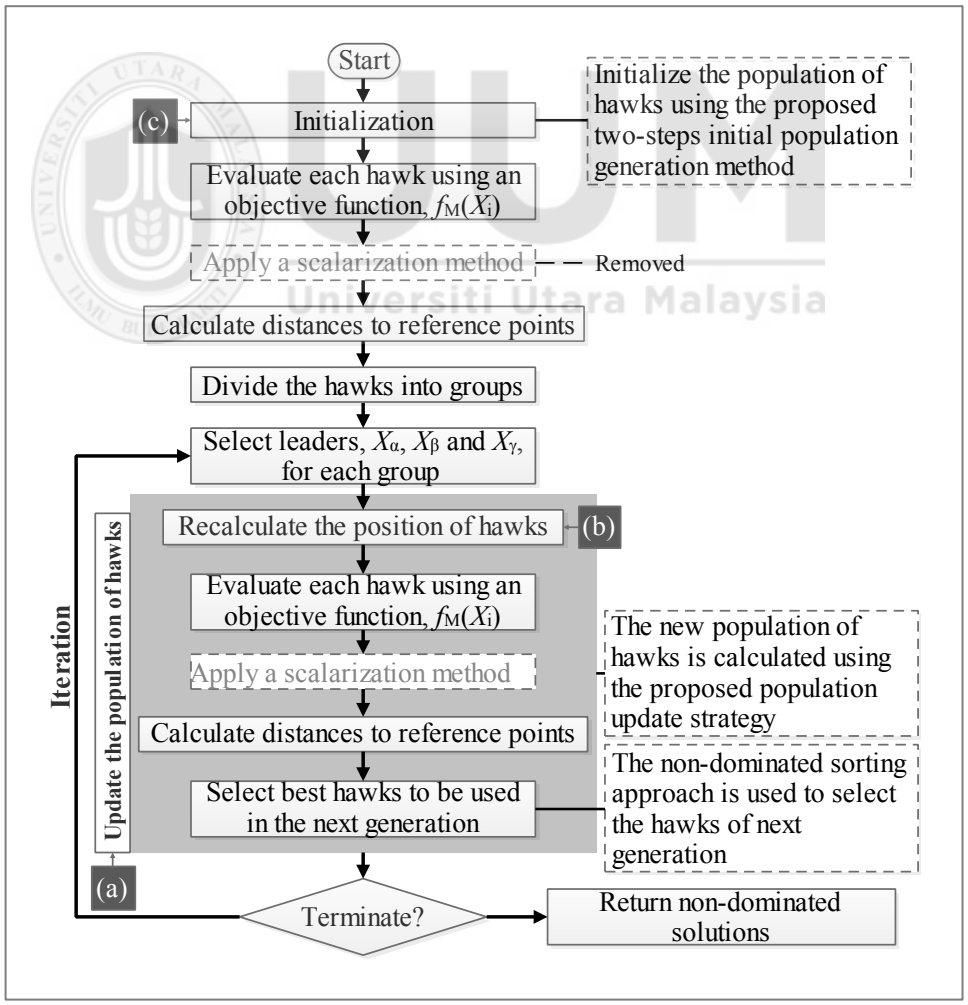


Figure 4.2. Main steps of 2S-ENDSHHMO algorithm

The enhancements are described in the following sections.

4.4 Population Update Strategy

In any optimization algorithm, the population update strategy can be considered as the core of the algorithm, which is used to produce new solutions. This section discusses the population updated in the HHMO algorithm and introduces the proposed strategy.

4.4.1 Population Update Strategy in HHMO

In the HHMO algorithm (DeBruyne & Kaur, 2016), the position of hawks represents candidate solutions in the decision space. To simulate the collective hunting behaviour of the Harris's hawks, it is assumed that X_α, X_β and X_δ hawks have a better understanding of the potential position of the prey. Therefore, during each iteration, their positions are saved and used to comprehensively determine the direction of the X_γ hawks and their positions are updated to move toward the prey. The behaviour of the group approaching and surrounding the prey is formulated as shown in Equation (4.1) (DeBruyne & Kaur, 2016).

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} * \vec{D} \quad (4.1)$$

$$\vec{D} = C * \vec{X}_p(t) - \vec{X}(t) \quad (4.2)$$

$\vec{D}$ is the distance between the hawk and the prey, which is calculated for each group; t is the number of current iterations, and $X_p = (x_{p1}, x_{p2};\ldots, x_{pd})$ is the position vector of the prey, while $X = (x_1, x_2, \cdots, x_d)$ represents the position vector of the hawks in d dimension. $\vec{A}$ and $\vec{C}$ are adjustment factors, calculated for each group and formulated as shown in Equation (4.3) and (4.4) (DeBruyne & Kaur, 2016), respectively.

$$\vec{A} = 2\vec{a} * r_1 - \vec{a} \quad (4.3)$$

$$\vec{C} = 2 * r_2 \quad (4.4)$$

where r_1 and r_2 are random vectors in interval [0,1]. a is the control parameter, in the range [0,2] and decreases linearly during the optimization process, with the number of iterations t. The average position of X_α, X_β and X_δ hawks is used to calculate a new position of hawks, for each group, as shown in Equations (4.5) (DeBruyne & Kaur, 2016).

$$\vec{X}(t+1) = \frac{(\vec{X}_\alpha(t) + \vec{X}_\beta(t) + \vec{X}_\delta(t))}{3}$$
$$\vec{X}_\alpha(t+1) = \vec{X}_\alpha(t) - \vec{A}_\alpha * \vec{D}_\alpha \quad (4.5)$$
$$\vec{X}_\beta(t+1) = \vec{X}_\beta(t) - \vec{A}_\alpha * \vec{D}_\beta$$
$$\vec{X}_\delta(t+1) = \vec{X}_\delta(t) - \vec{A}_\alpha * \vec{D}_\delta$$

According to the Equations (4.5), the HHMO algorithm only considers the position information of the three leaders, X_α, X_β and X_δ, and does not consider the positions of X_γ in calculating the new position of hawks (DeBruyne & Kaur, 2016; Long et al., 2019). This leads to a loss of population diversity due to high selection pressure in which the algorithm depends only on the three best solutions to guide the search process (Al-Betar, Awadallah, Faris, Aljarah, & Hammouri, 2018). The high selection pressure and loss of population diversity leads to poor convergence toward the Pareto front (Hussain & Muhammad, 2019).

4.4.2 Proposed population update strategy

This section presents the proposed population update strategy integrated with the HHMO algorithm to produce a non-dominated sorting Harris's hawk multi-objective

optimization (NDSHHMO) algorithm. The proposed population update strategy of hawks consists of two main stages. The first stage is to calculate the new position of hawks by using a new movement strategy and the second stage is to select the hawks to be used in the next generation. Figure 4.3 shows the stages of the proposed population update strategy.

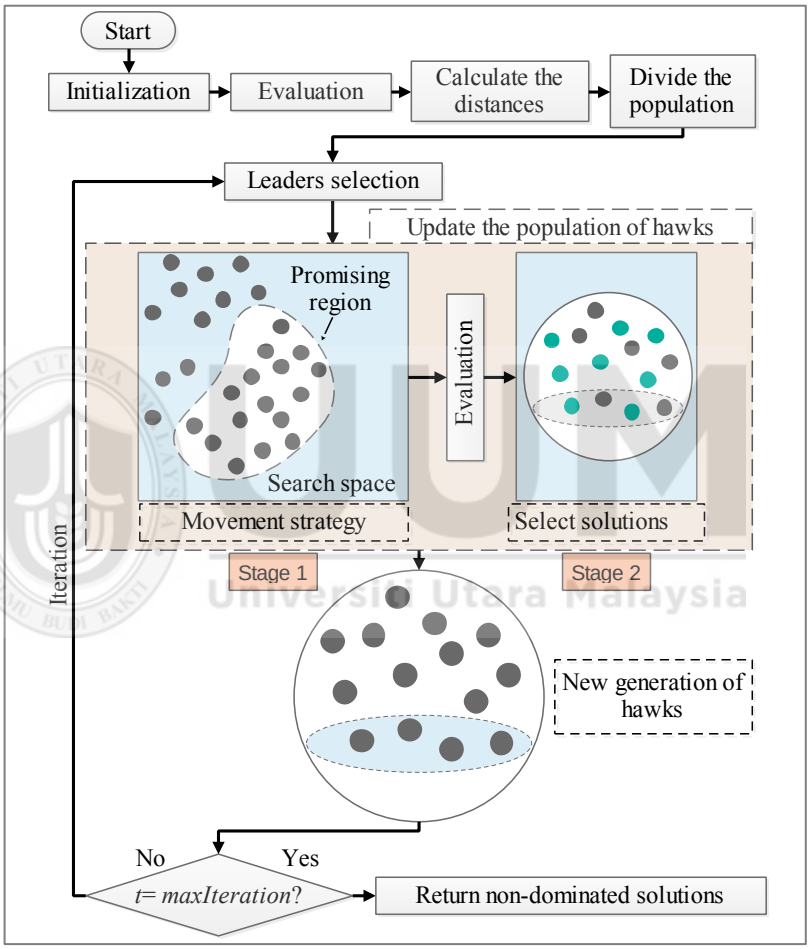


Figure 4.3. Proposed population update strategy

As shown in Figure 4.3 the NDSHHMO algorithm starts by initializing the population of hawks using the random number generator method. Each hawk is evaluated using the objective function. In the first stage, a new movement strategy is proposed to calculate the new position of hawks.

The new proposed movement strategy is developed based on the hunting behaviour employed by Harris's hawks in nature. The Harris's hawk attacks are quite coordinated. According to (Bednarz, 1988), who observed Harris's hawks over a period of years, the hunting behaviour of Harris's hawks involves different tactics. These tactics are often intermixed and intensity varies in an unpredictable sequence, apparently dependent on the changing circumstances that occur during pursuit of prey. One of these tactics is called the flush-and-ambush (Bednarz, 1988). This tactic is employed when a prey finds temporary refuge or cover, as illustrated in Figure 4.4.

Figure 4.4. Flush-and-ambush tactic: the prey finds temporary refuge or cover

In the flush-and-ambush tactic, the hawks are alert in watching the location where the prey disappeared; meanwhile, one or possibly two hawks attempt to penetrate the

cover. Then, when the prey is flushed, one or more of the hawks pounce and kill the prey (Bednarz, 1988). Based on the hunting tactics of hawks, the proposed movement strategy is formulated as shown in Equation (4.6).

$$\vec{X}(t+1) = \begin{cases} \dfrac{\left(\vec{X}_{\alpha}(t) + \vec{X}_{\beta}(t) + \vec{X}_{\delta}(t)\right)}{3} & ; \quad \text{if } p \geq 0.5 \\ \vec{X}_{\mathrm{FA}} & ; \quad \text{otherwise} \end{cases} \tag{4.6}$$

where p is a random value in interval [0,1]. In the proposed movement strategy, the new position, $\vec{X}(t+1)$ is calculated based on the random-proportional rule. This rule is an action choice rule typically used in Q-learning (Dorigo & Gambardella, 1996). With this rule, the action is chosen randomly with a probability of 50%. This means that the old and proposed position update strategy have exactly the same probability to be chosen to calculate new position of hawks. The random-proportional rule, also has been used in other algorithms, such as the single objective whale (Mirjalili & Lewis, 2016) and Harris's hawk (Heidari et al., 2019) optimization algorithms. Figure 4.5 illustrates the proposed movement strategy.

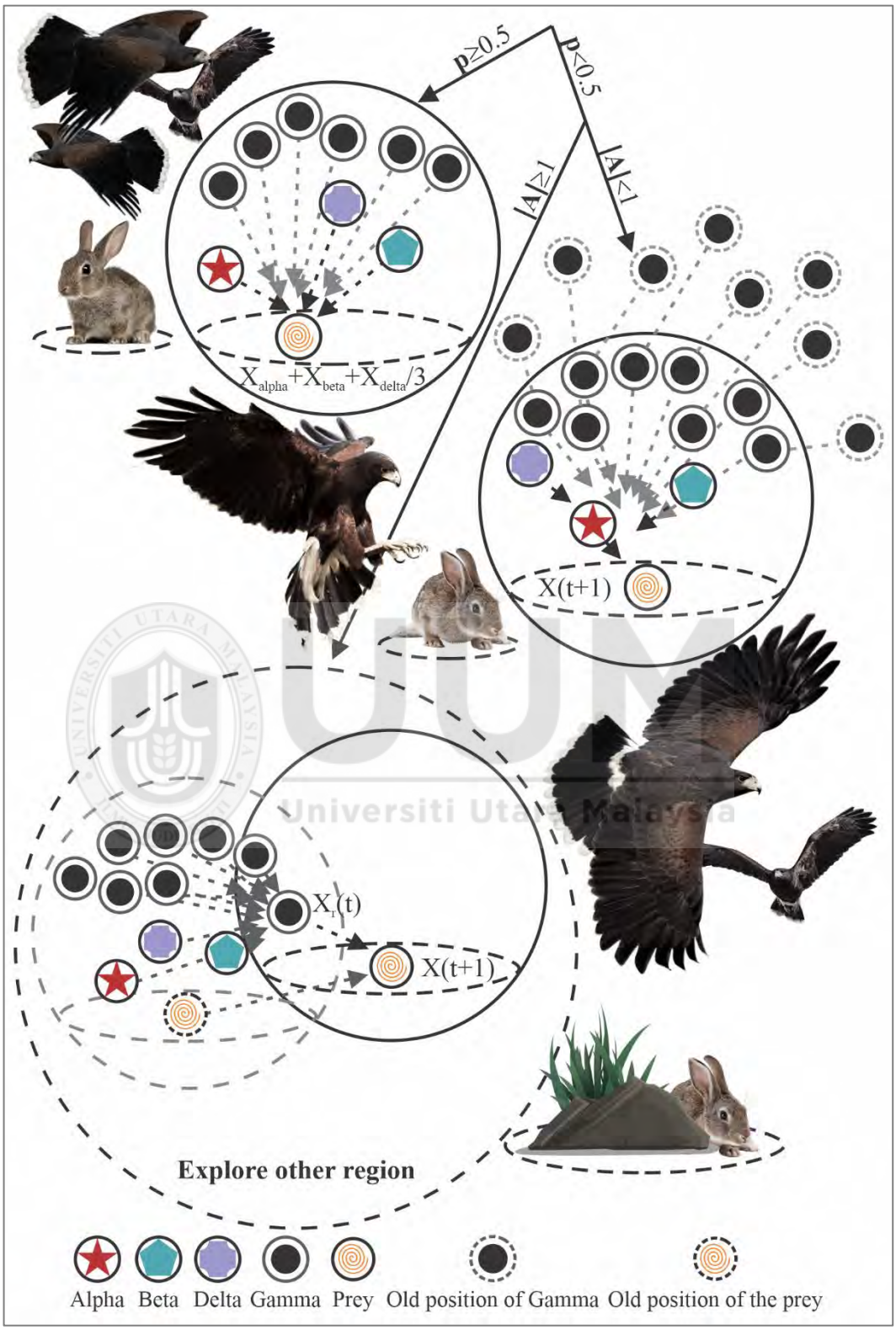


Figure 4.5. Proposed flush-and-ambush movement strategy

If $p \geq 0.5$, this indicates that X_α, X_β and X_δ hawks have spotted the location of prey. In this case, Equations (4.5) in the original update strategy has been be employed to update the positions of hawks according to the positions of leaders. Otherwise, if the pray escapes away, the positions of hawks will be updated based on the proposed flush-and-ambush movement strategy, which is represented by $\vec{X}_{\mathrm{FA}}$ value, as shown in Equation (4.7).

$$\vec{X}_{\mathrm{FA}}(t+1) = \begin{cases} \vec{X}_r(t) - \vec{A} * \vec{D}_r & ; \quad \text{if } |A| \geq 1 \\ \vec{X}_\alpha(t) - \vec{A} * \vec{D}_\alpha & ; \quad \text{otherwise} \end{cases} \tag{4.7}$$

The value of $\vec{X}_{\mathrm{FA}}$ is proportional to the value of $|A|$ where $|\ |$ is the absolute value. In the proposed flush-and-ambush movement strategy, the same approach based on the variation of the A vector can be utilized to search for a prey (Mirjalili et al., 2014). In this approach, the hawks move forward and backward from the prey based on the value of A, as shown in Figure 4.6.

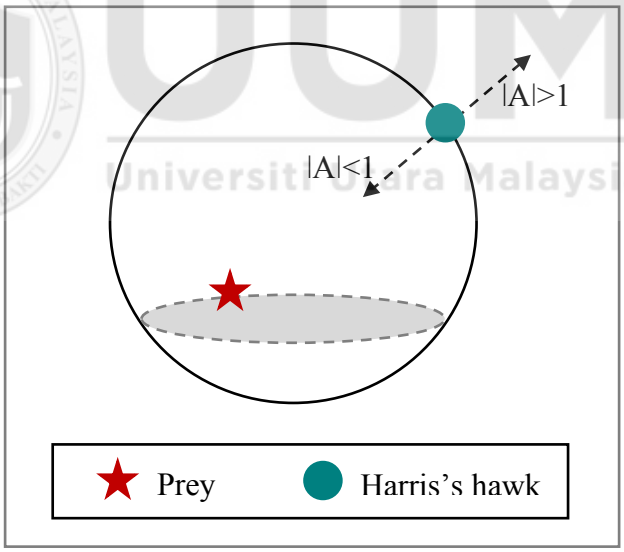


Figure 4.6. Direction of hawk based on the value of A. If $|A| < 1$, a hawk moves toward a prey; If $|A| > 1$, a hawk moves far away from a prey position (Adapted from Mirjalili et al. (2014)).

If $|A| \geq 1$, the hawks will explore the desert site looking for a potential prey (DeBruyne & Kaur, 2016). If $|A| < 1$ forces the hawks to move towards the prey. In the proposed flush-and-ambush movement strategy, Equation (4.7), if $|A| \geq 1$, this

indicates the prey has successfully escaped from the hawks and found a temporary cover (refer Figure 4.5). In this case, $\vec{X}_{\mathrm{FA}}$ is calculated using a random hawk, $\vec{X}_r$, which is selected from the current population, represented by X_γ hawks, to guide the search process. The random position of the hawk represents the exploration of different regions to find the location of the covered prey. If $|A| < 1$, this represents penetrating the cover of the prey. In this case, the X_α hawk makes the final move to kill the prey. In other words, the position of the hawk is calculated according to the position of $\vec{X}_\alpha$, in a group, which represents nearest Hawks to the prey position.

The second stage of updating the population of hawks requires selecting the non-dominated solution to be used in the next generation. To select the non-dominated solutions, the non-dominated sorting approach is used (Deb & Sundar, 2006). In this approach, the population of parent and offspring are combined to produce a population of size 2*N*. This population is sorted and classified according to the Pareto-dominance relation between the solutions, forming several front levels. Figure 4.7 illustrates the principle of non-dominated sorting.

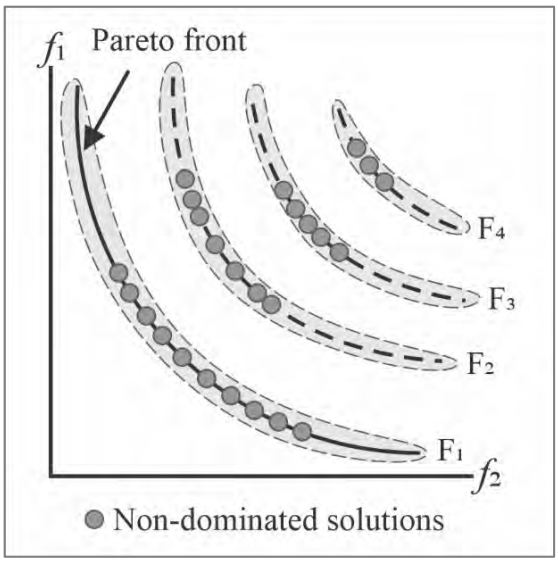


Figure 4.7. Dividing a population into four levels of front (F1, F2, F3, F4) by the non-dominated sorting approach (Adapted from Sadatsakkak, Ahmadi, Bayat, Pourkiaei, and Feidt (2015))

In Figure 4.7, the individuals that have the best quality in the population are considered as a first level of frontier, F_1 and assigned the rank 1. Subsequently, these individuals are temporarily eliminated from the competition. The non-dominated individuals in the remaining population are selected to construct the second level of frontier, F_2 and assigned the rank 2. These processes are repeated until there is no individual left. In this way, the population is divided into multiple non-dominated frontiers, each defining a specific quality level. Figure 4.8. shows the main steps of non-dominated sorting approach (Deb et al., 2002a).

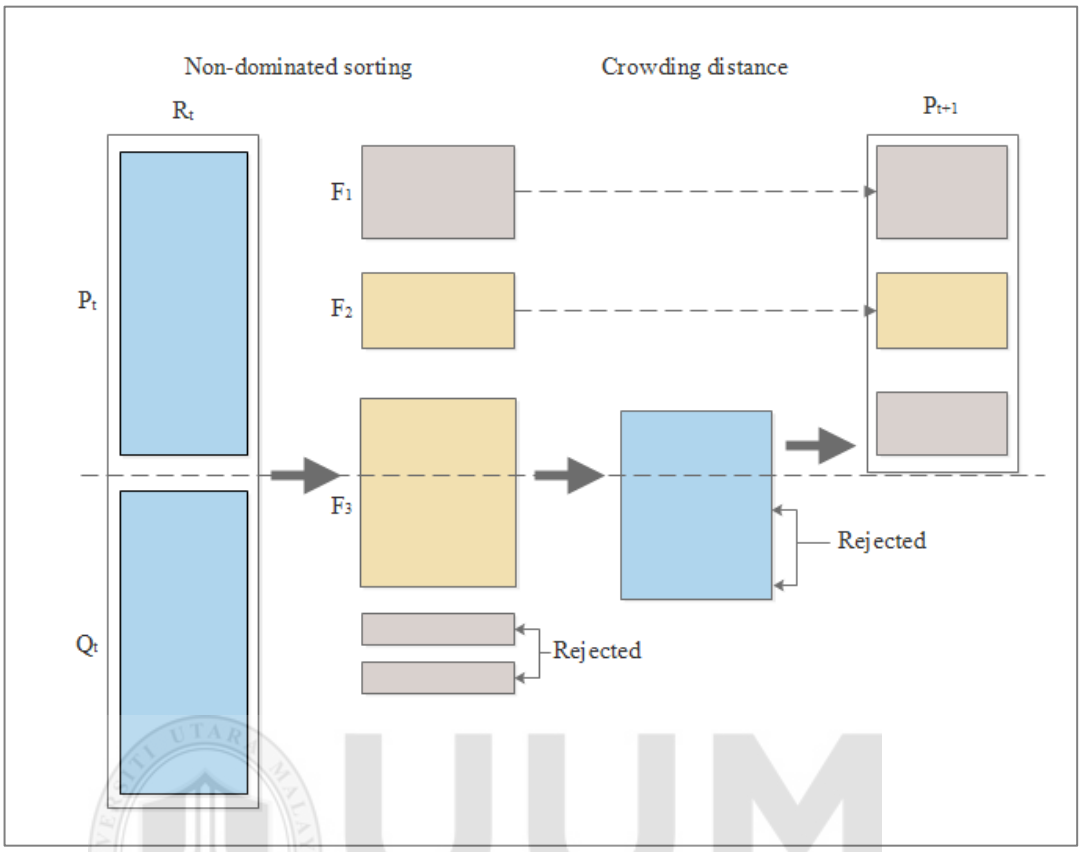


Figure 4.8. Concept of non-dominated sorting approach (Adapted from Deb et al. (2002a))

To perform non-dominated sorting, the two population parents, P_t and off-spring, Q_t are combined into a single R_t population composed of $2N$ individuals. To select the best solutions, the elements of the R_t population are sorted and classified according to the Pareto dominance relation between the solutions, forming several front levels, namely F_1, F_2 and F_3.

The first front level, F_1, corresponds to the non-dominated solutions of the entire population, the second, F_2, to all non-dominated solutions, after the removal of the solutions from the first front, the third, F_3, to non-dominated solutions after removing the first and second fronts, and so on until all solutions are sorted into several fronts.

In general, with non-dominated sorting-based algorithms, when making survival choices, if the number of solutions in F_1 is less than the predefined population size, the rest will be selected from the next front, F_2. If the total number of selected solutions exceeded the population size, N, the solutions of F_1 will be moved to the next generation and the rest will be selected from F_2 based on another quality criterion.

Several studies have proved the effectiveness of the non-dominated sorting (Deb et al., 2002a)with many MOO algorithms (Cai, Li, Fan, & Zhang, 2015; Chen et al., 2019; Deb et al., 2002a; Jangir & Jangir, 2018; Tian, Wang, Zhang, & Jin, 2017; Tian, Zhang, Cheng, & Jin, 2016; Zhou & Yao, 2017). The non-dominated sorting approach helps in improving the convergence of the algorithm towards the true Pareto front, especially for dealing with complex MOPs with a large number of local PFs (Tian et al., 2017).

In the non-dominated sorting approach (Deb et al., 2002a), the crowding distance is used to determine which individuals will survive for the next generation. The crowding distance value of a particular solution represents the average distance of its two neighbouring solutions. Solutions that are on the edge of the search space (extreme solutions) have only one neighbour, but they are the most diverse of the border, so they obtain high values and, consequently, are at the top of the order. The solutions with bigger crowding distance are preferred. However, in some cases, the crowding distance approach cannot be used to select appropriate solutions, which may affect the diversity of solutions. Figure 4.9 illustrates the limitation of the crowding distance approach.

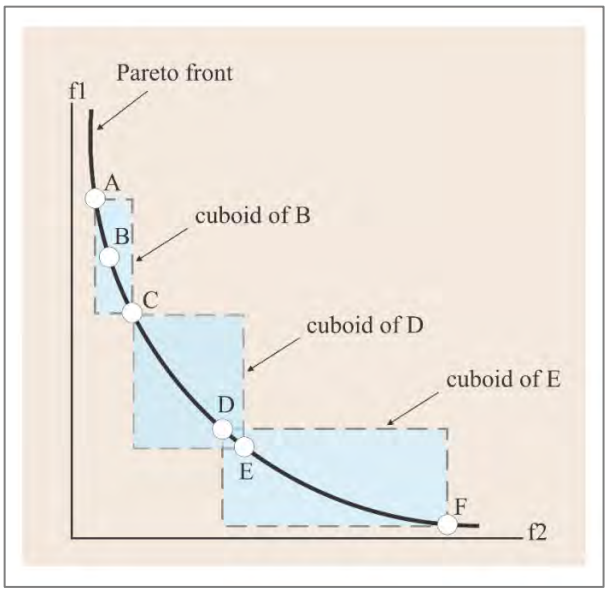


Figure 4.9. Drawback of the crowding distance approach (Adapted from Vachhani et al. (2016))

In Figure 4.9, A, B, C, D, E and F represents non-dominated solutions in the Pareto front. The cuboids represents the crowding distance of all solutions except, the extreme solutions, namely F and A, which have an infinite crowding-distance (Deb, 2014). Five out of six solutions should be selected. To maintain population diversity, solution *B* should be selected with any one out of solution number *D* and *E* (Vachhani et al., 2016). However, based on the concept of crowding distance, solutions *D* and *E* are selected instead of solution *B* because they have larger crowding distance values. In this case, the diversity of the selected solutions is not preserved and leads to poor population diversity (Vachhani et al., 2016).

In this study, the crowding distance measure is replaced by an epsilon-clearing (ε-clearing) strategy, as suggested by Deb and Sundar (2006). This strategy is used as a second quality criterion to select between individuals that belong to the same front (have the same rank). This strategy divides the objective space into grids of size, ε.

Solutions with a difference less than ε in the i-th objective are not allowed to be non-dominated with each other. This helps in preserving diversity among solutions of the same front. If there are more than enough points to complement the new population, the Euclidean distance is used to select the individual with minimum distance to the reference points. The population update strategy of hawks aims to improve the algorithm in terms of convergence toward the true Pareto front and maintain population diversity**.** It is worth mentioning that the value of ε in the ε-clearing strategy allows the DM to control the density of the Approximated Pareto front by choosing an appropriate value (Laumanns, Thiele, Deb, & Zitzler, 2002).

In DeBruyne and Kaur (2016); DeBruyne (2018), to enhance the performance of the HHMO algorithm, the authors integrated the selection of the best (μ+λ)-ES (Rechenberg, 1989; Schwefel, 1993), with HHMO. In this strategy, the parent and offspring populations are combined, then the best individuals are selected to produce the next generation. The best hawks are selected based on the fitness value, which is calculated using the weighted sum or ε-constraint method. However, these methods have limitations, as discussed in chapter two. To overcome these limitations, another method is required to select non-dominated solutions and improve the convergence toward the true Pareto front. Compared to the position update strategy in the HHMO algorithm, the proposed position update strategy takes in consideration the positions of other hawks, X_γ, in updating the position of hawks in the search space. This aims to promote sharing the information between the hawks in the population. Sharing the information preserves the population diversity, prevent the HHMO algorithm from

trapping in local optima and improve the convergence toward the Pareto front (Allawi et al., 2019; Zhang, Liu, Yang, & Dai, 2016).

In the NDSHHMO, the calculation of fitness value by using the scalarization method has been eliminated. Instead, the normalized Euclidean distance (Deb & Sundar, 2006) from each solution in the objective space to the reference points is calculated and used as a fitness value for each hawk. The main advantages of this process are reducing the number of parameters, represented by the ε value in the ε-constraint method and the weight vector in the weighted sum method, which significantly affects the performance of the algorithm. Inappropriate selection of these parameters may lead to poor convergence toward the True Pareto front. To produce the next generation of hawks, first, the new position of hawks is updated using the proposed movement strategy. Then, these hawks are evaluated using the objective function. This is followed by performing the non-dominated sorting approach to select the best hawks that will be moved to the next generation.

4.5 Parameter Adjustment Strategy

The parameter of an algorithm plays an important role in controlling its performance. According to Eiben et al. (2007), parameter setting/configuration directly affects the quality of the final solution and it is necessary to know the most promising configuration that should be used. The convergence parameter, a, affects hawks in the process of updating their position. Therefore, in this study, an improved parameter

adjustment strategy is proposed to improve the performance of the HHMO algorithm. This algorithm is called enhanced HHMO (EHHMO).

4.5.1 Parameter Adjustment Strategy in HHMO

According to Equation (4.3), the control parameter *a* is mainly used to generate the coefficient vector, *A* which, in turn, affects the exploration and exploitation of the algorithm. If $|A| > 1$, the hawks group will expand the encirclement to explore the search space for a more promising region; when $|A| < 1$, the hawks will narrow the search range in a promising region to find the best non-dominated solutions (refer to Section 4.4) (Mirjalili et al., 2014). Therefore, the global exploration and local exploitation ability of the HHMO algorithm depends to a large extent on the value of convergence parameter *a*. The adjustment strategy of the convergence parameter *a* is as shown in Equation (4.8) (DeBruyne & Kaur, 2016).

$$a = 2 - t(2/MaxIteration) \tag{4.8}$$

where *t* current iteration and the value of $t/MaxIteration$ increases on interval [0,1]. In general, in metaheuristics, the optimization process starts by exploring the search space, to find a promising region. Then, in the later stage of optimization, the algorithm should be focused more on a particular region to find the best set of solutions. In the HHMO, according to the adjustment strategy of convergence parameter *a* in Equation (4.8), the first half of the iteration is allocated for the global search to find a promising region followed by local search within a promising region in the second half. This strategy overlooks the state of search space and the right balance between exploration and exploitation (Yan et al., 2019), which leads to a loss of population diversity and

poor approximation for the Pareto front (Barbosa & Senne, 2017; Huang et al., 2019; Yan et al., 2019; X.-S. Yang et al., 2019).

4.5.2 Improved Parameter Adjustment Strategy

The balanced trade-off between the exploration and exploitation processes is important to achieve an effective performance for the EHHMO algorithm. The balanced trade-off does not mean a 50:50 split (Hussain et al., 2019). According to X. S. Yang et al. (2014a), an effective optimization algorithm consists of balanced trade-off with a global search, which is often slow, and a fast local search. In general, in solving a complex optimization problem an optimization algorithm is required to perform more exploitation to find the global optimal solution. For example, solving a multimodal problem, requires more exploration than solving a unimodal problem (Črepinšek, Liu, & Mernik, 2013). Therefore, since a MOP is more complex than a SOP, in MOO, the requirement for the ability to explore the search space by an algorithm is high (Oliva & Elaziz, 2020). In this context, the proposed parameter adjustment strategy is developed to give more chance for the EHHMO algorithm to explore the search space in the early stage of the optimization process followed by fast exploitation in the later stage.

In the proposed parameter adjustment strategy, the value of the convergence parameter a is adjusted by adaptively controlling the decremented steps of the convergence parameter a. Several studies have demonstrated the feasibility of the adaptive adjustment strategy of control parameters in improving the performance of SI-based

optimization algorithms (Amoshahy, Shamsi, & Sedaaghi, 2016; Liu, Lu, Cheng, & Shi, 2019; Parpinelli et al., 2019; Song, Liu, & Cheng, 2019; Wang, 2015). Inspired by these studies, in this thesis, an adaptive adjustment strategy is proposed to control the parameter *a*. Equation (4.9) shows the proposed parameter adjustment strategy of the convergence parameter *a*.

$$a = 2 - 2\left(\frac{t}{MaxIteration}\right)^{DF} \tag{4.9}$$

$$DF = \frac{1}{\sqrt{Score}} + 2$$
$$Score = PD\big(f(X)\big); \tag{4.10}$$

where the diversity factor (*DF*) is calculated based on the pure diversity (PD) method (Wang, Jin, & Yao, 2016). This method was developed based on the dissimilarity measure represented by L_p-norm-based distance, where p=0.1, as suggested in the paper (Wang et al., 2016). The main advantages of the PD method is parameter-free and does not require a reference set to calculate the diversity of solutions (Wang et al., 2016). In Equation (4.10), the value of *DF* is adaptively adjusted based on the diversity of the population in the current iteration, *t* and $MaxIteration$ represent the total number of iterations. *Score* represents the diversity value calculated by using the PD method. The input of PD is *f*(*X*), which represents the objective values of the current population. During the iterative search process, the value of parameter *a* decreases based on the *DF* value, as shown in Figure 4.10 which describes the difference between the original and improved adjustment strategies.

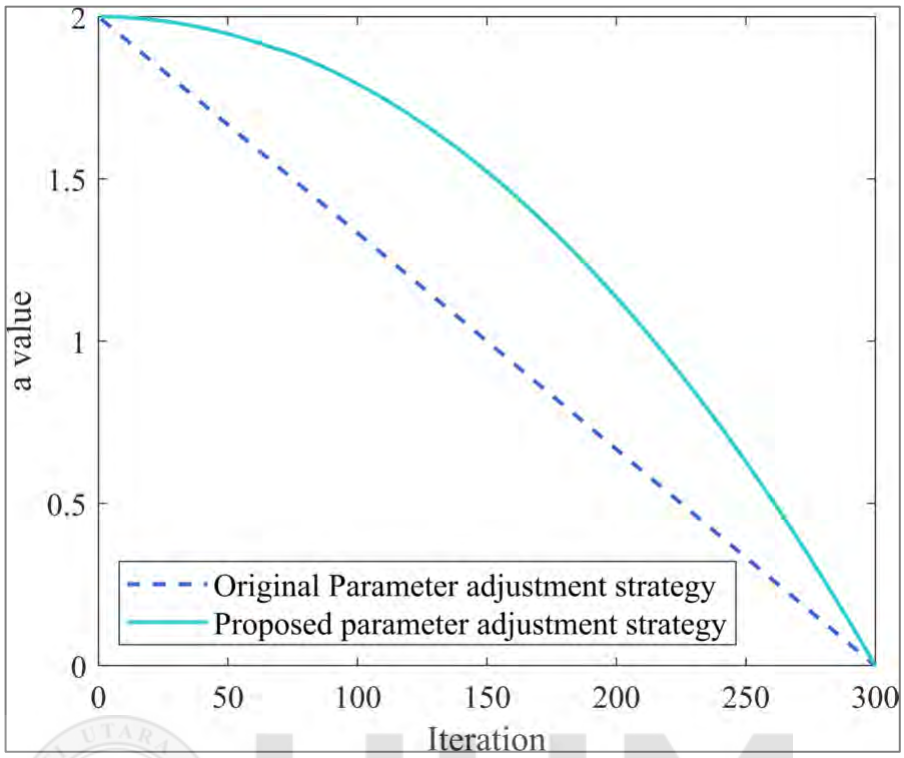


Figure 4.10. Original and proposed improved parameter adjustment strategies.

Difference between original and proposed improved parameter adjustment strategies:

The dashed line and continuous line in Figure 4.10 represent the fluctuations of convergence parameter *a* value with number of iterations in the original and proposed adjustment strategy, respectively. In the original parameter adjustment strategy, the first half of the iterations are allocated for the global search to find a promising region followed by local search within a promising region in the second half. In other words, exploration and exploitation perform equaly with an increasing number of iterations. However, the balance between exploration and explotation does not imply that the exploration and exploitation processes ane equally performed (Hussain et al., 2019). Therefore, this strategy overlooks the balanced trade-off between the exploration and exploitation, which leads to a loss of population diversity and poor approximation for

the Pareto front (Barbosa & Senne, 2017; Huang et al., 2019; Yan et al., 2019; X.-S. Yang et al., 2019).

On the other hand, the proposed parameter adjustment strategy of the convergence parameter *a* is developed to gives more chance to explore the search space in the early stage of the optimization process. In the later stage, the value of convergence parameter *a* decreases rapidly to perform the exploitation process. This aims to maintain population diversity and avoid being trapped in local optima, which leads to premature convergence. During the optimization process of the algorithm, the value of *A* changes continuously with the change of the convergence parameter *a*. Figure 4.11 shows the difference between the fluctuations of the *A* value obtained using original and improved adjustment strategies, respectively.

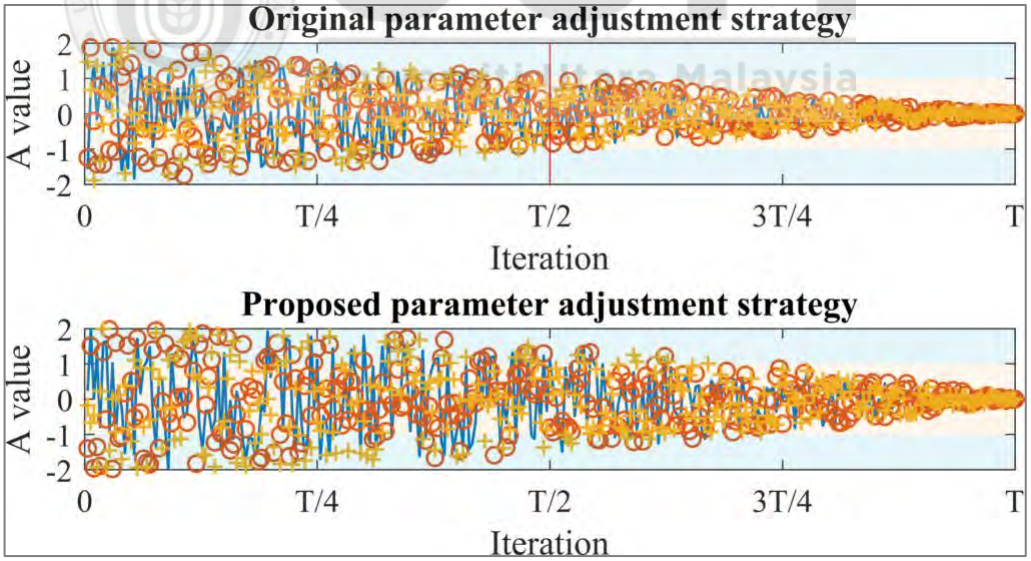


Figure 4.11. Original and proposed parameter adjustment strategies with respect to *A* value.

In Figure 4.11, the parameter adjustment strategies in the HHMO and EHHMO algorithms are run three times. In each run the values of *A* are calculated using

Equation (4.3). The value of the convergence factor, *A,* changes according to the value of the parameter, *a*. If $|A| \geq 1$, the hawks will explore the desert site looking for a potential prey (DeBruyne & Kaur, 2016). If $|A| < 1$ the hawks are forced to move towards the prey.

In the proposed parameter adjustment strategy, the value of *A*, in the first half or iterations (T/2), is bigger than one which helps the EHHMO algorithm to spend more time on exploring the search space at the early stage of optimization process. On the other hand, in the original parameter adjustment strategy, the value of *A* becomes less than one in the first half which indicates poor exploration the early stage. Integrating the proposed adjustment strategy of the convergence parameter, *a* with HHMO aims to increase the proportion of the exploration to search for more feasible solutions. To ensure better convergence toward the Pareto front, the values of parameter *a* change rapidly in the later stages of the algorithm, thereby, obtaining better solutions accuracy.

4.6 Initial Population Generator Method in HHMO

In the MOSI-based metaheuristic, the definition of the initial population has a significant impact on the performance of the algorithm (Digehsara et al., 2020; Kazemzadeh Azad, 2018; Talbi, 2013). In the generation of the initial population, the main criterion to be considered is distribution of the solutions in the search space (Digehsara et al., 2020). If an initial population is well distributed, this will lead to improve the convergence of the solutions during the optimization process (Cruz-Chávez & Martínez-Oropeza, 2016; Digehsara et al., 2020).

4.6.1 Initial Population Generator Method in HHMO

The HHMO algorithm uses the RNG method to initialize the population of solutions (DeBruyne & Kaur, 2016). In this method, the individuals are randomly generated within pre-determined boundaries (domain of the problem) and the search space is sampled approximately uniformly. In other words, the initial population is generated using a RNG between two boundaries, as shown in Equation (4.10) (Talbi, 2009).

$$\vec{X} = \overrightarrow{lb} + \vec{r} * \left(\overrightarrow{ub} - \overrightarrow{lb}\right) \tag{4.10}$$

where *r* is a random number in interval [0,1]. *lb* and *ub* are lower and upper bounds of the decision variables, respectively. However, as shown in Figure 4.12, the RNG method does not guarantee that the solutions can be evenly distributed in the search space of the problem, which may affect the efficiency of the HHMO algorithm (Jana et al., 2018).

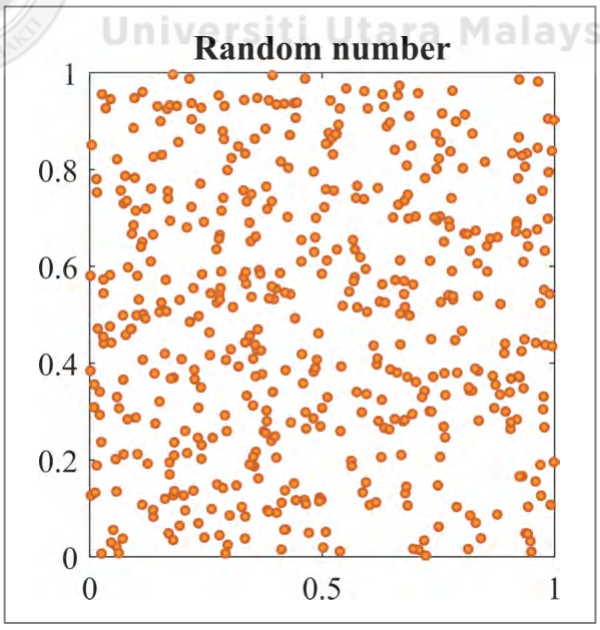


Figure 4.12. Sampling using random number generator method

In Figure 4.12, the simple random sampling of a point inside a unit square exhibits a clustering of points and that there are also regions that contain no points at all ('white noise'). Therefore, in this study, the R-sequence method (Roberts, 2018; Slater, 2019) will be used instead of the RNG to generate random numbers.

4.6.2 Proposed Two-step Initial Population Generator Method

In this study, the proposed initial population generator is developed based on the Multi-step technique. This technique composes of two or more techniques in which at least one of them cannot be used autonomously (Kazimipour et al., 2014). The proposed initial population generator method consists of two steps. The first step consists of the generation of random numbers using the R-sequence (Roberts, 2018). In the second step, the proposed partial OBL approach is applied on the generated random numbers to produce the initial population. Figure 4.13 illustrates the proposed two-step initial population generation method.

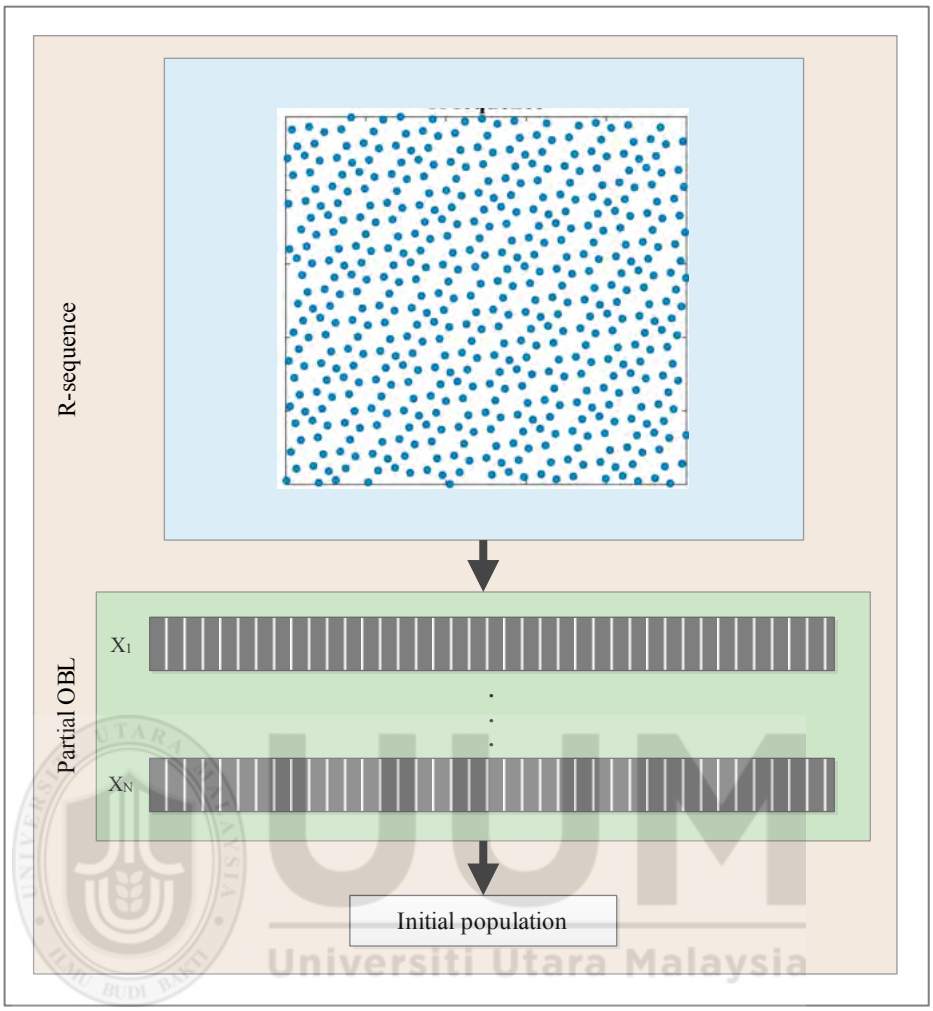


Figure 4.13. High level flowchart of the proposed two-step initial generator methods

i. Step one: R-sequence

In this step, the R-sequence is used to generate low-discrepancy sequences with high levels of uniformity. The *n*-th terms of R-sequence in *d* dimension are generated as shown in Equation (4.11) (Roberts, 2018; Slater, 2019).

$$R_d(\emptyset_d): t_n = \alpha_0 + n\alpha (\text{mod } 1); \;\; n = 1,2,3, \ldots \tag{4.11}$$

$$\alpha = \left(\frac{1}{\emptyset_d}, \frac{1}{\emptyset_d^2}, \frac{1}{\emptyset_d^3}, \ldots \frac{1}{\emptyset_d^d} \right) \tag{4.12}$$

where α is any irrational number. $\emptyset = \frac{1+\sqrt{5}}{2}$ is the golden ratio. The generalized version of $\emptyset_d$ is defined as the unique positive root of x^{d+1}=x+1 and α_0=0.5 (Roberts, 2018; Slater, 2019).

The calculation of R-sequence for 3-*d* is as follow:

For d=1, $\emptyset_1 = 1.618033989$, which is the canonical golden ratio.

For d=2, $\emptyset_2 = 1.3247179572$, which is often called the plastic constant.

For d=3, $\emptyset_3 = 1.2207440846$

$\alpha = (0.819173, 0.671044, 0.549700)$

$R_d(\emptyset_d) =$
(0.319173, 0.171044, 0.0497005);
(0.138345, 0.842087, 0.599401);
(0.957518, 0.513131, 0.149101);
(0.77669, 0.184174, 0.698802)
(0.595863, 0.855218, 0.248502) ...

ii. Step two: Partial opposition-based learning

The main idea of OBL is to calculate a solution located in the opposite direction, $\breve{X}$ of a candidate solution, $\vec{X}$. Figure 4.14 illustrates the concept of OBL.

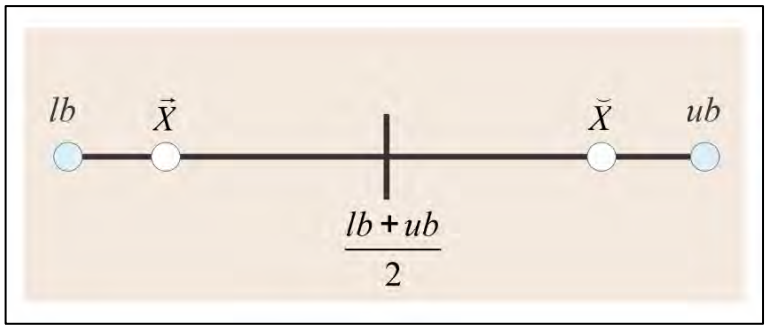


Figure 4.14. Concept of OBL in one dimension (Adapted from Tizhoosh (2005))

Both, $\vec{X}$ and $\check{X}$, simultaneously, give a higher chance to find the promising region (Mahdavi, Rahnamayan, & Deb, 2018). The opposite point in the *d*-dimension space is calculated as shown in Equation (4.13) (Tizhoosh, 2005) .

$$\check{X} = \overrightarrow{lb} + \overrightarrow{ub} - \vec{X} \tag{4.13}$$

where *ub* and *lb* are upper and lower bounds, respectively. In the partial OBL a solution located in the opposite direction contains opposite numbers to the original numbers in some dimensions with respect the degree of partial opposition (Hu et al., 2014).

The partial OBL strategy proposed in this thesis is different from the traditional partial OBL (Hu et al., 2014; Si & Dutta, 2019). The tradition partial OBL is developed to work with single objective optimization algorithm. The partial OBL in the proposed two-step method is developed to initialize the population for the reference-point based MOO algorithms. In the tradition partial OBL, the opposite population is calculated for all solutions in the population. However, in the two-step initial population generator method, the population is first divided into two subpopulations and the opposition value of a randomly selection dimension is calculated. Then, Pareto-dominance relation is applied to compare and select the solutions. Figure 4.15 illustrates the partial OBL.

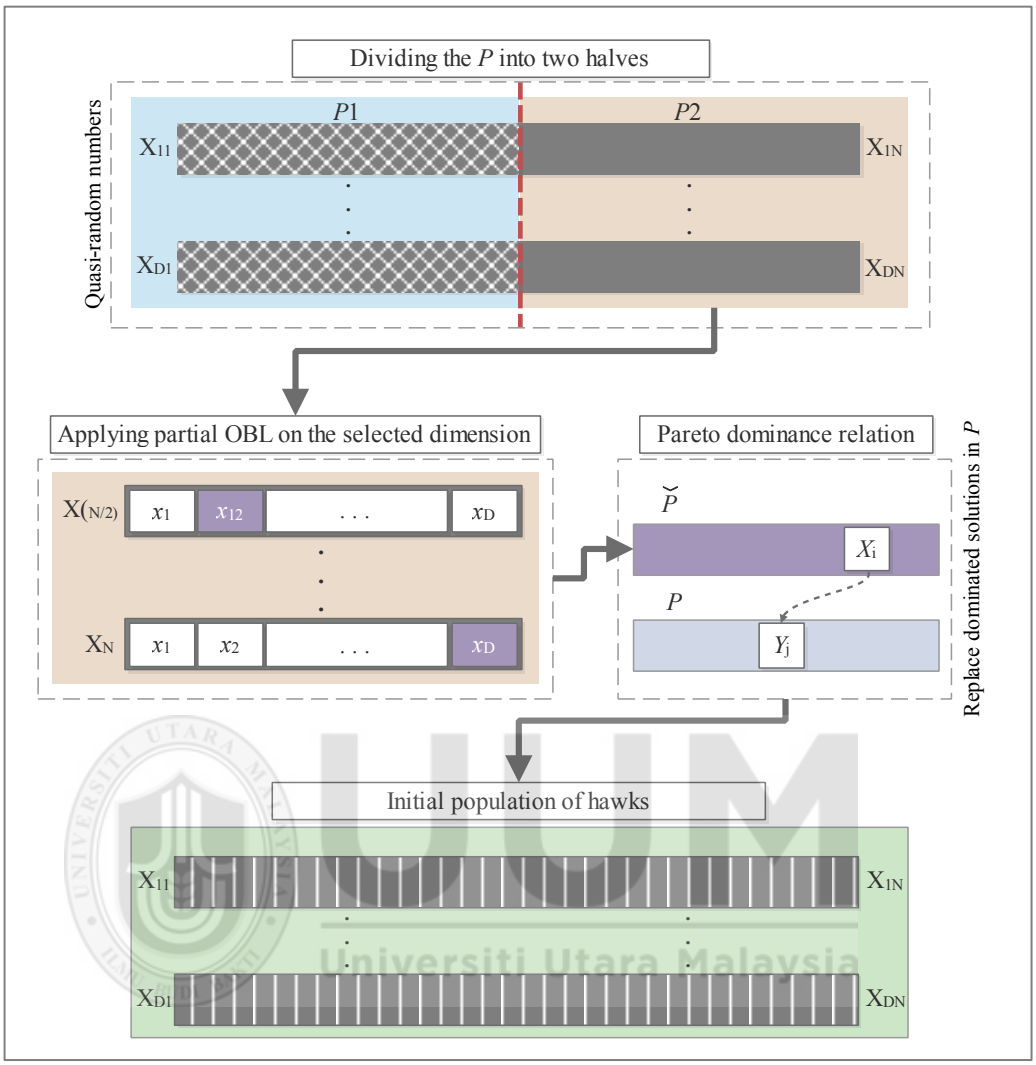


Figure 4.15. Partial OBL in the 2S-ENDSHHMO algorithm

In the proposed partial OBL concept, the population of hawks, P is sorted based on the fitness value and divided into two halves, $P1$ and $P2$. The fitness value represents the normalized Euclidean distance (Deb & Sundar, 2006) between each point in the objective space, $f(X)$ and the reference point Z, calculated as shown in Equation (4.14) (Deb & Sundar, 2006).

$$ED = \sqrt{\frac{\sum_{i=1}^{M}(f_i(x) - Z_i)^2}{\left(\mathrm{Max}(f_i(x)) - \mathrm{Min}(f_i(x))\right)}} \tag{4.14}$$

where M is the number of objectives.

In $P1$, the population is kept without any change. The position of hawks in the $P2$ (worse positions) becomes partially opposite. Thus, the hawk becomes opposite in just one dimension. This dimension is selected randomly to ensure that partial useful information of the position is kept. The positions of hawks in the $P2$ are calculated using Equation (4.15). The difference between this equation and Equation (4.13) is that, Equation (4.15) has been applied only on a randomly selected dimension.

$$\check{X}_{id} = \overrightarrow{lb} + \overrightarrow{ub} - \vec{X}_{id} \tag{4.15}$$

where $\vec{X}$ is the position of hawk i in the $P2$ of the population. d is a randomly selected dimension. This aims to increase the diversity of the $P2$ and generate better solutions. The outcomes of this step is a partial OBL population, $\check{P}$. The solutions in P and $\check{P}$ are compared using Pareto-dominance approach to select non-dominated solutions among them. In other words, if X_i in $\check{P}$ dominates Y_j in P, then, the solution Y_j will be replaced by X_i.

Definition 1 (Pareto-dominance) Assume that p and q are any two feasible solutions for the MOP; p is said to dominate q, written as $p \prec q$ if $f_i(p) < f_i(q)$ $\exists j \in 1, 2, \ldots, M$ and $f_i(p) \leq f_i(q)$ $\forall i \in 1, 2, \ldots, M$. where M is the number of objective functions. In other words, solution p is better than q in at least one objective, and p is better than or equal to q in all objective functions (Emmerich & Deutz, 2018).

The proposed two-step population initialization method aims to improve the quality of the initial population by distributing the solutions as evenly as possible in the search space. This leads to effectively improve the performance of the HHMO algorithm. Figure 4.16 shows the main steps of the initial population generator method.

Algorithm 1: Two-step initial population generator method

```
Generation a set of quasi-random numbers using the R-sequence.
Calculate the fitness value of each X in the population using Equation (4.14)
Sort the population based on the fitness values.
Divide the population into two halves
For each X in the second half
    Randomly select a dimension d
    Calculate X̌ using Equation (4.15)
End For
Perform Pareto-dominance relation to eliminate the dominated solutions
Return the initial population of hawks
```

Figure 4.16. Pseudo code of the two-step initial population generator method

The 2S-ENDSHHMO algorithm integrates all improvements discussed in the previous sections to produce a multi-objective SI-based optimization algorithm with better convergence. The pseudo code of the proposed 2S-ENDSHHMO is shown in Figure 4.17.

Algorithm 2: 2S- ENDSHHMO

```
Given the set of objective functions F(X) = (f1(X), f2(X), ... fM(X))
Initialize the population of hawk, X using the proposed two-steps initial
population generator method, within the boundaries of the decision space
(Algorithm 1).
Evaluate each hawk in the population according to objective functions
Divide the search agents into groups based on number of reference points
while ( t <= maximum number of iterations)
    For each reference point group
        Select leaders (α, β and δ) (Algorithm 4).
            i. Calculate the Euclidean distances to a reference point, Zi for each
            hawk, Xi, on all objectives F(X).
            ii. Select first three hawks that have a shortest distance to a reference
            point, Zi, to be α, β and δ.
        Update the population of hawks
```

```
        For each hawk in the group
            Calculate the new position of a hawk based on Equations set (4.6)
        end for
        For each hawks in the group
            Calculate fitness values for all hawks in a group
        end for
        Select the next generation of hawks using non-dominated sorting (Algorithm 3).
    end for
    Evaluate each hawk in the population according to objective functions
    t = t+1
end while
Return the best non-dominated solutions
```

Figure 4.17. Pseudo code of the 2S-ENDSHHMO optimization algorithm

In the proposed 2S-ENDSHHMO algorithm, the optimization process starts by initializing the population of hawks using the two-step population generator method. Then, each hawk, X_i, is evaluated by calculating all objectives, $f_m(X_i)$. The population of hawks is subsequently divided into groups based on the number of reference points, as illustrated in Figure 4.18.

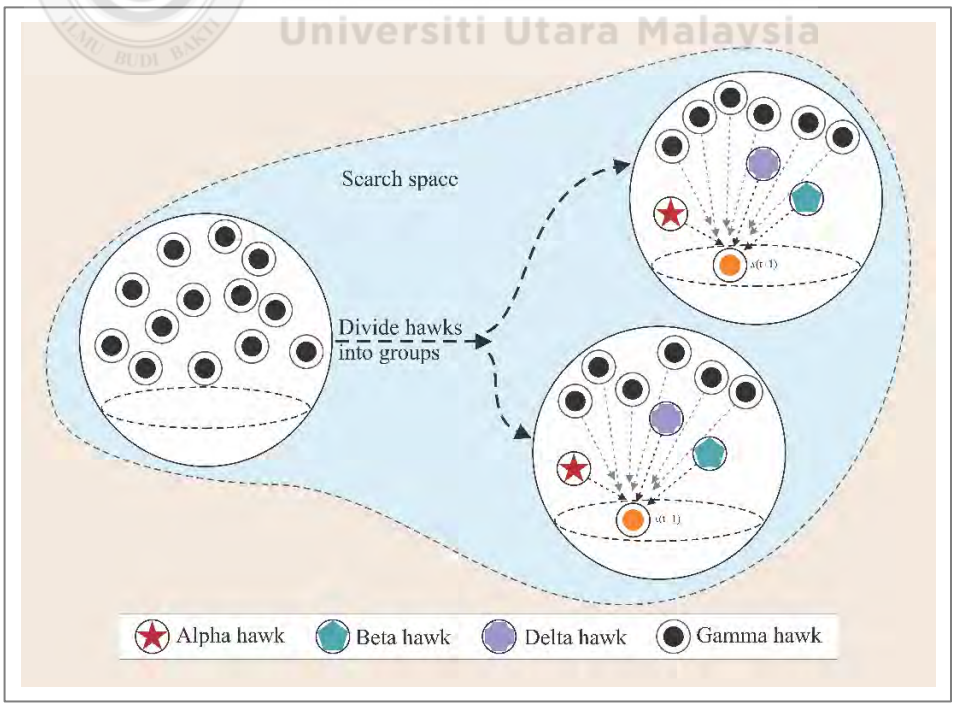


Figure 4.18. Divide the population of hawks into groups based on the number of reference points

In Figure 4.18, each group has a three leaders, X_α, X_β and X_δ and the rest are the X_γ hawks. At each iteration, the population of hawks is updated using the proposed population update strategy, which includes two stages as described in Section 4.4. In the first stage, the position of hawks is updated using the proposed movement strategy. In the second stage, the best hawks are selected to be used in the next generation. Figure 4.19 shows the *selection* procedure.

Algorithm 3: *Selection*

Combine P and O to generate R
Calculate the Euclidean distance, between $f_m(X_i)$ and Z_i
Performs non-dominated sorting to produce the front levels.
    Select individual from the fronts to produce next generation:
    If number of individuals in the current front > N, perform ε-clearing strategy.
        i. Not enough individuals? move to the next front.
        ii. There is more than enough individuals, chooses the ones with the minimum Euclidean distance.
    end if

Figure 4.19. Main steps of *selection* procedure

In the *selection* procedure, the populations P and O are combined to produce R, where the size of R is $2N$. This followed by calculating the Euclidean distance between each solution in the objective space, $f_m(X_i)$, and reference point, Z_i. Then, the best hawks are selected by performing the non-dominated sorting with ε-clearing strategy to produce a population of size N. The new positions are evaluated using the objective function and the new leaders are selected from the new population based on the shortest Euclidean distance to the reference point. Figure 4.20 shows the procedure of *selecting leaders*.

Algorithm 4: *Select_Leaders*

For each hawk in a group
    Calculate the Euclidean distance between a hawk in the objective space, $f_m(X_i)$, and reference point Z, Equation (4.14).
end for

11	Sort the Euclidean distances
12	Find minimum first three values to be X_α, X_β and X_δ, respectively.

Figure 4.20. Main steps of Select_Leaders procedure

These processes are repeated until the loop termination condition is met and, finally, output of the non-dominated solutions is set. The basic procedures of the proposed 2S-ENDSHHMO algorithm are:

a) Initialize a population of hawks.

b) Evaluate each hawk in the population.

c) Select the leaders, α, β and γ from the initial population, based on the smallest Euclidean distance to a reference point.

d) Update the population of hawks.

e) Evaluate each hawk.

4.7 Summary

In this chapter, three main improvements have been proposed and integrated with HHMO to enhance its performance. The population update strategy of the HHMO algorithm is improved by proposing a new update strategy. This strategy consists of calculating a new position of hawks using the proposed formulation based on the tactics used by the Harris's hawks in nature and incorporating the non-dominated sorting and ε–clearing strategy to produce a new generation of hawks. This helped in maintaining the population diversity and convergence toward True Pareto front. The convergence parameter adjustment strategy of the HHMO algorithm is improved by introducing an improved adjustment strategy, which helps in improving the exploration of the algorithm. The initial population generation method is improved by

utilizing a two-step method, which includes two techniques, namely the R-sequence to generate the quasi-random sequence and partial OBL to maintain the initial population diversity. The improved proposed algorithm is called the 2S-ENDSHHMO algorithm, which will be evaluated in the next chapter.

CHAPTER FIVE
PERFORMANCE EVALUATION OF THE TWO-STEP ENHANCED NON-DOMINATED SORTING HARRIS'S HAWK MULTI OBJECTIVE OPTIMIZER

5.1 Introduction

Experimentation is an important step used to evaluate whether a particular algorithm fits with the purpose created for it. This chapter presents how the experimentation phases of the 2S-ENDSHHMO algorithm are organized to compare its results and performance with other state-of-the-art MOO algorithms. To achieve this, this chapter is organized as follows. In Section 5.2, the main phases of the experiment design are presented, while the experimental design of solving test MOPs are presented in Section 5.3. In Section 5.4 the results of integrating the HHMO algorithm with the proposed population update strategy are considered. Section 5.5 discusses the results of integrating the parameter adjustment strategy with the HHMO algorithm. Section 5.6 presents the results of integrating the proposed initial population generation method with the HHMO algorithm. In Section 5.7, the results of evaluating the proposed 2S-ENDSHHMO algorithm are presented and compared with the original HHMO and other MOSI-Based algorithms. Section 5.8 presents the engineering applications. Followed by the experimental design and results of 2S-ENDSHHMO and other state-of-the-art MOO algorithms, in solving well-known real-world engineering MOPs, in Section 5.9. Section 5.10 presents the solution correspond to the extreme points. Section 5.11 presents a method to select a compromise solution from the Pareto set. This is followed by the summary in Section 5.12.

5.2 Experiment Design

In the experiments, a set of 12 test problems and three engineering MOPs, namely, welded beam, four-bar truss and OPF MOPs, presented in Chapter 3, were used to evaluate the performance of each algorithm, with respect to convexity, non-convexity multimodality and non-uniformity. Different performance metrics and statistical measures were used to measure the performance of each algorithm.

The performance of NDSHHMO, EHHMO, introduced in Chapter 4, were compared with the original HHMO. The 2S-HHMO has been compared with five initial population generator methods namely, RNG, which is used in the HHMO, and other QRNG methods, namely, OBL, Sobol and Latin hypercube sampling (LHS) sequences (McKay, Beckman, & Conover, 1979) and Hammersley, which, also, were integrated with HHMO algorithms to initialize the populations of hawks. For simplicity, these methods will be called OBL-HHMO, Sobol-HHMO, LHS-HHMO and Hammersley-HHMO algorithms, in the remaining part of this thesis.

The proposed 2S-ENDSHHMO algorithm was compared with the original HHMO and state-of-the-art MOO algorithms, namely MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II. The first four are state-of-the-art MOSI-based algorithms, which were developed based on the archive approach and widely applied in solving different MOPs (Falehi, 2020; Falehi & Rafiee, 2019; Jiang, Li, & Li, 2019; Mahmoodabadi & Rezaee Babak, 2020; Rahman & Rashid, 2020; Vikram, Ratnam, Lakshmi, Kumar, & Ramakanth, 2018). The others, namely, NSGSA and NSABC, are

MOSI optimization algorithms incorporated with the non-dominated sorting approach. The R-NSGA-II algorithm is an EA developed based on the reference-point approach, which extends the well-known NSGA-II algorithm (Deb et al., 2002a). Figure 5.1 shows the structure of experimental design.

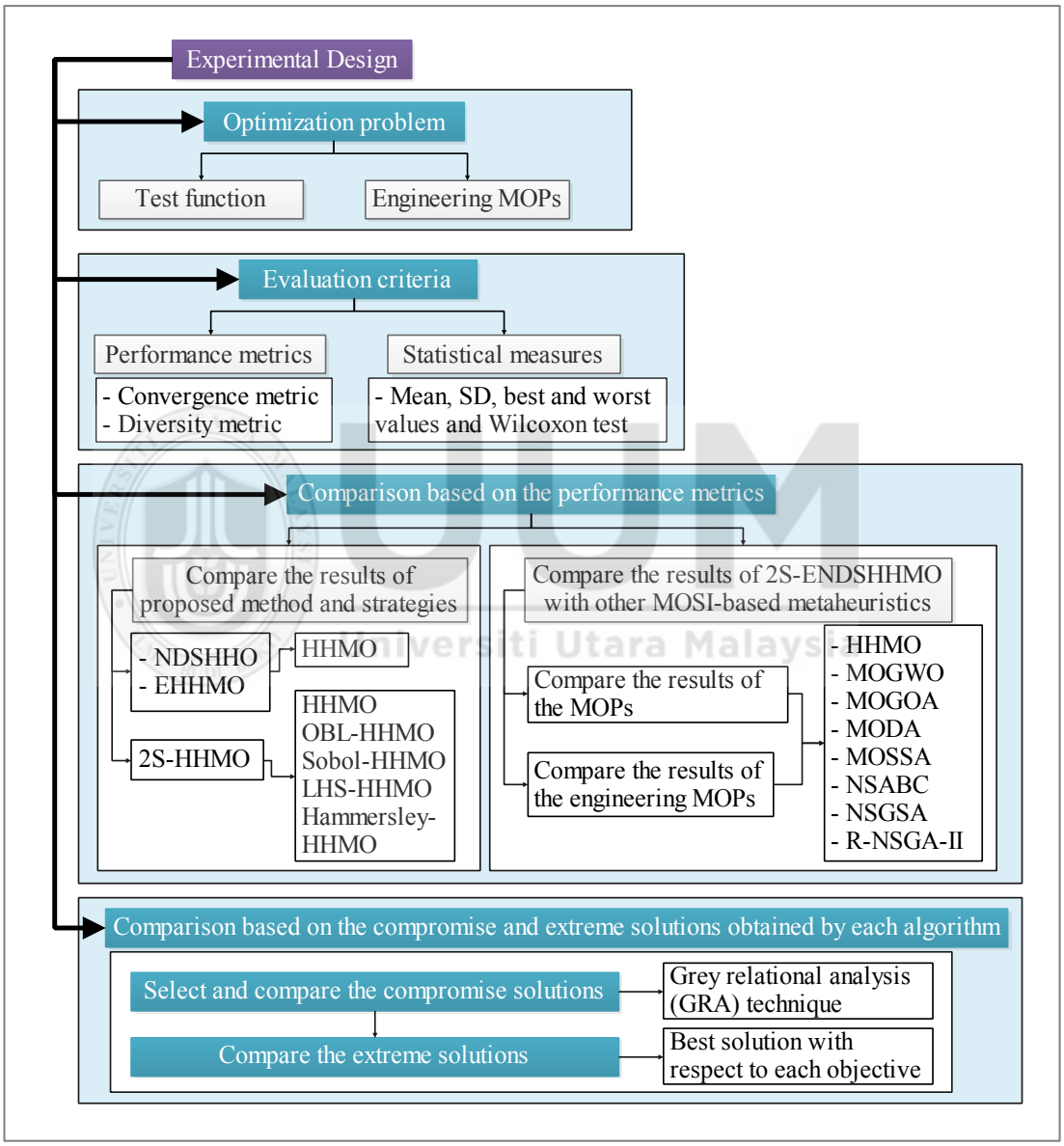


Figure 5.1. Main phases of the experiment

5.3 Experimental design of test problems

For fair comparison the performance experiments of the 2S-ENDSHHMO, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II algorithms were compared under the same conditions. For each algorithm, the maximum number of iterations (*MaxIteration*) was set at 300, with a population size of 100 individuals for each problem. Each algorithm was run independently 10 times (Mellal & Salhi, 2020; Mirjalili, 2016; Mirjalili et al., 2016; Mirjalili et al., 2018) and the algorithm stopped when the number of iterations of each run reached the maximum number of iterations. In solving MOPs using reference-point-based algorithms, different reference points (Deb & Sundar, 2006; Li & Deb, 2016) were used with each test problem, as shown in Table 5.1

Table 5.1
Settings of reference points

Problem	*M*	Reference point
ZDT1	2	(0.8 0.2); (0.2 0.8)
ZDT2		(0.2 0.8); (0.9 0.4)
ZDT3		(0.15 0.4); (0.4 0.0)
ZDT4		(0.1 0.6);(0.5 0.2)
ZDT6		(0.90,0.3); (0.5,0.7)
DTLZ2	3	(0.2,0.2,0.6); (0.8,0.6,1.0)
	5	(0.5, 0.5, 0.5, 0.5, 0.5); (0.2 0.2 0.2 0.2 0.8)
	10	$0.25 \times [1,M]$
DTLZ4	3	(0.2 0.5 0.6); (0.7 0.8 0.5)
DTLZ5		(0.1 0.3 0.5); (0.6 0.7 0.5)
DTLZ6		
DTLZ7		(0.165 0.71 4.678); (0.6 0.6 4.0)

The parameters for all algorithms are set as recommended by their respective authors. The detailed configuration for each algorithm is as shown in Table 5.2.

Table 5.2
Parameter settings of optimization algorithms

Algorithm	Parameter	Value
MOGOA Mirjalili et al. (2018)	cMax	1
	cMin	0.00001
	Archive size	100
MODA Mirjalili (2016)	*s*: separation weight	0.1
	a: alignment weight	0.1
	c: cohesion weight	0.7
	f: food factor	1
	e: enemy factor	1
	w: inertia weight	0.9
MOSSA Mirjalili et al. (2017)	Archive size	100
	Archive size	100
NSABC Kishor et al. (2016)	Archive size	100
	limit	(NP *D /2)
	Percent of elitism	0.5
NSGSA Zellagui et al. (2017)	Coefficient of search interval	2.5
	Sign mutation probability	0.9
	Uniform mutation probability	0.01
	Reordering mutation probability	0.4
	Initial value of inertial coefficient (W_0)	0.9
	Final value of inertial coefficient (W_1)	0.5
HHMO DeBruyne and Kaur (2016)	Archive size	100
	Radius	0.2
R-NSGA-II Deb et al. (2002a)	Crossover factor	0.9
	Mutation factor	1/N

5.4 Results of Integrating the Proposed Population Update Strategy with HHMO Algorithm

This section presents the results of the NDSHHMO algorithm. In this version the proposed population update strategy is integrated with the HHMO algorithm to produce the NDSHHMO algorithm. The test problems are used to evaluate the HHMO and NDSHHMO algorithms. The mean, SD, best and worst values of R-IGD and R-

HV metrics are calculated to be used in the comparison. Table 5.3 shows the results of HHMO and NDSHHMO algorithms.

Table 5.3
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions, obtained by using HHMO and NDSHHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	HHMO	R-IGD	2.2409E-01	2.0820E-01	3.7968E-03	4.3617E-01
		R-HV	3.3826E+00	6.0412E-01	4.0466E+00	2.7423E+00
	NDSHHMO	R-IGD	**3.7907E-03**	4.6473E-05	3.7132E-03	3.8689E-03
		R-HV	**4.0476E+00**	2.6139E-03	4.0531E+00	4.0461E+00
ZDT2	HHMO	R-IGD	1.3763E-01	1.3700E-01	1.4112E-02	4.8704E-01
		R-HV	3.3518E+00	3.8325E-01	3.8114E+00	2.5300E+00
	NDSHHMO	R-IGD	**1.4150E-02**	4.7178E-04	1.3336E-02	1.4839E-02
		R-HV	**3.8102E+00**	3.6586E-03	3.8136E+00	3.8005E+00
ZDT3	HHMO	R-IGD	1.7811E-01	1.1515E-01	1.5740E-02	2.9921E-01
		R-HV	3.3721E+00	3.5069E-01	3.9525E+00	2.9929E+00
	NDSHHMO	R-IGD	**1.5436E-02**	3.5459E-04	1.4570E-02	1.5775E-02
		R-HV	**3.9505E+00**	3.6598E-03	3.9529E+00	3.9402E+00
ZDT4	HHMO	R-IGD	7.8105E-02	9.3784E-02	2.9646E-03	2.8425E-01
		R-HV	3.8259E+00	3.5263E-01	4.1340E+00	3.1398E+00
	NDSHHMO	R-IGD	**2.7136E-03**	3.3463E-05	2.6358E-03	2.7618E-03
		R-HV	**4.1342E+00**	6.5459E-04	4.1358E+00	4.1336E+00
ZDT6	HHMO	R-IGD	4.2760E-01	4.8118E-01	4.5574E-03	1.0499E+00
		R-HV	3.2860E+00	1.2621E+00	4.4118E+00	1.6868E+00
	NDSHHMO	R-IGD	**6.3356E-02**	1.7179E-01	4.4646E-03	5.5084E-01
		R-HV	**4.2348E+00**	4.8357E-01	4.4121E+00	2.8741E+00
DTLZ2 (3)	HHMO	R-IGD	**1.5596E-01**	2.8048E-02	1.2197E-01	2.1355E-01
		R-HV	**8.5351E+00**	2.3963E-01	8.8669E+00	8.1150E+00
	NDSHHMO	R-IGD	3.0220E-01	1.9630E-01	1.6491E-01	5.8838E-01
		R-HV	7.5001E+00	1.2408E+00	8.4029E+00	5.6880E+00
DTLZ2 (5)	HHMO	R-IGD	1.8237E-01	7.2567E-02	1.1081E-01	3.7323E-01
		R-HV	3.0502E+01	2.3642E+00	3.2860E+01	2.4449E+01
	NDSHHMO	R-IGD	**1.1351E-01**	1.8309E-03	1.1115E-01	1.1745E-01
		R-HV	**3.2511E+01**	1.1083E-01	3.2690E+01	3.2252E+01
DTLZ2 (10)	HHMO	R-IGD	6.9929E-01	1.0747E-01	6.0993E-01	9.6582E-01
		R-HV	7.2149E+02	1.4911E+02	8.8498E+02	3.8534E+02
	NDSHHMO	R-IGD	**5.7162E-01**	1.2627E-03	5.7011E-01	5.7339E-01
		R-HV	**1.0402E+03**	2.7635E+00	1.0449E+03	1.0366E+03
DTLZ4	HHMO	R-IGD	5.7004E-01	3.3366E-01	7.9573E-02	1.1994E+00
		R-HV	4.2076E+00	1.5124E+00	6.8459E+00	1.7029E+00
	NDSHHMO	R-IGD	**8.0168E-02**	2.5488E-02	5.3741E-02	1.1646E-01
		R-HV	**7.0354E+00**	2.0318E-01	7.2861E+00	6.7035E+00
DTLZ5	HHMO	R-IGD	3.3389E-01	9.5281E-02	2.3352E-01	4.8342E-01
		R-HV	4.1879E+00	5.0189E-01	4.7658E+00	3.4364E+00
	NDSHHMO	R-IGD	**1.7126E-01**	3.3195E-03	1.6824E-01	1.7992E-01
		R-HV	**5.3365E+00**	4.9218E-02	5.4000E+00	5.2357E+00
DTLZ6	HHMO	R-IGD	2.2187E-01	1.9510E-03	2.1707E-01	2.2336E-01
		R-HV	4.8669E+00	1.7625E-02	4.9089E+00	4.8524E+00
	NDSHHMO	R-IGD	**2.0957E-01**	1.3234E-02	1.8424E-01	2.2259E-01
		R-HV	**4.9698E+00**	1.1173E-01	5.1900E+00	4.8599E+00
DTLZ7	HHMO	R-IGD	4.2128E+00	4.1361E-01	3.7130E+00	4.9373E+00

NDSHHMO	R-HV	4.7921E+00	2.5149E+00	8.2216E+00	1.3925E+00
	R-IGD	**3.7949E+00**	2.1625E-01	3.7163E+00	4.4103E+00
	R-HV	**7.6049E+00**	1.5372E+00	8.1700E+00	3.2312E+00

The results in Table 5.3 show that, in terms of convergence toward the true Pareto front and diversity of non-dominated solutions, the NDSHHMO algorithm has the lowest mean R-IGD and highest mean R-HV values in solving 11 out of 12 problems. The HHMO has produced the lowest mean R-IGD and highest mean R-HV values for four problems. Thus, NDSHHMO has managed to achieve lowest mean R-IGD and highest mean R-HV values in solving 91.7% of the problems. while the HHMO algorithm achieved highest diversity in solving 8.3 % of the problems.

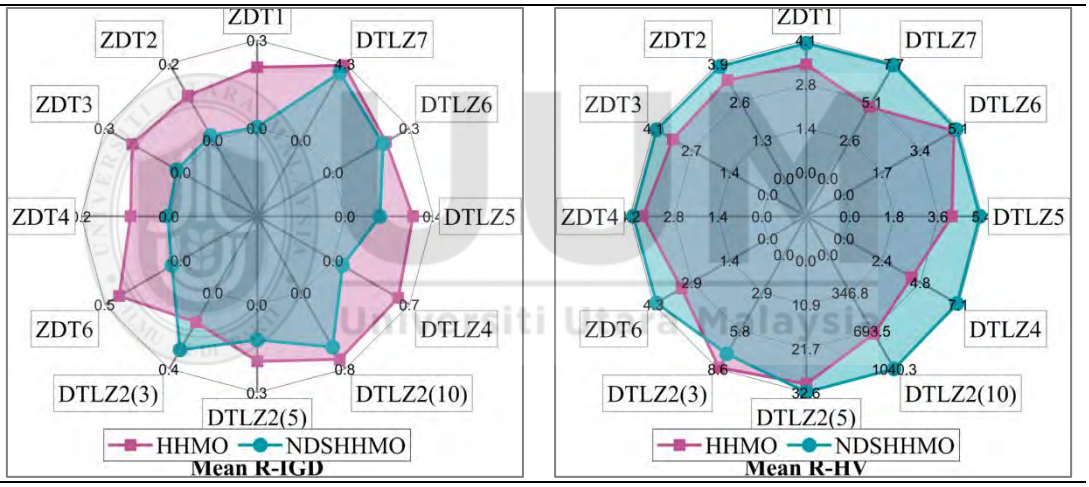


Figure 5.2. Summary of mean R-IGD and R-HV values of obtained solutions, for both HHMO and NDSHHMO algorithms

In Figure 5.2, the square and circle shapes represent the mean R-metric obtained by HHO and EHHMO, respectively. In general terms, the results obtained for the experiments carried out (refer to Table 5.3), indicate that the use of the proposed population update strategy can significantly improve the distribution of the solutions in the objective space, while preserving other characteristics of the algorithm, namely the convergence ability toward the True Pareto front and diversity of obtained

solutions. The results emphasize that the population update strategy has the advantage of solving problems with multiple local fronts, such as the ZDT4 problem, and non-uniform front, such as, ZDT3 and ZDT6 problems. Furthermore, the results imply the ability of the NDSHHMO algorithm in solving many-objective optimization problems, such as DTLZ2 with five and 10 objectives. However, the NDSHHMO algorithm can be further improved by improving the exploration ability of the algorithm. This will be discussed in the next section.

5.5 Results of Integrating the Proposed Parameter Adjustment Strategy with HHMO Algorithm

This section presents the experimental results of the EHHMO algorithm, which involves integration between the proposed parameter adjustment strategy and the HHMO algorithm to control the value of the convergence parameter *a*. The MOPs are used to evaluate the performance of the EHHMO algorithm and to verify the effectiveness of the proposed parameter adjustment strategy. To measure the performance of EHHMO, the mean, SD, best and worst values of R-IGD and R-HV metrics are calculated and compared with the results of HHMO, as shown in Table 5.4.

Table 5.4
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions, obtained by using HHMO and EHHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	HHMO	R-IGD	1.7452E-01	2.0812E-01	3.7968E-03	4.1923E-01
		R-HV	3.5292E+00	6.0766E-01	4.0553E+00	2.7423E+00
	EHHMO	R-IGD	**5.1506E-02**	1.2814E-01	3.7798E-03	4.1520E-01
		R-HV	**3.9048E+00**	3.6633E-01	4.0468E+00	2.8663E+00
ZDT2	HHMO	R-IGD	**7.3419E-02**	6.8278E-02	1.4112E-02	1.5239E-01
		R-HV	**3.5628E+00**	2.8573E-01	3.8114E+00	3.2328E+00
	EHHMO	R-IGD	8.4402E-02	7.1699E-02	1.3586E-02	1.5239E-01
		R-HV	3.5167E+00	2.9941E-01	3.8129E+00	3.2328E+00
ZDT3	HHMO	R-IGD	1.9808E-01	1.2774E-01	1.3844E-02	2.9921E-01

ZDT4	EHHMO	R-HV	3.3364E+00	4.2821E-01	3.9525E+00	2.9929E+00
		R-IGD	**4.9940E-02**	9.8103E-02	1.4240E-02	3.2789E-01
		R-HV	**3.8301E+00**	3.2199E-01	3.9520E+00	2.9224E+00
	HHMO	R-IGD	6.4837E-02	9.3394E-02	2.7279E-03	2.7911E-01
		R-HV	**3.8912E+00**	3.4153E-01	4.1350E+00	3.2041E+00
	EHHMO	R-IGD	**6.3138E-02**	5.9699E-02	2.7249E-03	1.5940E-01
		R-HV	3.8762E+00	2.5768E-01	4.1345E+00	3.5010E+00
ZDT6	HHMO	R-IGD	3.5845E-01	4.5463E-01	4.5574E-03	1.0481E+00
		R-HV	3.4636E+00	1.1987E+00	4.4118E+00	1.6900E+00
	EHHMO	R-IGD	**2.9750E-01**	4.6272E-01	4.4870E-03	1.0508E+00
		R-HV	**3.6352E+00**	1.2144E+00	4.4122E+00	1.6850E+00
DTLZ2 (3)	HHMO	R-IGD	**1.5363E-01**	2.7041E-02	1.2197E-01	2.1466E-01
		R-HV	**8.5393E+00**	2.4574E-01	8.8531E+00	8.0152E+00
	EHHMO	R-IGD	2.1878E-01	1.3667E-01	1.5610E-01	6.0430E-01
		R-HV	8.0849E+00	8.7308E-01	8.5050E+00	5.6207E+00
DTLZ2 (5)	HHMO	R-IGD	2.5236E-01	2.8999E-01	1.1081E-01	1.0739E+00
		R-HV	2.9048E+01	6.8342E+00	3.2860E+01	9.8213E+00
	EHHMO	R-IGD	**2.4552E-01**	3.2791E-01	1.1744E-01	1.1774E+00
		R-HV	**2.9476E+01**	7.4170E+00	3.2446E+01	8.4729E+00
DTLZ2 (10)	HHMO	R-IGD	6.9927E-01	1.0747E-01	6.0993E-01	9.6582E-01
		R-HV	7.2157E+02	1.4914E+02	8.8498E+02	3.8534E+02
	EHHMO	R-IGD	**6.9314E-01**	1.1821E-01	5.7317E-01	9.6495E-01
		R-HV	**7.5245E+02**	1.8779E+02	9.7198E+02	3.8310E+02
DTLZ4	HHMO	R-IGD	5.1482E-01	3.2072E-01	7.9573E-02	1.1994E+00
		R-HV	4.4052E+00	1.4413E+00	6.8459E+00	1.7029E+00
	EHHMO	R-IGD	**2.4016E-01**	1.3019E-01	9.8085E-02	4.4414E-01
		R-HV	**5.7653E+00**	8.2673E-01	6.7687E+00	4.5332E+00
DTLZ5	HHMO	R-IGD	**2.2587E-01**	8.8818E-02	1.7362E-01	4.5742E-01
		R-HV	**4.9333E+00**	5.9407E-01	5.3636E+00	3.4859E+00
	EHHMO	R-IGD	2.4192E-01	1.3106E-01	1.7048E-01	4.8727E-01
		R-HV	4.8883E+00	8.0739E-01	5.4118E+00	3.3568E+00
DTLZ6	HHMO	R-IGD	2.2187E-01	1.9510E-03	2.1707E-01	2.2336E-01
		R-HV	4.8669E+00	1.7625E-02	4.9089E+00	4.8524E+00
	EHHMO	R-IGD	**2.1932E-01**	4.2563E-03	2.1247E-01	2.2295E-01
		R-HV	**4.8880E+00**	3.4597E-02	4.9448E+00	4.8563E+00
DTLZ7	HHMO	R-IGD	**4.2089E+00**	4.1834E-01	3.7130E+00	4.9373E+00
		R-HV	**4.8098E+00**	2.5388E+00	8.2216E+00	1.3925E+00
	EHHMO	R-IGD	4.2722E+00	5.0057E-01	3.7129E+00	4.9425E+00
		R-HV	4.6123E+00	2.8789E+00	8.2197E+00	1.3746E+00

Results in Table 5.4 proved that, in terms of the convergence toward the true Pareto front, the EHHMO algorithm has the lowest mean R-IGD values in solving nine out of 12 problems. HHMO has produced the lowest values for three problems. Thus, EHHMO has managed to achieve lower mean R-IGD value in solving 75% of the problems.

In terms of the diversity of obtained non-dominated solutions, highest values of mean R-HV have been obtained by the EHHMO algorithm for eight out of 12 problems while HHMO has only obtained highest values for four problems. Thus, for diversity test, EHHMO algorithm has higher mean R-HV values in solving 66% of the problems, while the HHMO algorithm achieved higher diversity in solving 33% of the problems. Figure 5.3 shows a comparison between the mean R-IGD and R-HV values obtained by EHHMO and HHMO algorithms. The square and circle shapes represent the mean R-metric obtained by HHO and EHHMO, respectively. Based on the mean R-IGD values, the proposed parameter adjustment strategy has improved the performance of the HHMO algorithm, in terms of convergence toward True Pareto front in solving 75% of the problems. Based on mean R-HV values, the EHHMO algorithm is managed to produce non-dominated solutions with higher diversity in solving 66% of the problems. This is due to the proposed parameter adjustment strategy improves the ability of HHMO algorithm in exploring and exploiting the search space, which leads to an improvement in convergence toward the True Pareto front and diversity of the obtained solutions.

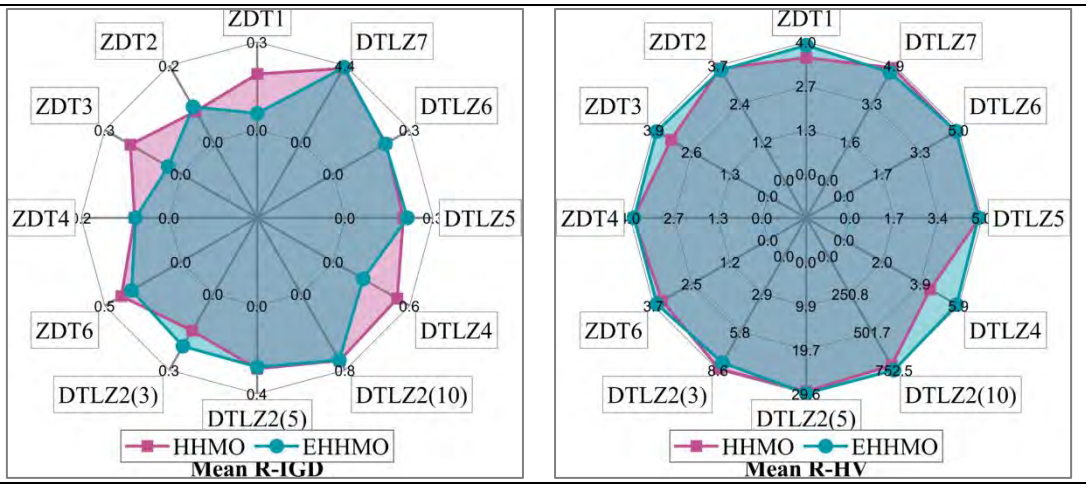


Figure 5.3. Summary of mean R-IGD and R-HV values of obtained solutions, for both HHMO and EHHMO algorithms

5.6 Results of Integrating Proposed Two-step Initial Population Generator Method with HHMO Algorithm

The proposed two-step initial population generator method is integrated with the HHMO algorithm to produce the 2S-HHMO algorithm. To validate the effectiveness of the proposed algorithm, the 2S-HHMO algorithm is used to solve the test problems. The obtained results are compared with the original HHMO, OBL-HHMO, Sobol-HHMO, LHS-HHMO and Hammersley-HHMO algorithms.

5.6.1 Comparing the Two-step Initial Population Generator and Random Number Generator Methods

This section presents the result of comparing the two-step initial population generator and traditional RNG methods in terms of convergence and diversity of the solutions in the search space. Additionally, it presents the results of comparing the two-step initial population generator with RNG and other methods in solving MOPs. The proposed

two-step initial population generator method has been compared against other techniques, in solving MOPs.

Table 5.5 shows the results of mean, SD, best and worst values of the R-IGD and R-HV metrics. In terms of the convergence toward the true Pareto front, the 2S-HHMO algorithm has the lowest mean R-IGD values in solving eight out of 12 problems. HHMO has produced the lowest values for four problems. Thus, 2S-HHMO has managed to achieve lowest mean R-IGD value in solving 66% of the problems. In terms of the diversity of obtained non-dominated solutions, highest values of mean R-HV have been obtained by the 2S-HHMO algorithm for eight out of 12 problems while HHMO has only obtained highest values for four problems. Thus, for diversity test, 2S-HHMO algorithm has highest mean R-HV values in solving 66% of the problems, while the HHMO algorithm achieved highest diversity in solving 33% of the problems.

Table 5.5
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions obtained by using HHMO and 2S-HHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	HHMO	R-IGD	1.6962E-01	2.1162E-01	3.3871E-03	4.1902E-01
		R-HV	3.5708E+00	6.0937E-01	4.0553E+00	2.8555E+00
	2S-HHMO	R-IGD	**7.2087E-02**	1.2672E-01	4.3569E-03	4.1809E-01
		R-HV	**3.7976E+00**	3.7624E-01	4.0482E+00	2.8568E+00
ZDT2	HHMO	R-IGD	9.0407E-02	6.6643E-02	1.4001E-02	1.5239E-01
		R-HV	3.4833E+00	2.8253E-01	3.8114E+00	3.2328E+00
	2S-HHMO	R-IGD	**7.4865E-02**	6.4853E-02	1.4444E-02	1.5239E-01
		R-HV	**3.5450E+00**	2.7892E-01	3.8101E+00	3.2329E+00
ZDT3	HHMO	R-IGD	1.1269E-01	1.1974E-01	1.4769E-02	2.9921E-01
		R-HV	3.5773E+00	3.7627E-01	3.9525E+00	2.9929E+00
	2S-HHMO	R-IGD	**6.9168E-02**	1.0577E-01	1.3770E-02	2.9476E-01
		R-HV	**3.7632E+00**	3.4370E-01	3.9524E+00	3.0708E+00
ZDT4	HHMO	R-IGD	7.8035E-02	9.3771E-02	2.7279E-03	2.8425E-01
		R-HV	3.8258E+00	3.5249E-01	4.1341E+00	3.1398E+00
	2S-HHMO	R-IGD	**2.6615E-02**	4.8944E-02	2.3182E-03	1.4011E-01
		R-HV	**4.0331E+00**	2.1961E-01	4.1557E+00	3.5371E+00
ZDT6	HHMO	R-IGD	4.2760E-01	4.8118E-01	4.5574E-03	1.0499E+00
		R-HV	3.2860E+00	1.2621E+00	4.4118E+00	1.6868E+00
	2S-HHMO	R-IGD	**4.2721E-01**	4.8087E-01	4.5270E-03	1.0481E+00
		R-HV	**3.2869E+00**	1.2614E+00	4.4117E+00	1.6900E+00

DTLZ2 (3)	HHMO	R-IGD	**1.6218E-01**	3.2397E-02	1.2197E-01	2.1363E-01
		R-HV	**8.4735E+00**	2.8706E-01	8.8669E+00	8.0957E+00
	2S-HHMO	R-IGD	2.1125E-01	1.7138E-01	1.4206E-01	6.9711E-01
		R-HV	8.1508E+00	1.0856E+00	8.6657E+00	5.0891E+00
DTLZ2 (5)	HHMO	R-IGD	1.8453E-01	7.6561E-02	1.1081E-01	3.8688E-01
		R-HV	3.0359E+01	2.6035E+00	3.2860E+01	2.3636E+01
	2S-HHMO	R-IGD	**1.2495E-01**	1.2722E-02	1.0524E-01	1.4682E-01
		R-HV	**3.2388E+01**	6.7974E-01	3.3867E+01	3.1514E+01
DTLZ2 (10)	HHMO	R-IGD	6.9927E-01	1.0747E-01	6.0993E-01	9.6582E-01
		R-HV	7.2157E+02	1.4914E+02	8.8498E+02	3.8534E+02
	2S-HHMO	R-IGD	**6.6107E-01**	5.6263E-02	5.9793E-01	7.4668E-01
		R-HV	**7.9489E+02**	1.1116E+02	9.3754E+02	6.0796E+02
DTLZ4	HHMO	R-IGD	4.0738E-01	2.4392E-01	7.9573E-02	7.4640E-01
		R-HV	4.9124E+00	1.2691E+00	6.8459E+00	3.3358E+00
	2S-HHMO	R-IGD	**3.0510E-01**	1.0056E-01	2.0025E-01	4.5295E-01
		R-HV	**5.3792E+00**	6.0560E-01	6.0512E+00	4.4726E+00
DTLZ5	HHMO	R-IGD	**2.0055E-01**	4.4899E-02	1.7362E-01	3.1669E-01
		R-HV	**5.0877E+00**	3.5670E-01	5.3636E+00	4.2632E+00
	2S-HHMO	R-IGD	2.1152E-01	1.0613E-01	1.7057E-01	5.1307E-01
		R-HV	5.0757E+00	6.3882E-01	5.3988E+00	3.2763E+00
DTLZ6	HHMO	R-IGD	**2.2187E-01**	1.9510E-03	2.1707E-01	2.2336E-01
		R-HV	**4.8669E+00**	1.7625E-02	4.9089E+00	4.8524E+00
	2S-HHMO	R-IGD	2.6441E-01	1.3624E-01	2.2080E-01	6.5216E-01
		R-HV	4.6582E+00	6.7807E-01	4.8771E+00	2.7284E+00
DTLZ7	HHMO	R-IGD	**4.2089E+00**	4.1834E-01	3.7130E+00	4.9373E+00
		R-HV	**4.8098E+00**	2.5388E+00	8.2216E+00	1.3925E+00
	2S-HHMO	R-IGD	4.3358E+00	3.0774E-01	3.7504E+00	4.6299E+00
		R-HV	3.8626E+00	1.9232E+00	7.7278E+00	2.4360E+00

5.6.2 Comparing the Two-Step Initial Population Generator and OBL Methods

This section presents the results obtained by the 2S-HHMO and OBL-HHMO algorithms. Table 5.6 shows the results of mean, SD, best and worst values of the R-IGD and R-HV metrics. The convergence toward the true Pareto front and diversity of non-dominated solutions, the 2S-HHMO algorithm has the lowest mean R-IGD and highest mean R-HV values in solving eight out of 12 problems. OBL-HHMO has produced the lowest mean R-IGD and highest mean R-HV values for four problems. Thus, 2S-HHMO has managed to achieve lowest mean R-IGD and highest mean R-HV values in solving 75% of the problems. while the OBL-HHMO algorithm achieved highest diversity in solving 25% of the problems.

Table 5.6

Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions obtained by using OBL-HHMO and 2S-HHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	OBL-HHMO	R-IGD	2.1102E-01	2.4384E-01	4.8293E-03	6.7192E-01
		R-HV	3.4310E+00	6.8307E-01	4.0428E+00	2.1663E+00
	2S-HHMO	R-IGD	**1.1230E-01**	1.7003E-01	4.2603E-03	4.2355E-01
		R-HV	**3.6949E+00**	4.9801E-01	4.0482E+00	2.8423E+00
ZDT2	OBL-HHMO	R-IGD	1.2133E-01	5.7144E-02	1.3707E-02	1.5239E-01
		R-HV	3.3585E+00	2.4019E-01	3.8127E+00	3.2328E+00
	2S-HHMO	R-IGD	**9.6648E-02**	1.5028E-01	1.4444E-02	4.1834E-01
		R-HV	**3.5507E+00**	4.2434E-01	3.8103E+00	2.7116E+00
ZDT3	OBL-HHMO	R-IGD	2.4370E-01	2.7992E-01	1.5537E-02	7.5374E-01
		R-HV	3.2703E+00	7.5165E-01	3.9514E+00	1.9797E+00
	2S-HHMO	R-IGD	**6.9398E-02**	1.0568E-01	1.4646E-02	2.9476E-01
		R-HV	**3.7624E+00**	3.4377E-01	3.9524E+00	3.0708E+00
ZDT4	OBL-HHMO	R-IGD	5.6352E-02	5.9711E-02	2.6610E-03	1.4504E-01
		R-HV	3.9047E+00	2.5913E-01	4.1352E+00	3.5313E+00
	2S-HHMO	R-IGD	**2.2318E-02**	3.6836E-02	2.3182E-03	9.9553E-02
		R-HV	**4.0452E+00**	1.7485E-01	4.1557E+00	3.6881E+00
ZDT6	OBL-HHMO	R-IGD	**2.2447E-01**	2.8390E-01	4.5302E-03	5.5563E-01
		R-HV	**3.7932E+00**	7.9816E-01	4.4116E+00	2.8626E+00
	2S-HHMO	R-IGD	4.2721E-01	4.8087E-01	4.5270E-03	1.0481E+00
		R-HV	3.2869E+00	1.2614E+00	4.4117E+00	1.6900E+00
DTLZ2 (3)	OBL-HHMO	R-IGD	1.9768E-01	3.5392E-02	1.4868E-01	2.6980E-01
		R-HV	8.1673E+00	2.8015E-01	8.5637E+00	7.6858E+00
	2S-HHMO	R-IGD	**1.5638E-01**	1.2850E-02	1.4206E-01	1.8293E-01
		R-HV	**8.4890E+00**	1.1925E-01	8.6657E+00	8.2896E+00
DTLZ2 (5)	OBL-HHMO	R-IGD	2.3493E-01	2.9443E-01	1.1003E-01	1.0711E+00
		R-HV	2.9610E+01	6.9905E+00	3.2798E+01	9.8223E+00
	2S-HHMO	R-IGD	**1.2449E-01**	1.2984E-02	1.0524E-01	1.4682E-01
		R-HV	**3.2406E+01**	6.8418E-01	3.3867E+01	3.1514E+01
DTLZ2 (10)	OBL-HHMO	R-IGD	7.0322E-01	6.8352E-02	6.0501E-01	8.2131E-01
		R-HV	7.2338E+02	1.1957E+02	9.4341E+02	5.3300E+02
	2S-HHMO	R-IGD	**6.6107E-01**	5.6263E-02	5.9793E-01	7.4668E-01
		R-HV	**7.9489E+02**	1.1116E+02	9.3754E+02	6.0796E+02
DTLZ4	OBL-HHMO	R-IGD	3.6054E-01	1.4981E-01	1.4212E-01	5.7983E-01
		R-HV	5.0600E+00	8.3535E-01	6.3145E+00	3.9093E+00
	2S-HHMO	R-IGD	**3.2770E-01**	1.0906E-01	1.8121E-01	4.6086E-01
		R-HV	**5.1876E+00**	6.9183E-01	6.1569E+00	4.3831E+00
DTLZ5	OBL-HHMO	R-IGD	2.4369E-01	1.0833E-01	1.6907E-01	4.5971E-01
		R-HV	4.8275E+00	6.8385E-01	5.4041E+00	3.5364E+00
	2S-HHMO	R-IGD	**2.3808E-01**	1.2092E-01	1.7057E-01	4.9179E-01
		R-HV	**4.8901E+00**	7.5974E-01	5.3988E+00	3.3871E+00
DTLZ6	OBL-HHMO	R-IGD	**2.2218E-01**	1.0475E-03	2.2081E-01	2.2345E-01
		R-HV	**4.8648E+00**	9.5497E-03	4.8770E+00	4.8516E+00
	2S-HHMO	R-IGD	2.6441E-01	1.3624E-01	2.2080E-01	6.5216E-01
		R-HV	4.6582E+00	6.7807E-01	4.8771E+00	2.7284E+00
DTLZ7	OBL-HHMO	R-IGD	**4.2358E+00**	4.5405E-01	3.7203E+00	4.9392E+00
		R-HV	**4.7384E+00**	2.6822E+00	8.0469E+00	1.3823E+00
	2S-HHMO	R-IGD	4.3431E+00	3.0969E-01	3.7504E+00	4.6299E+00
		R-HV	3.8050E+00	1.9393E+00	7.7278E+00	2.4360E+00

5.6.3 Comparing the Two-Step Initial Population Generator Method and Sobol Sequence

This section presents the results obtained by the 2S-HHMO and Sobol-HHMO algorithms. Table 5.7 shows the results of mean, SD, best and worst values of the R-IGD and R-HV metrics.

Table 5.7
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions obtained by using Sobol-HHMO and 2S-HHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	Sobol-HHMO	R-IGD	1.4572E-01	**1.8949E-01**	**3.7948E-03**	**4.1589E-01**
		R-HV	3.5915E+00	5.6091E-01	**4.0505E+00**	**2.7370E+00**
	2S-HHMO	R-IGD	**1.1879E-01**	2.0445E-01	4.3569E-03	5.5964E-01
		R-HV	**3.6951E+00**	**5.8596E-01**	4.0482E+00	2.4669E+00
ZDT2	Sobol-HHMO	R-IGD	**8.3027E-02**	**7.2591E-02**	**1.3352E-02**	**1.5240E-01**
		R-HV	**3.5219E+00**	3.0321E-01	**3.8140E+00**	**3.2328E+00**
	2S-HHMO	R-IGD	1.0956E-01	1.3401E-01	1.4444E-02	4.2355E-01
		R-HV	3.4631E+00	**4.3208E-01**	3.8104E+00	2.5587E+00
ZDT3	Sobol-HHMO	R-IGD	2.1466E-01	1.8647E-01	**1.3288E-02**	5.2920E-01
		R-HV	3.3213E+00	**5.7013E-01**	3.9520E+00	2.4530E+00
	2S-HHMO	R-IGD	**9.3185E-02**	**1.1794E-01**	1.4646E-02	**2.9476E-01**
		R-HV	**3.6831E+00**	3.8334E-01	**3.9524E+00**	**3.0708E+00**
ZDT4	Sobol-HHMO	R-IGD	7.3961E-02	1.1122E-01	2.7011E-03	3.2243E-01
		R-HV	3.8678E+00	**3.8759E-01**	4.1343E+00	3.0594E+00
	2S-HHMO	R-IGD	**2.7541E-02**	**4.8722E-02**	**2.3182E-03**	**1.4011E-01**
		R-HV	**4.0247E+00**	2.1775E-01	**4.1557E+00**	**3.5371E+00**
ZDT6	Sobol-HHMO	R-IGD	4.2729E-01	4.8090E-01	4.5386E-03	**1.0481E+00**
		R-HV	3.2867E+00	**1.2616E+00**	**4.4118E+00**	**1.6900E+00**
	2S-HHMO	R-IGD	**4.2721E-01**	**4.8087E-01**	**4.5270E-03**	1.0481E+00
		R-HV	**3.2869E+00**	1.2614E+00	4.4117E+00	1.6900E+00
DTLZ2 (3)	Sobol-HHMO	R-IGD	1.8411E-01	2.9148E-02	**1.3409E-01**	2.2175E-01
		R-HV	8.2505E+00	**2.4992E-01**	**8.7037E+00**	**7.9118E+00**
	2S-HHMO	R-IGD	**1.5942E-01**	**2.2306E-02**	1.4206E-01	**2.1743E-01**
		R-HV	**8.4722E+00**	2.2336E-01	8.6657E+00	7.9036E+00
DTLZ2 (5)	Sobol-HHMO	R-IGD	3.2212E-01	2.5073E-01	1.2758E-01	1.0132E+00
		R-HV	2.6577E+01	**5.9673E+00**	3.2286E+01	1.0658E+01
	2S-HHMO	R-IGD	**1.2449E-01**	**1.2984E-02**	**1.0524E-01**	**1.4682E-01**
		R-HV	**3.2412E+01**	6.8429E-01	**3.3867E+01**	**3.1514E+01**
DTLZ2 (10)	Sobol-HHMO	R-IGD	6.7440E-01	5.7426E-02	**5.8231E-01**	7.6012E-01
		R-HV	7.4518E+02	**1.1588E+02**	9.2994E+02	5.8450E+02
	2S-HHMO	R-IGD	**6.6107E-01**	**5.6263E-02**	5.9793E-01	**7.4668E-01**
		R-HV	**7.9489E+02**	1.1116E+02	**9.3754E+02**	**6.0796E+02**
DTLZ4	Sobol-HHMO	R-IGD	**2.3128E-01**	**9.3218E-02**	**1.3096E-01**	**3.9010E-01**
		R-HV	**5.7741E+00**	5.9071E-01	**6.3963E+00**	**4.7182E+00**
	2S-HHMO	R-IGD	3.4114E-01	1.8432E-01	1.8121E-01	7.1312E-01
		R-HV	5.2054E+00	**1.0080E+00**	6.1569E+00	3.2910E+00
DTLZ5	Sobol-HHMO	R-IGD	2.6670E-01	1.4099E-01	**1.6959E-01**	5.1751E-01
		R-HV	4.7127E+00	**8.4487E-01**	**5.4245E+00**	3.3548E+00

DTLZ6	2S-HHMO	R-IGD	**2.0042E-01**	**5.0364E-02**	1.7057E-01	**3.3180E-01**
		R-HV	**5.0880E+00**	4.1448E-01	5.3988E+00	**4.0863E+00**
	Sobol-HHMO	R-IGD	**2.6242E-01**	**1.2840E-01**	2.2085E-01	**6.2784E-01**
		R-HV	**4.6663E+00**	6.3733E-01	4.8766E+00	**2.8526E+00**
	2S-HHMO	R-IGD	2.6441E-01	1.3624E-01	**2.2080E-01**	6.5216E-01
		R-HV	4.6582E+00	**6.7807E-01**	**4.8771E+00**	2.7284E+00
DTLZ7	Sobol-HHMO	R-IGD	4.3763E+00	**2.6809E-01**	**3.7279E+00**	4.7489E+00
		R-HV	3.7361E+00	1.6197E+00	**7.9961E+00**	2.0127E+00
	2S-HHMO	R-IGD	**4.3438E+00**	3.0983E-01	3.7504E+00	**4.6385E+00**
		R-HV	**3.8013E+00**	**1.9403E+00**	7.7278E+00	**2.3651E+00**

The results in Table 5.7 show that forthe convergence and diversity, the 2S-HHMO algorithm has the lowest mean R-IGD and highest mean R-HV values in solving nine out of 12 problems. Sobol-HHMO has produced the lowest mean R-IGD and highest mean R-HV values for four problems. Thus, 2S-HHMO has managed to achieve lowest mean R-IGD and highest mean R-HV values in solving 75% of the problems while the Sobol-HHMO achieved highest convergence and diversity in solving 25% of the problems.

5.6.4 Comparing the Two-Step Initial Population Generator Method and LHS Sequence

This section presents the results obtained by the 2S-HHMO and LHS-HHMO algorithms. Table 5.8 shows the results of mean, SD, best and worst values of the R-IGD and R-HV metrics.

Table 5.8
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions obtained by using LHS-HHMO and 2S-HHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	LHS-HHMO	R-IGD	2.0610E-01	1.9693E-01	7.6789E-03	4.1516E-01
		R-HV	3.4261E+00	5.4476E-01	4.0374E+00	2.8657E+00
	2S-HHMO	R-IGD	**5.0148E-02**	1.3133E-01	4.2603E-03	4.2356E-01
		R-HV	**3.9127E+00**	3.7659E-01	4.0482E+00	2.8423E+00
ZDT2	LHS-HHMO	R-IGD	**8.3594E-02**	7.2033E-02	1.2880E-02	1.5240E-01
		R-HV	**3.5190E+00**	3.0038E-01	3.8142E+00	3.2328E+00
	2S-HHMO	R-IGD	1.1536E-01	1.5775E-01	1.4444E-02	4.9778E-01
		R-HV	3.4561E+00	4.8084E-01	3.8103E+00	2.4042E+00

ZDT3	LHS-HHMO	R-IGD	**3.1969E-02**	2.6474E-02	1.4701E-02	8.6623E-02
		R-HV	**3.8539E+00**	1.4988E-01	3.9525E+00	3.5692E+00
	2S-HHMO	R-IGD	1.0540E-01	1.3128E-01	1.4975E-02	3.2647E-01
		R-HV	3.6316E+00	4.2325E-01	3.9524E+00	2.9258E+00
ZDT4	LHS-HHMO	R-IGD	7.6815E-02	7.4391E-02	2.3524E-03	1.5866E-01
		R-HV	3.8219E+00	3.1495E-01	4.1524E+00	3.5014E+00
	2S-HHMO	R-IGD	**3.0926E-02**	5.7993E-02	2.3182E-03	1.4170E-01
		R-HV	**4.0169E+00**	2.5316E-01	4.1557E+00	3.5366E+00
ZDT6	LHS-HHMO	R-IGD	5.2891E-01	4.8909E-01	4.4910E-03	1.0481E+00
		R-HV	3.0198E+00	1.2798E+00	4.4119E+00	1.6900E+00
	2S-HHMO	R-IGD	**4.2721E-01**	4.8087E-01	4.5270E-03	1.0481E+00
		R-HV	**3.2869E+00**	1.2614E+00	4.4117E+00	1.6900E+00
DTLZ2 (3)	LHS-HHMO	R-IGD	1.7510E-01	2.8732E-02	1.2499E-01	2.2996E-01
		R-HV	8.3542E+00	2.7135E-01	8.8550E+00	7.8618E+00
	2S-HHMO	R-IGD	**1.5522E-01**	9.3412E-03	1.4206E-01	1.6712E-01
		R-HV	**8.5075E+00**	1.0532E-01	8.6657E+00	8.3749E+00
DTLZ2 (5)	LHS-HHMO	R-IGD	1.7211E-01	7.0780E-02	9.8325E-02	3.2204E-01
		R-HV	3.0737E+01	2.2567E+00	3.3381E+01	2.5946E+01
	2S-HHMO	R-IGD	**1.2505E-01**	1.2532E-02	1.0524E-01	1.4682E-01
		R-HV	**3.2338E+01**	6.8578E-01	3.3867E+01	3.1514E+01
DTLZ2 (10)	LHS-HHMO	R-IGD	6.9443E-01	5.5138E-02	6.2432E-01	8.1699E-01
		R-HV	7.1939E+02	8.8647E+01	8.2352E+02	5.2590E+02
	2S-HHMO	R-IGD	**6.6107E-01**	5.6263E-02	5.9793E-01	7.4668E-01
		R-HV	**7.9484E+02**	1.1115E+02	9.3754E+02	6.0796E+02
DTLZ4	LHS-HHMO	R-IGD	3.9232E-01	1.7187E-01	8.1885E-02	6.5949E-01
		R-HV	4.9138E+00	9.1491E-01	6.8194E+00	3.7493E+00
	2S-HHMO	R-IGD	**3.3989E-01**	1.5636E-01	1.8121E-01	6.5762E-01
		R-HV	**5.2203E+00**	8.4311E-01	6.1569E+00	3.7380E+00
DTLZ5	LHS-HHMO	R-IGD	**2.1123E-01**	8.2719E-02	1.7067E-01	4.3551E-01
		R-HV	**5.0219E+00**	5.7321E-01	5.4192E+00	3.5785E+00
	2S-HHMO	R-IGD	2.7318E-01	1.3629E-01	1.7057E-01	5.3443E-01
		R-HV	4.6520E+00	8.3100E-01	5.3988E+00	3.2199E+00
DTLZ6	LHS-HHMO	R-IGD	**2.2229E-01**	7.2036E-04	2.2082E-01	2.2332E-01
		R-HV	**4.8652E+00**	6.7378E-03	4.8770E+00	4.8530E+00
	2S-HHMO	R-IGD	2.6441E-01	1.3624E-01	2.2080E-01	6.5216E-01
		R-HV	4.6582E+00	6.7807E-01	4.8771E+00	2.7284E+00
DTLZ7	LHS-HHMO	R-IGD	**3.9762E+00**	3.3870E-01	3.6972E+00	4.6471E+00
		R-HV	**6.2031E+00**	2.1103E+00	8.2955E+00	2.3822E+00
	2S-HHMO	R-IGD	4.3374E+00	3.0916E-01	3.7504E+00	4.6299E+00
		R-HV	3.8415E+00	1.9395E+00	7.7278E+00	2.4022E+00

For the convergence and diversity, results of the 2S- HHMO algorithm has the lowest mean R-IGD and highest mean R-HV values in solving seven out of 12 problems. LHS-HHMO has produced the lowest mean R-IGD and highest mean R-HV values for four problems. Thus, 2S-HHMO has managed to achieve lowest mean R-IGD and highest mean R-HV values in solving 60% of the problems while the LHS-HHMO achieved highest convergence and diversity in solving 40% of the problems.

5.6.5 Comparing the Two-Step Initial Population Generator Method and Hammersley Sequence

This section presents the results obtained by the 2S-HHMO and Hammersley-HHMO algorithms. Table 5.9 shows the results of mean, SD, best and worst values of the R-IGD and R-HV metrics.

Table 5.9
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions obtained by using Hammersley-HHMO and 2S-HHMO algorithms

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	Hammersley-HHMO	R-IGD	**9.8198E-02**	1.7899E-01	3.8128E-03	4.1944E-01
		R-HV	**3.7743E+00**	5.1495E-01	4.0529E+00	2.8511E+00
	2S-HHMO	R-IGD	1.1246E-01	1.6827E-01	4.3219E-03	4.2355E-01
		R-HV	3.6952E+00	4.9262E-01	4.0479E+00	2.8424E+00
ZDT2	Hammersley-HHMO	R-IGD	**8.0921E-02**	7.0708E-02	1.3311E-02	1.5240E-01
		R-HV	**3.5300E+00**	2.9742E-01	3.8143E+00	3.2328E+00
	2S-HHMO	R-IGD	1.6299E-01	1.9161E-01	1.4899E-02	4.9583E-01
		R-HV	3.3418E+00	5.3884E-01	3.8099E+00	2.4306E+00
ZDT3	Hammersley-HHMO	R-IGD	2.1628E-01	2.5231E-01	1.4186E-02	6.4622E-01
		R-HV	3.3455E+00	6.8738E-01	3.9521E+00	2.2410E+00
	2S-HHMO	R-IGD	**5.1783E-02**	8.7024E-02	1.4646E-02	2.9476E-01
		R-HV	**3.8103E+00**	2.7850E-01	3.9524E+00	3.0708E+00
ZDT4	Hammersley-HHMO	R-IGD	7.5237E-02	1.1067E-01	2.6950E-03	3.3565E-01
		R-HV	3.8548E+00	3.9606E-01	4.1364E+00	3.0184E+00
	2S-HHMO	R-IGD	**3.9029E-02**	5.9179E-02	2.3182E-03	1.4170E-01
		R-HV	**3.9780E+00**	2.6260E-01	4.1557E+00	3.5366E+00
ZDT6	Hammersley-HHMO	R-IGD	**1.6402E-01**	3.5565E-01	4.5430E-03	1.0481E+00
		R-HV	**3.9845E+00**	9.4177E-01	4.4119E+00	1.6900E+00
	2S-HHMO	R-IGD	4.2721E-01	4.8087E-01	4.5270E-03	1.0481E+00
		R-HV	3.2869E+00	1.2614E+00	4.4117E+00	1.6900E+00
DTLZ2 (3)	Hammersley-HHMO	R-IGD	1.7577E-01	2.7187E-02	1.4053E-01	2.2024E-01
		R-HV	8.3484E+00	2.8217E-01	8.7483E+00	7.8824E+00
	2S-HHMO	R-IGD	**1.5455E-01**	1.0498E-02	1.4206E-01	1.7545E-01
		R-HV	**8.5108E+00**	1.0904E-01	8.6657E+00	8.3163E+00
DTLZ2 (5)	Hammersley-HHMO	R-IGD	2.4925E-01	6.2743E-02	1.6696E-01	3.5052E-01
		R-HV	2.8074E+01	2.1185E+00	3.0564E+01	2.4558E+01
	2S-HHMO	R-IGD	**1.2449E-01**	1.2984E-02	1.0524E-01	1.4682E-01
		R-HV	**3.2412E+01**	6.8429E-01	3.3867E+01	3.1514E+01
DTLZ2 (10)	Hammersley-HHMO	R-IGD	6.9418E-01	4.8060E-02	6.2572E-01	7.6212E-01
		R-HV	6.9518E+02	7.9710E+01	8.2674E+02	5.8797E+02
	2S-HHMO	R-IGD	**6.6107E-01**	5.6263E-02	5.9793E-01	7.4668E-01
		R-HV	**7.9489E+02**	1.1116E+02	9.3754E+02	6.0796E+02
DTLZ4	Hammersley-HHMO	R-IGD	**3.0354E-01**	1.7250E-01	1.2119E-01	6.6433E-01
		R-HV	**5.3975E+00**	9.2517E-01	6.4506E+00	3.6348E+00
	2S-HHMO	R-IGD	3.5796E-01	1.9453E-01	2.0025E-01	7.9367E-01
		R-HV	5.0842E+00	1.0435E+00	6.0512E+00	2.9208E+00
DTLZ5	Hammersley-HHMO	R-IGD	2.6708E-01	1.2189E-01	1.7144E-01	4.9533E-01
		R-HV	4.6553E+00	7.6107E-01	5.3302E+00	3.4036E+00

DTLZ6	2S-HHMO	R-IGD	**1.7907E-01**	8.4864E-03	1.7058E-01	1.9950E-01
		R-HV	**5.2561E+00**	1.2368E-01	5.3982E+00	4.9791E+00
	Hammersley-HHMO	R-IGD	**2.2123E-01**	2.9337E-03	2.1328E-01	2.2335E-01
		R-HV	**4.8731E+00**	2.3638E-02	4.9365E+00	4.8526E+00
	2S-HHMO	R-IGD	2.6441E-01	1.3624E-01	2.2080E-01	6.5216E-01
		R-HV	4.6582E+00	6.7807E-01	4.8771E+00	2.7284E+00
DTLZ7	Hammersley-HHMO	R-IGD	**4.1824E+00**	4.9758E-01	3.7050E+00	4.9456E+00
		R-HV	**5.1489E+00**	2.7809E+00	8.0539E+00	1.3751E+00
	2S-HHMO	R-IGD	4.3374E+00	3.0916E-01	3.7504E+00	4.6299E+00
		R-HV	3.8415E+00	1.9395E+00	7.7278E+00	2.4022E+00

The results in Table 5.9 show that in terms of the convergence and diversity, the 2S-HHMO algorithm has the lowest mean R-IGD and highest mean R-HV values in solving seven out of 12 problems. Hammersley-HHMO has produced the lowest mean R-IGD and highest mean R-HV values for four problems. Thus, 2S-HHMO has managed to achieve lowest mean R-IGD and highest mean R-HV values in solving 60% of the problems while the Hammersley-HHMO achieved highest convergence and diversity in solving 40% of the problems. Appendix B displays the comparison of the mean values of R-IGD, R-HV and epsilon obtained by each algorithm in solving MOPs.

5.7 Comparison of the Performance of 2S-ENDSHHMO with MOSI-Based Algorithms

This section presents the results of integrating the proposed population update and parameter adjustment strategies and the two-step initial population generator method, with the HHMO algorithm to produce the 2S-ENDSHHMO algorithm. To have statistical validity, it is necessary to use an appropriate number of algorithms and test problems in the performance comparisons. In doing so, the performance of the 2S-ENDSHHMO algorithm is compared with HHMO algorithm and eight other state-of-the-art algorithms. In the first phase, the algorithms are used to solve a set of MOPs

and their performances are compared with the 2S-ENDSHHMO algorithm in terms of the convergence and diversity of the obtained solutions. In the second phase, the performance 2S-ENDSHHMO algorithm is compared with the results of each algorithm based on the original publications. Appendix B.1 shows the distribution of non-dominated solutions obtained by the 2S-ENDSHHMO, HHMO and other algorithms.

The obtained distribution for the non-dominated solutions indicates that the 2S-ENDSHHMO algorithm can better handle concave, disconnected, multimodal, and degenerated MOPs. However, the distribution does not show how much better. Therefore, different performance metrics were used to quantitatively measure the performance of the algorithms. The mean, SD, best, worst R-IGD, R-HV and epsilon values of the non-dominated solutions obtained by each algorithm on all MOPs can be found in Appendix B.2.

Based on the mean R-IGD, the 2S-ENDSHHMO algorithm shows competitive performance compared to the other algorithms in solving eight out of 12 problems, which represents 66.7% of the problems. The lower value of R-IGD indicates that the 2S-ENDSHHMO algorithm have better convergence toward the true Pareto front in solving most MOPs compared to other algorithms. The second best convergence achieved by the R-NSGA-II algorithm, which have lower mean R-IGD values in solving 4 out of 12, which represents 33.3% of the problems. According to the mean epsilon values, the 2S-ENDSHHMO algorithm achieved lower epsilon values in

solving 50% of the problems. The second best performance achieved by the R-NSGA-II algorithm which have lowest epsilon values in solving 25% of the problems.

Based on the mean R-HV, the R-NSGA-II algorithm have highest R-HV values in solving 58.3% of the MOPs compared to other algorithms. The second best diversity achieved by the 2S-ENDSHHMO algorithm, in solving 41.6% of the MOPs. Although the R-NSGA-II algorithm outperformed the 2S-ENDSHHMO algorithm in terms of R-HV values, 2S-ENDSHHMO algorithm showed superior performance in terms of R-IGD and epsilon metrics. Figure 5.4 shows the average of R-IGD, R-HV and epsilon ranks obtained by the 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, with respect to each MOPs. The ranks are calculated based on the results in Appendix C.

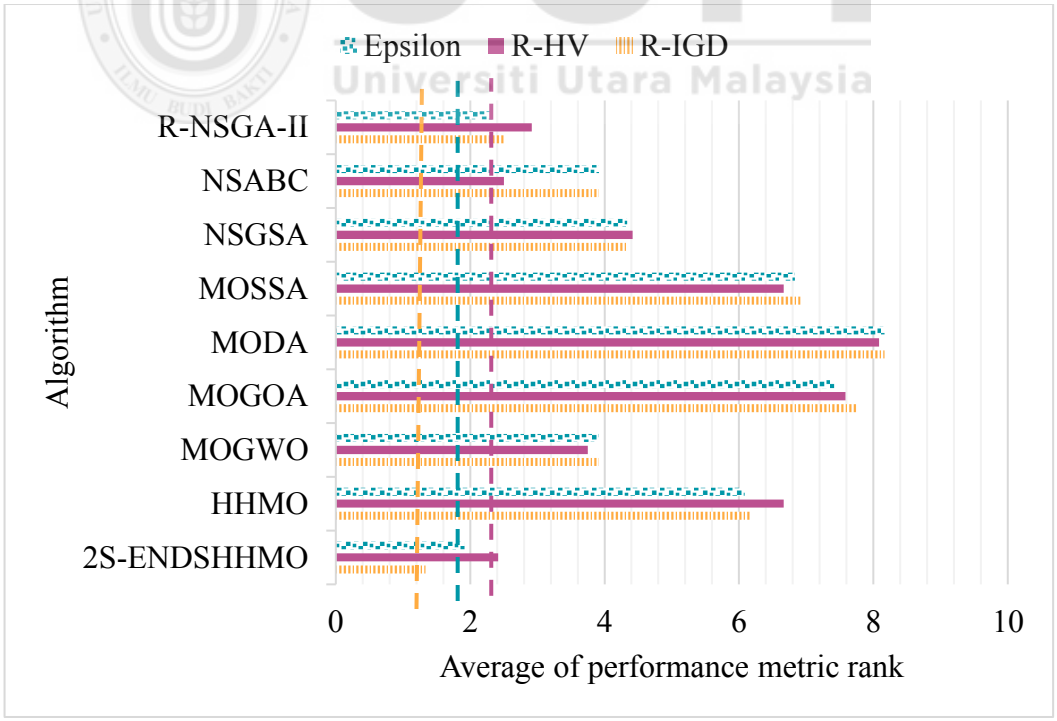


Figure 5.4. Average of R-IGD, R-HV and epsilon ranks obtained by the 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, in solving MOPs

In Figure 5.4, the striped and solid and dotted bars show the average of R-IGD, R-HV and epsilon ranks, respectively. The lower rank indicates better performance. Based on the results, the 2S-ENDSHHMO algorithm has achieved lowest rank in terms of R-IGD (1.33%) and epsilon (1.9%), in solving all MOPs.

In terms of R-HV, 2S-ENDSHHMO algorithm has achieved the lower rank of 2.4% and the second best performance achieved by NSABC with rank 2.5%. Based on the R-IGD and epsilon metrics, the second best performance achieved by R-NSGA-II with rank 2.5% and 2.3%, respectively.

In general, the results imply that the 2S-ENDSHHMO algorithm outperforms most of other algorithms, in terms of convergence toward the Pareto front and diversity of the non-dominated solution, in solving more than half of the MOPs. This indicates that the 2S-ENDSHHMO algorithm has a stronger ability to jump out of local optima and higher efficiency in solving complex MOPs with different PFs. This is due to the strategies used in the algorithm which show a balanced trade-off between exploration and exploitation, and ability in maintaining diversity of population. The general results of comparing the 2S-ENDSHHMO algorithm with other algorithms in terms of convergence and diversity by using mean R-IGD, R-HV and epsilon metrics are presented in Table 5.10.

Table 5.10
ZDT and DTLZ2 MOPs results based on the mean R-IGD, R-HV and *epsilon* values

MOP	Algorithm	2S-ENDSHHMO			MOP	Algorithm	2S-ENDSHHMO		
		R-IGD	R-HV	Epsilon			R-IGD	R-HV	Epsilon
ZDT1	HHMO	+	+	+	DTLZ2 (5)	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+
	NSGSA	+	+	+		NSGSA	+	+	+
	NSABC	+	+	+		NSABC	+	+	+
	R-NSGA-II	-	-	+		R-NSGA-II	+	+	-
ZDT2	HHMO	+	+	+	DTLZ2 (10)	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+
	NSGSA	+	+	+		NSGSA	+	+	+
	NSABC	+	+	+		NSABC	+	+	+
	R-NSGA-II	-	-	+		R-NSGA-II	-	-	-
ZDT3	HHMO	+	+	+	DTLZ4	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+
	NSGSA	+	+	+		NSGSA	+	+	+
	NSABC	+	+	+		NSABC	+	+	+
	R-NSGA-II	+	-	+		R-NSGA-II	+	-	+
ZDT4	HHMO	+	+	+	DTLZ5	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+
	NSGSA	+	+	+		NSGSA	+	+	+
	NSABC	+	+	+		NSABC	+	+	-
	R-NSGA-II	+	+	+		R-NSGA-II	+	-	+
ZDT6	HHMO	+	+	+	DTLZ6	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+
	NSGSA	+	+	+		NSGSA	+	+	+
	NSABC	+	+	+		NSABC	+	+	-
	R-NSGA-II	+	+	-		R-NSGA-II	+	-	+
DTLZ2 (3)	HHMO	+	+	+	DTLZ7	HHMO	+	+	+
	MOGWO	+	+	+		MOGWO	+	+	+
	MOGOA	+	+	+		MOGOA	+	+	+
	MODA	+	+	+		MODA	+	+	+
	MOSSA	+	+	+		MOSSA	+	+	+

NSGSA	+	+	+	NSGSA	+	+	+
NSABC	+	+	+	NSABC	+	+	-
R-NSGA-II	+	+	+	R-NSGA-II	-	+	+

In Table 5.10, green cells (+) denote cases where the 2S-ENDSHHMO algorithm statistically is better than the other algorithms. Cells marked in red cells (-) are cases where the algorithm yielded better results when compared to the 2S-ENDSHHMO algorithm.

For further evaluate the performance of the 2S-ENDSHHMO algorithm, it has been executed according to the parameters, performance metrics and MOPs used in the original publication of each algorithm, namely MOGWO (Mirjalili et al., 2016), MODA (Mirjalili, 2016), MOGOA (Mirjalili et al., 2018), MOSSA (Mirjalili et al., 2017), NSABC algorithm (Kishor et al., 2016) and R-NSGA-II (Deb & Sundar, 2006; Filatovas, Lančinskas, Kurasova, & Žilinskas, 2017) algorithms.

The metrics that have been used to measure the performance of the algorithms are IGD, GD, maximum spread (MS) and spacing (SP) metrics (Mirjalili et al., 2016). The MS and SP metrics are used to measure the diversity of the obtained non-dominated solutions., while the convergence is measured by using the GD and IGD metrics. Table 5.11 shows the performance metrics, parameter configuration and MOPs used with each algorithm.

Table 5.11
Summary of the performance metrics, parameters configurations and MOPs based on each publication

Algorithm Reference	MOP	Population size	*MaxIteration*	Metric	Number of Runs
MOGWO Mirjalili et al. (2016)	UF1-UF10	100	3000	IGD MS SP	10
MODA Mirjalili (2016)	ZDT1, ZDT2, ZDT3	30	500	IGD	10
MOGOA Mirjalili et al. (2018)	UF1-UF9	30	500	IGD MS SP	10
MOSSA Mirjalili et al. (2017),	UF1-UF10	60	1000	IGD	30
NSABC Kishor et al. (2016)	UF1 UF10	100	3000	IGD	30
R-NSGA-II Filatovas et al. (2017)	ZDT1, ZDT2, ZDT3, ZDT4, ZDT6	100	100	GD	100
	DTLZ2, DTLZ4, DTLZ5, DTLZ6	150	150	HV	

Comparisons of results have been performed. Tables 5.12-5.17 show the values of performance metrics in solving MOPs, with respect to each algorithm.

Table 5.12
2S-ENDSHHMO vs. MOGWO in solving UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 MOPs

MOP	Algorithm	Metric	Mean	SD	Best	Worst
UF1	2S-ENDSHHMO	IGD	**0.0008**	0.0001	0.0006	0.0010
		SP	0.0148	0.0052	0.0061	0.0271
		MS	**1.2280**	0.0462	1.1610	1.3660
	MOGWO	IGD	0.1144	0.0195	0.0802	0.1577
		SP	**0.0124**	0.0054	0.0146	0.0008
		MS	0.9268	0.9327	0.0688	0.8180
UF2	2S-ENDSHHMO	IGD	**0.0008**	0.0001	0.0006	0.0010
		SP	0.0136	0.0031	0.0082	0.0206
		MS	**1.3380**	0.0775	1.2200	1.4770
	MOGWO	IGD	0.0583	0.0074	0.0498	0.0732
		SP	**0.0111**	0.0095	0.0036	0.0076
		MS	0.9097	0.9104	0.0287	0.8470
UF3	2S-ENDSHHMO	IGD	**0.0019**	0.0002	0.0015	0.0024
		SP	**0.0362**	0.0263	0.0069	0.1119

		MS	**1.4230**	0.2033	1.1230	1.8420
		IGD	0.1223	0.0107	0.1049	0.1437
	MOGWO	SP	0.0459	0.0486	0.0145	0.0155
		MS	0.8720	0.8744	0.0056	0.8599
		IGD	**0.0008**	0.0000	0.0008	0.0008
	2S-ENDSHHMO	SP	0.0109	0.0074	0.0053	0.0348
		MS	**1.4150**	0.0046	1.4090	1.4270
UF4		IGD	0.0587	0.0005	0.0580	0.0594
	MOGWO	SP	**0.0097**	0.0086	0.0039	0.0058
		MS	0.9424	0.0009	0.9433	0.9410
		IGD	**0.0665**	0.0098	0.0507	0.0886
	2S-ENDSHHMO	SP	**0.0310**	0.0181	0.0093	0.0728
		MS	**1.5350**	0.1986	1.2840	1.9950
UF5		IGD	0.7971	0.3786	0.4680	1.7386
	MOGWO	SP	0.1523	0.0878	0.1625	0.0084
		MS	0.3950	0.1749	0.6104	0.0301
		IGD	**0.0034**	0.0006	0.0024	0.0045
	2S-ENDSHHMO	SP	0.0228	0.0315	0.0038	0.1789
		MS	**1.1380**	0.2228	0.8979	2.1000
UF6		IGD	0.2794	0.1045	0.1934	0.5504
	MOGWO	SP	**0.0145**	0.0111	0.0125	0.0019
		MS	0.6736	0.1232	0.8149	0.3884
		IGD	**0.0032**	0.0011	0.0009	0.0043
	2S-ENDSHHMO	SP	0.0154	0.0187	0.0048	0.0849
		MS	**1.1090**	0.3648	0.8078	2.0600
UF7		IGD	0.1604	0.1391	0.0628	0.4014
	MOGWO	SP	**0.0082**	0.0055	0.0086	0.0003
		MS	0.8013	0.3087	0.9875	0.0225
		IGD	**0.0017**	0.0000	0.0016	0.0018
	2S-ENDSHHMO	SP	**0.0047**	0.0008	0.0037	0.0062
		MS	**1.0060**	0.0682	0.9167	1.1090
UF8		IGD	2.0578	1.1455	0.4613	3.8789
	MOGWO	SP	0.0069	0.0047	0.0047	0.0037
		MS	0.4457	0.1857	0.8638	0.1886
		IGD	**0.0020**	0.0002	0.0016	0.0024
	2S-ENDSHHMO	SP	**0.0044**	0.0008	0.0034	0.0063
		MS	**1.0140**	0.1293	0.7491	1.4180
UF9		IGD	0.1917	0.0925	0.1291	0.4479
	MOGWO	SP	0.0174	0.0183	0.0063	0.0065
		MS	0.8399	0.1976	0.9375	0.2875
		IGD	**0.0023**	0.0003	0.0017	0.0027
	2S-ENDSHHMO	SP	**0.0135**	0.0092	0.0043	0.0428
		MS	**0.9923**	0.2121	0.7186	1.7780
UF10		IGD	3.5945	3.4883	1.0431x4	12.9564
	MOGWO	SP	0.0252	0.0150	0.0154	0.0000
		MS	0.2972	0.3465	0.9283	0.0319

Table 5.13
2S-ENDSHHMO vs. MOGOA in solving UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 MOPs

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		IGD	**0.0004**	0.0001	0.0004	0.0006
UF1	2S-ENDSHHMO	SP	0.0079	0.0037	0.0023	0.0146
		MS	**1.1518**	0.0282	1.1177	1.1929
	MOGOA	IGD	0.1811	0.0250	0.1430	0.1811

		SP	**0.0012**	0.0011	0.0000	0.0012
		MS	0.7270	0.1507	0.9120	0.7270
		IGD	**0.0004**	0.0000	0.0004	0.0005
	2S-ENDSHHMO	SP	0.0097	0.0033	0.0051	0.0149
		MS	**1.2046**	0.0949	1.1315	1.4069
UF2		IGD	0.0959	0.0386	0.0488	0.0959
	MOGOA	SP	**0.0007**	0.0011	0.0000	0.0007
		MS	0.8845	0.0353	0.9360	0.8845
		IGD	**0.0010**	0.0001	0.0008	0.0013
	2S-ENDSHHMO	SP	0.0111	0.0051	0.0040	0.0192
		MS	**1.1164**	0.0696	1.0177	1.2335
UF3		IGD	0.2380	0.0662	0.1682	0.2380
	MOGOA	SP	**0.0019**	0.0024	0.0000	0.0019
		MS	0.1100	0.7060	0.4026	0.1100
		IGD	**0.0005**	0.0001	0.0004	0.0006
	2S-ENDSHHMO	SP	0.0122	0.0185	0.0047	0.0647
		MS	**1.4060**	0.0054	1.4000	1.4186
UF4		IGD	0.0702	0.0048	0.0639	0.0702
	MOGOA	SP	**0.0001**	0.0002	0.0000	0.0001
		MS	0.9050	0.0139	0.9310	0.9050
		IGD	**0.1599**	0.0266	0.1260	0.2023
	2S-ENDSHHMO	SP	0.0465	0.0837	0.0085	0.2799
		MS	**1.3437**	0.1943	1.0214	1.6659
UF5		IGD	1.1559	0.1661	0.8978	1.1559
	MOGOA	SP	**0.0007**	0.0005	0.0001	0.0007
		MS	0.2379	0.1131	0.4894	0.2379
		IGD	**0.0020**	0.0002	0.0017	0.0025
	2S-ENDSHHMO	SP	0.0181	0.0112	0.0057	0.0364
		MS	**1.0662**	0.2298	0.8901	1.5261
UF6		IGD	0.7771	0.2769	0.4939	0.7771
	MOGOA	SP	**0.0003**	0.0004	0.0000	0.0003
		MS	0.1294	0.4600	0.0695	0.1294
		IGD	**0.0011**	0.0002	0.0007	0.0013
	2S-ENDSHHMO	SP	0.0064	0.0039	0.0035	0.0170
		MS	**0.8998**	0.1138	0.6866	1.0897
UF7		IGD	0.1726	0.0633	0.1150	0.1726
	MOGOA	SP	**0.0001**	0.0001	0.0000	0.0001
		MS	0.8460	0.0792	0.9570	0.8460
		IGD	**0.0018**	0.0002	0.0016	0.0021
	2S-ENDSHHMO	SP	**0.0072**	0.0030	0.0026	0.0127
		MS	**1.0460**	0.1105	0.8264	1.2318
UF8		IGD	0.2805	0.0749	0.2154	0.2805
	MOGOA	SP	0.0175	0.0085	0.0069	0.0175
		MS	0.4417	0.1586	0.6342	0.4417
		IGD	**0.0027**	0.0003	0.0021	0.0029
	2S-ENDSHHMO	SP	**0.0080**	0.0023	0.0045	0.0104
		MS	**0.9440**	0.2261	0.6343	1.4248
UF9		IGD	0.4885	0.1445	0.3336	0.4885
	MOGOA	SP	0.0234	0.0041	0.0172	0.0234
		MS	0.1635	0.6424	0.0677	0.1635

Table 5.14
2S-ENDSHHMO vs. NSABC in solving UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 MOPs

MOP	Algorithm	Metric	Mean	SD	Best	Worst
UF1	2S-ENDSHHMO	IGD	**0.0008**	0.0001	0.0006	0.0008
	NSABC	IGD	0.0085	0.0036	0.0035	0.0085
UF2	2S-ENDSHHMO	IGD	**0.0008**	0.0001	0.0006	0.0008
	NSABC	IGD	0.0070	0.0008	0.0059	0.0070
UF3	2S-ENDSHHMO	IGD	**0.0019**	0.0002	0.0015	0.0019
	NSABC	IGD	0.0482	0.0311	0.0081	0.0482
UF4	2S-ENDSHHMO	IGD	**0.0008**	0.0000	0.0008	0.0008
	NSABC	IGD	0.0220	0.0197	0.0266	0.0220
UF5	2S-ENDSHHMO	IGD	**0.0665**	0.0098	0.0507	0.0665
	NSABC	IGD	1.8896	0.8852	0.8137	1.8896
UF6	2S-ENDSHHMO	IGD	**0.0034**	0.0006	0.0024	0.0034
	NSABC	IGD	0.1836	0.1499	0.0258	0.1836
UF7	2S-ENDSHHMO	IGD	**0.0032**	0.0011	0.0009	0.0032
	NSABC	IGD	0.0146	0.0315	0.0037	0.0146
UF8	2S-ENDSHHMO	IGD	**0.0017**	0.0000	0.0016	0.0017
	NSABC	IGD	0.0502	0.0123	0.0522	0.0502
UF9	2S-ENDSHHMO	IGD	**0.0020**	0.0002	0.0016	0.0020
	NSABC	IGD	0.0608	0.0619	0.0345	0.0608
UF10	2S-ENDSHHMO	IGD	**0.0023**	0.0003	0.0017	0.0023
	NSABC	IGD	0.1167	0.1156	0.0542	0.1167

Table 5.15
2S-ENDSHHMO vs. MOSSA in solving UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 MOPs

MOP	Algorithm	Metric	Mean	SD	Best	Worst
UF1	2S-ENDSHHMO	IGD	**0.0003**	0.0000	0.0002	0.0004
	MOSSA	IGD	0.1024	0.0062	0.0897	0.1093
UF2	2S-ENDSHHMO	IGD	**0.0003**	0.0000	0.0002	0.0004
	MOSSA	IGD	0.0576	0.0048	0.0479	0.0657
UF3	2S-ENDSHHMO	IGD	**0.0008**	0.0001	0.0006	0.0011
	MOSSA	IGD	0.2628	0.0727	0.1711	0.4005
UF4	2S-ENDSHHMO	IGD	**0.0003**	0.0000	0.0003	0.0003
	MOSSA	IGD	0.0902	0.004	0.0855	0.0984
UF5	2S-ENDSHHMO	IGD	**0.0975**	0.0195	0.0720	0.1451
	MOSSA	IGD	0.6659	0.0986	0.4495	0.7914
UF6	2S-ENDSHHMO	IGD	**0.0015**	0.0002	0.0011	0.0020
	MOSSA	IGD	0.1903	0.0457	0.1163	0.2666
UF7	2S-ENDSHHMO	IGD	**0.0010**	0.0004	0.0003	0.0014
	MOSSA	IGD	0.069	0.0059	0.061	0.0796
UF8	2S-ENDSHHMO	IGD	**0.0017**	0.0001	0.0016	0.0018
	MOSSA	IGD	0.2743	0.0447	0.2249	0.3794
UF9	2S-ENDSHHMO	IGD	**0.0023**	0.0002	0.0020	0.0027
	MOSSA	IGD	0.4441	0.1084	0.2849	0.6422
UF10	2S-ENDSHHMO	IGD	**0.0021**	0.0004	0.0017	0.0027
	MOSSA	IGD	0.9769	0.2189	0.6082	1.3142

Table 5.16
2S-ENDSHHMO vs. MODA in solving the ZDT1, ZDT2, ZDT3 MOPs

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	2S-ENDSHHMO	IGD	**0.0006**	0.0000	0.0006	0.0006
	MODA	IGD	0.00612	0.002863	0.0024	0.0096
ZDT2	2S-ENDSHHMO	IGD	**0.0006**	0.0003	0.0005	0.0015
	MODA	IGD	0.00398	0.00160424	0.0023	0.006
ZDT3	2S-ENDSHHMO	IGD	**0.0016**	0.0000	0.0016	0.0016
	MODA	IGD	0.02794	0.004021	0.02	0.0304

Table 5.17
2S-ENDSHHMO vs. R-NSGA-II in solving the ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ2 DTLZ4, DTLZ5 and DTLZ6 MOPs

MOP	Algorithm	Metric	Mean
ZDT1	2S-ENDSHHMO	GD	**1.62E-04**
		HV	**5.19E-01**
	R-NSGA-II	GD	4.54E-04
		HV	1.13E-01
ZDT2	2S-ENDSHHMO	GD	2.73E-03
		HV	**2.77E-01**
	R-NSGA-II	GD	**1.05E-03**
		HV	2.74E-02
ZDT3	2S-ENDSHHMO	GD	**4.39E-03**
		HV	**6.31E-01**
	R-NSGA-II	GD	8.62E-03
		HV	3.61E-02
ZDT4	2S-ENDSHHMO	GD	**1.15E-04**
		HV	**4.86E-01**
	R-NSGA-II	GD	5.37E-02
		HV	1.13E-01
ZDT6	2S-ENDSHHMO	GD	**3.74E-07**
		HV	**2.44E-01**
	R-NSGA-II	GD	3.41E-04
		HV	7.28E-02
DTLZ2 (3)	2S-ENDSHHMO	GD	**1.56E-03**
		HV	**1.68E-01**
	R-NSGA-II	GD	4.92E-03
		HV	3.61E-03
DTLZ4	2S-ENDSHHMO	GD	6.95E-03
		HV	**4.26E-01**
	R-NSGA-II	GD	**2.52E-03**
		HV	4.43E-04
DTLZ5	2S-ENDSHHMO	GD	5.73E-04
		HV	**2.48E-01**
	R-NSGA-II	GD	**5.32E-04**
		HV	1.58E-03
DTLZ6	2S-ENDSHHMO	GD	**4.59E-07**
		HV	**1.48E-01**
	R-NSGA-II	GD	9.87E-04
		HV	2.98E-02

In general, the results in Tables 5.12-5.17 support the results of experiment in the first phase. The 2S-ENDSHHMO showed superior performance compared to each algorithm in solving different MOPs and based on different performance metrics.

Compared to the MOGWO, the 2S-ENDSHHMO algorithm showed lower mean IGD and higher MS values in solving all UF problems. The lower value of IGD indicates the 2S-ENDSHHMO algorithm have better convergence, while the higher MS indicates the 2S-ENDSHHMO algorithm have better diversity compared to the MOGWO. Based on the SP metric, the 2S-ENDSHHMO algorithm achieved lower values in solving 50% of the problems.

Although, the MOGOA achieved lower SP values in solving 7 out of 9 problems, which represents 77.9% of the problems, the 2S-ENDSHHMO algorithm showed lower mean IGD and higher MS values in solving all UF problems.

Compared to the NSABC, MOSSA and MODA, the 2S-ENDSHHMO algorithm showed lower mean IGD values in solving all UF problems. The lower value of IGD indicates the 2S-ENDSHHMO algorithm have better convergence compared to the NSABC, MOSSA and MODA. Compared to the R-NSGA-II, the 2S-ENDSHHMO algorithm showed lower mean GD values in solving seven out of nine, which represents 77.8% of the problems. However, based on the HV values, the 2S-ENDSHHMO algorithm showed higher HV values in solving all problems.

The general results indicate that the integration of the proposed population update and parameter adjustment strategies and the two-step initial population method with HHMO algorithm has improved its performance in solving most MOPs. The population update strategy includes the non-dominated sorting, which effectively reduces the number of non-dominated solutions in the high-dimensional space, enhances the selection pressure of the non-dominated solutions and improves the convergence of the algorithm. The parameter adjustment strategy helped improve exploration and exploitation of the algorithm, especially in solving MOPs with many objectives. The two-step initial population generator method initializes the population with candidate solutions, evenly distributed in the search space.

To statically measure the degree of differences in terms of convergence and diversity between the 2S-ENDSHHMO and other algorithms, the Wilcoxon rank sum test has been utilized. The *p*-values of mean R-IGD, R-HV and epsilon metrics in solving MOPs were calculated to perform pairwise comparison between the 2S-ENDSHHMO and each algorithm. The desired level of significance (α) is set to 0.05 (Roque, Fontes, & Fontes, 2017; Wang & Sun, 2018). The assumption is that there is no difference between the performance of the two algorithms. This represents the null hypothesis (Derrac et al., 2011). Table 5.18 shows the p-value obtained by 2S-ENDSHHMO compared to each algorithm.

Table 5.18
p-*values of the* Wilcoxon rank sum test on R-IGD,R-HV and epsilon values of 2S-ENDSHHMO and other algorithms

MOP	Algorithm	2S-ENDSHHMO					
		R-IGD		R-HV		Epsilon	
		H	*p*-value	H	*p*-value	H	*p*-value
ZDT1	HHMO	+	2.57E-02	+	1.83E-04	+	1.83E-04
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	+	2.17E-05	+	2.17E-05	+	2.17E-05
	MODA	+	2.17E-05	+	2.17E-05	+	2.17E-05
	MOSSA	+	2.17E-05	+	2.17E-05	+	2.17E-05
	NSGSA	+	1.83E-04	-	7.57E-02	+	1.83E-04
	NSABC	-	1.00E+00	-	3.07E-01	-	8.50E-01
	R-NSGA-II	+	2.20E-03	-	8.50E-01	-	5.21E-01
ZDT2	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	+	1.83E-04	+	2.17E-05	+	1.83E-04
	MODA	+	1.83E-04	+	2.17E-05	+	1.83E-04
	MOSSA	+	1.83E-04	+	2.17E-05	+	1.83E-04
	NSGSA	+	1.79E-04	+	7.57E-02	+	1.79E-04
	NSABC	-	8.90E-02	-	3.07E-01	+	1.73E-02
	R-NSGA-II	+	1.71E-03	-	8.50E-01	-	3.45E-01
ZDT3	HHMO	+	4.57E-05	+	9.14E-05	+	9.14E-05
	MOGWO	+	1.83E-04	+	3.30E-04	+	3.30E-04
	MOGOA	+	1.83E-04	+	1.73E-04	+	1.83E-04
	MODA	+	1.83E-04	+	6.39E-05	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.73E-04	+	1.83E-04
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	+	1.31E-03	+	2.83E-03	+	2.83E-03
ZDT4	HHMO	-	7.20E-01	+	2.17E-05	+	2.17E-05
	MOGWO	+	1.83E-04	+	2.83E-03	+	1.83E-04
	MOGOA	+	1.83E-04	+	6.39E-05	+	1.83E-04
	MODA	+	1.83E-04	+	6.39E-05	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	2.83E-03	+	2.57E-02	+	1.83E-04
	NSABC	+	2.83E-03	+	2.57E-02	+	1.83E-04
	R-NSGA-II	+	2.57E-02	-	6.78E-01	+	2.57E-02
ZDT6	HHMO	+	1.82E-04	+	1.82E-04	+	1.82E-04
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MODA	+	1.83E-04	+	1.49E-04	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	1.82E-04	+	1.82E-04	+	1.82E-04
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	-	6.78E-01	-	1.86E-01	-	1.62E-01
DTLZ2 (3)	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	+	2.57E-02	-	1.00E+00	+	2.83E-03
	MOGOA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MODA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOSSA	+	1.83E-04	-	1.40E-01	+	2.46E-04
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	+	1.83E-04	-	4.27E-01	+	1.83E-04
	R-NSGA-II	-	9.10E-01	-	6.78E-01	+	9.11E-03

DTLZ2 (5)	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MODA	+	3.30E-04	-	1.40E-01	+	1.83E-04
	MOSSA	+	1.83E-04	+	2.57E-02	+	1.83E-04
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	-	8.90E-02	+	1.83E-04	+	1.40E-02
	R-NSGA-II	+	1.83E-04	+	1.83E-04	+	2.83E-03
DTLZ2 (10)	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MODA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	+	1.83E-04	+	2.57E-02	-	1.40E-01
	R-NSGA-II	+	1.23E-03	+	1.03E-04	+	7.20E-04
DTLZ4	HHMO	+	1.03E-04	+	1.03E-04	+	1.03E-04
	MOGWO	-	2.12E-01	-	8.90E-02	-	8.50E-01
	MOGOA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MODA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	1.71E-03	+	1.01E-03	+	1.01E-03
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	+	1.83E-04	+	1.83E-04	+	1.83E-04
DTLZ5	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	+	2.83E-03	+	2.20E-03	+	1.83E-04
	MOGOA	+	2.46E-04	+	3.12E-02	+	4.52E-02
	MODA	+	1.83E-04	+	2.46E-04	+	5.80E-03
	MOSSA	+	2.57E-02	-	9.10E-01	+	2.57E-02
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	+	2.83E-03	+	2.83E-03	+	2.83E-03
DTLZ6	HHMO	+	1.83E-04	+	1.83E-04	+	1.31E-03
	MOGWO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGOA	-	1.40E-01	-	1.40E-01	-	4.73E-01
	MODA	+	1.83E-04	+	1.11E-04	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	+	1.83E-04	+	1.83E-04	+	1.83E-04
DTLZ7	HHMO	+	1.83E-04	+	1.83E-04	+	1.83E-04
	MOGWO	-	4.73E-01	-	1.00E+00	-	4.73E-01
	MOGOA	+	1.83E-04	+	1.49E-04	+	1.83E-04
	MODA	+	1.83E-04	+	6.39E-05	+	1.83E-04
	MOSSA	+	1.83E-04	+	1.83E-04	+	1.83E-04
	NSGSA	+	2.83E-03	+	2.57E-02	+	2.83E-03
	NSABC	+	1.83E-04	+	1.83E-04	+	1.83E-04
	R-NSGA-II	+	1.83E-04	+	1.83E-04	+	1.83E-04

In Table 5.18, the p-values less than 0.05 are marked with sign (+). The sign "+" and "-" indicate that the 2S-ENDSHHMO algorithm is superior to or inferior to other algorithm, respectively. According to the results, the approximation of 2S-

ENDSHHMO is significantly different than the other algorithm. Furthermore, the null hypothesis is rejected in most MOPs, denoted by the sign (+), which indicates that the 2S-ENDSHHMO outperforms the other algorithm with *p*-value.

5.8 Engineering Applications

The performance of the proposed 2S-ENDSHHMO algorithm is further evaluated using constraint engineering MOPs, namely, i) welded beam, ii) four-bar truss design, and iii) OPF problems. To solve the constrained optimization problems, the proposed 2S-ENDSHHMO algorithm is tested according to several studies that use penalty functions to handle constraints (Bassen et al., 2017; bin Mohd Zain et al., 2018; Farnad & Jafarian, 2018; Kohli & Arora, 2018; Savsani, 2014; Tawhid & Savsani, 2018a). The obtained results of the 2S-ENDSHHMO algorithm are compared with the results of other algorithms using R-IGD and R-HV performance metrics.

5.9 Experimental Design for Engineering problems

In solving the engineering problems, the performance of the 2S-ENDSHHMO algorithm is compared with the MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II algorithms. For each algorithm, the *MaxIteration* is set at 100, with a population size of 100 individuals. The control parameters of each algorithm are set as in Table 5.19. Each algorithm is run 10 times, producing 10 results. The mean, SD, best and worst R-IGD, R-HV and epsilon values of the obtained solutions are calculated for all algorithms. The algorithm stops if the number of iterations exceeds *MaxIteration*.

Table 5.19
Parameter settings of the optimization algorithms

Algorithm	Parameter	Value
MOGOA Mirjalili et al. (2018)	cMax	1
	cMin	0.00001
	Archive size	100
MODA Mirjalili (2016)	s: separation weight, a: alignment weight, c: cohesion weight, f: food factor, e: enemy factor, w: enemy factor	(Mirjalili, 2016)
MOSSA Mirjalili et al. (2017)	Archive size	100
NSABC Kishor et al. (2016)	Archive size	100
NSGSA Zellagui et al. (2017)	Percent of elitism	0.5
	Coefficient of search interval	2.5
	Sign mutation probability	0.9
	Uniform mutation probability	0.01
	Reordering mutation probability	0.4
	Initial value of inertial coefficient (W_0)	0.9
	Final value of inertial coefficient (W_1)	0.5
	Archive size	100
HHMO DeBruyne and Kaur (2016)	Radius	5
R-NSGA-II Deb et al. (2002a)	Crossover factor	0.9
	Mutation factor	1/N

5.9.1 Welded Beam Optimization Problem

The goal of optimizing the welded beam design is to find the best values for the design variables that minimize the end deflection and fabrication cost of the design. Apart from the parameter in Table 5.19, the reference points for the 2S-ENDSHHMO, HHMO and R-NSGA-II algorithms are set as (1.9,0.01); (9,0.0025); (37, 1.0000e-05). The optimization phases of the welded beam design are shown in Figure 5.5.

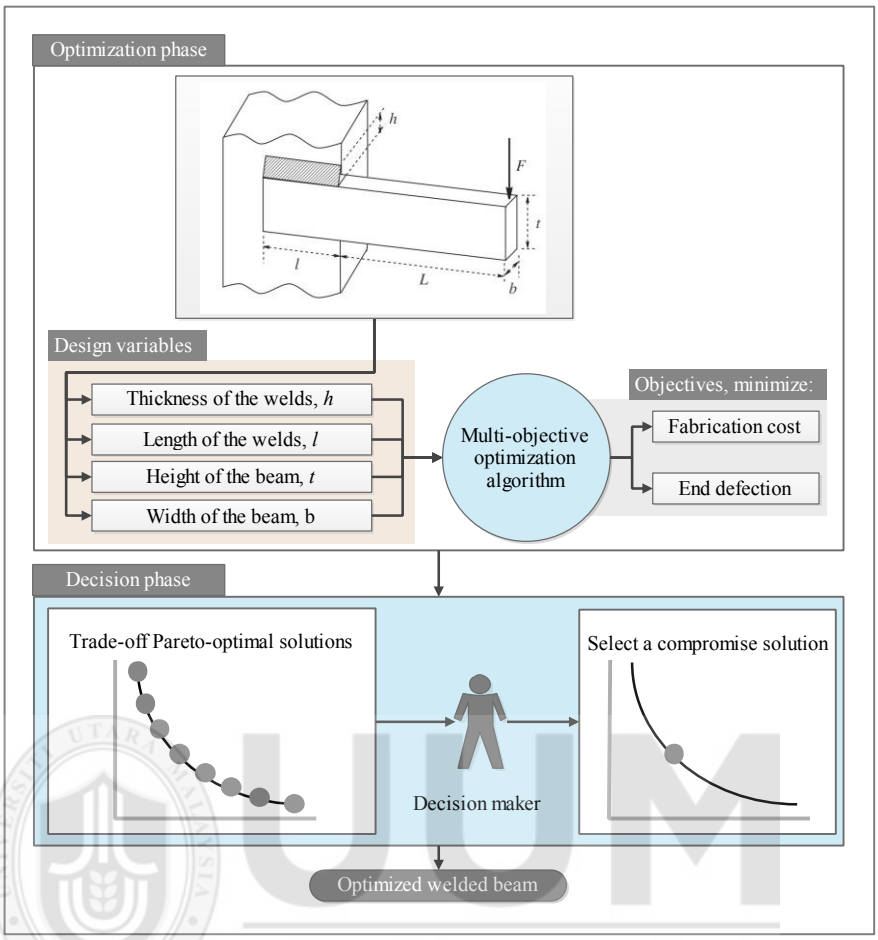


Figure 5.5. Optimization phases of the welded beam design MOP

Figure 5.5 shows that the welded beam design consists of four design variables (h, l, t, b); to find the best combination of these design variables, the optimization process starts by initializing the population with a set of potential solutions with four dimensions. Each algorithm is iterated several times (depending upon the number of *MaxIteration*); when the termination condition is met, the algorithm returns a set of non-dominated solutions. From this set, a compromise solution is selected manually by the DM or by using a post-Pareto analysis method. In solving the welded beam MOP, the mean, SD, best and worst R-IGD and R-HV values of the obtained solutions are calculated for all algorithms, as shown in Tables 5.20.

Table 5.20

Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions, obtained by 2S-ENDSHHMO and MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II algorithms, in solving welded beam optimization problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
Welded beam	2S-ENDSHHMO	R-IGD	**3.5057E-01**	1.2190E-01	2.5219E-01	6.8329E-01
		R-HV	**3.8620E+00**	7.1879E-01	4.1809E+00	1.8487E+00
	HHMO	R-IGD	9.5566E-01	1.7482E+00	2.7613E-01	5.2809E+00
		R-HV	2.8853E+00	1.2879E+00	4.1772E+00	0.0000E+00
	MOGWO	R-IGD	5.6968E-01	2.5026E-01	2.3801E-01	1.0851E+00
		R-HV	2.6064E+00	1.1024E+00	4.0168E+00	1.1207E+00
	MOGOA	R-IGD	3.8356E+00	6.1679E+00	2.3638E-01	2.0041E+01
		R-HV	1.4396E+00	1.4774E+00	4.0334E+00	0.0000E+00
	MODA	R-IGD	2.4432E+01	9.8502E+00	1.0915E+01	3.7613E+01
		R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	MOSSA	R-IGD	2.1849E+01	9.6645E+00	1.2866E+01	4.1401E+01
		R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	NSGSA	R-IGD	4.0732E+00	2.0506E+00	6.5731E-01	7.4301E+00
		R-HV	2.2326E-01	5.9942E-01	1.9037E+00	0.0000E+00
	NSABC	R-IGD	1.7462E+01	1.3396E+01	3.9063E+00	4.3908E+01
		R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	RNSGAII	R-IGD	2.0872E+00	2.7068E+00	2.7456E-01	9.4519E+00
		R-HV	1.0981E+00	9.9856E-01	3.0964E+00	0.0000E+00

In solving the welded beam MOP, in terms of convergence, the solution obtained by the 2S-ENDSHHMO algorithm has a lower mean R-IGD value of 3.5057E-01, which is 63.32%, 38.46%, 90.86%, 98.57%, 98.40%, 91.39%, 97.99% and 83.20% lower than HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, respectively. This indicates that the non-dominated solutions obtained by the 2S-ENDSHHMO algorithm have better convergence. In terms of diversity of the non-dominated solutions, the mean R-HV values obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms are 25.29%, 32.51%, 62.72%, 100.00%, 100.00%, 94.22%, 100.00% and 71.57% lower than 2S-ENDSHHMO algorithm. The higher mean R-HV value (3.8620E+00) indicates that the 2S-ENDSHHMO algorithm has better diversity.

5.9.2 Four-Bar Truss Design Optimization Problem

The goal is to optimize the four-bar truss design by finding the best values for the design variables that minimize the volume and deflection of the design. The same parameters presented in Table 5.19 are used apart from the reference points for the 2S-ENDSHHMO and HHMO algorithms which are set as (1120, 0.075); (1460, 0.05); (1860, 0.03). The optimization phases of the four-bar truss design are shown in Figure 5.6.

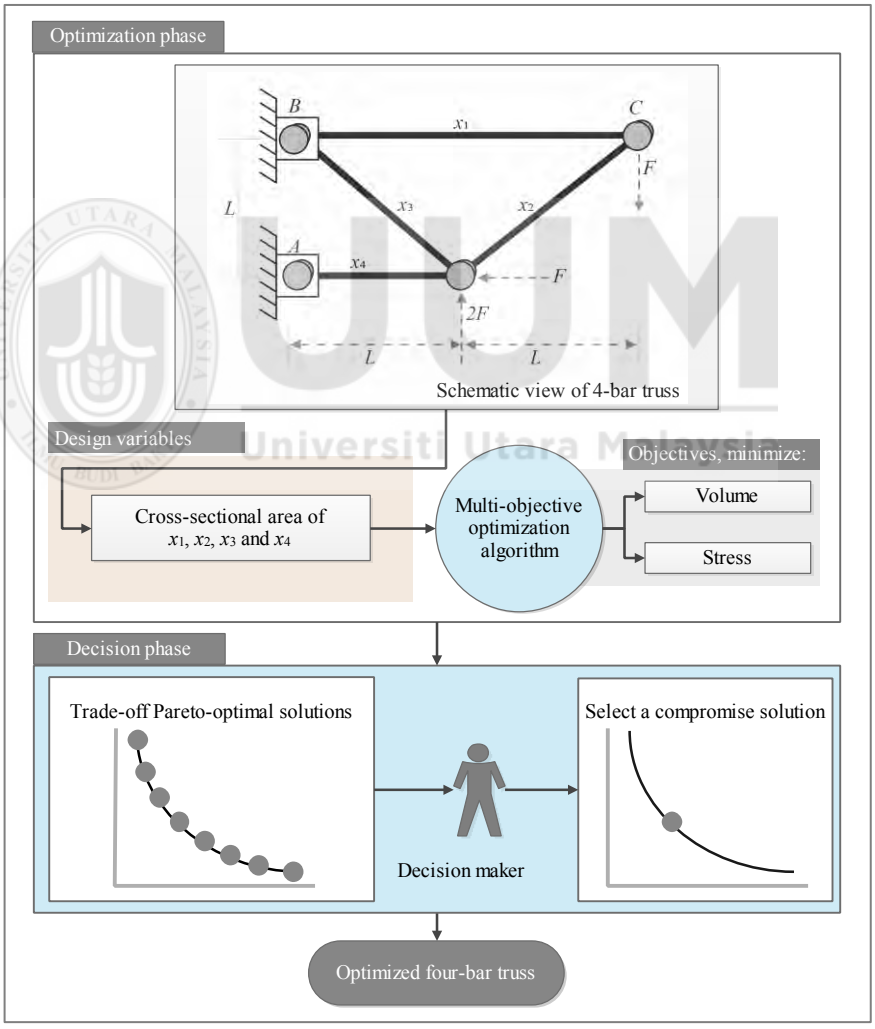


Figure 5.6. Optimization phases of the four-bar truss design MOP

The experiments are carried out to compare the performance of the 2S-ENDSHHMO algorithm to other algorithms. The mean, SD, best and worst R-IGD and R-HV values of the obtained solutions are calculated for all algorithms, as shown in Table 5.21.

Table 5.21
Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions, obtained by 2S-ENDSHHMO and MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II algorithms, in solving four-bar truss optimization problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
Welded beam	2S-ENDSHHMO	R-IGD	8.5352E-02	2.7544E-02	1.0767E-02	9.6641E-02
		R-HV	**4.0502E+00**	5.5279E-02	4.1999E+00	4.0275E+00
	HHMO	R-IGD	9.9625E-02	1.9236E-03	9.7674E-02	1.0274E-01
		R-HV	4.0158E+00	7.4669E-03	4.0234E+00	4.0037E+00
	MOGWO	R-IGD	9.8391E-02	1.1656E-03	9.7517E-02	1.0053E-01
		R-HV	4.0206E+00	4.5661E-03	4.0240E+00	4.0122E+00
	MOGOA	R-IGD	9.9196E-02	2.3676E-03	9.7448E-02	1.0533E-01
		R-HV	4.0175E+00	9.0939E-03	4.0243E+00	3.9940E+00
	MODA	R-IGD	**8.4688E-02**	2.9352E-02	1.9467E-02	1.1338E-01
		R-HV	4.0374E+00	6.0789E-02	4.1742E+00	3.9649E+00
	MOSSA	R-IGD	1.0040E-01	1.9406E-02	4.9005E-02	1.1860E-01
		R-HV	3.9976E+00	4.1206E-02	4.0976E+00	3.9467E+00
	NSGSA	R-IGD	9.7377E-02	3.6225E-04	9.6794E-02	9.7773E-02
		R-HV	4.0246E+00	1.4450E-03	4.0269E+00	4.0230E+00
	NSABC	R-IGD	9.7664E-02	9.7093E-04	9.6764E-02	9.9757E-02
		R-HV	4.0234E+00	3.8332E-03	4.0270E+00	4.0152E+00
	RNSGAII	R-IGD	9.6656E-02	1.5517E-05	9.6633E-02	9.6667E-02
		R-HV	4.0274E+00	6.2411E-05	4.0275E+00	4.0274E+00

For the convergent result, the 2S-ENDSHHMO algorithm has a lower mean R-IGD value of 8.5352E-02, which is 14.33%, 13.25%, 13.96%, 14.99%, 12.35%, 12.61% and 11.70% lower than HHMO, MOGWO, MOGOA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, respectively. This indicates that the non-dominated solutions obtained by the 2S-ENDSHHMO algorithm have better convergence. In terms of diversity of the non-dominated solutions, the mean R-HV values obtained by HHMO, MOGWO, MOGOA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms are 0.85%, 0.73%, 0.81%, 1.30%, 0.63%, 0.66% and 0.56% which are lower than 2S-ENDSHHMO algorithm. Although the mean R-IGD value of the MODA algorithm is

0.78% lower than algorithm. in terms of diversity, the mean R-HV value obtained by the MODA algorithm is 0.32% lower than 2S-ENDSHHMO algorithm, which indicates that the 2S-ENDSHHMO algorithm has better diversity compared to the MODA.

5.9.3 Optimal Power Flow Optimization Problem

Solving the OPF problem aims to show the performance of the proposed algorithm when considering the minimization of the generating cost of the system and active power losses in the transmission lines. The parameters of algorithms are set as presented in Table 5.19 apart from the *MaxIteration* which is set to 200 and the reference points for 2S-ENDSHHMO and HHMO algorithms set as (788, 8.4); (880, 4.5); (948, 2.5). The optimization phases of the OPF problem are shown in Figure 5.7.

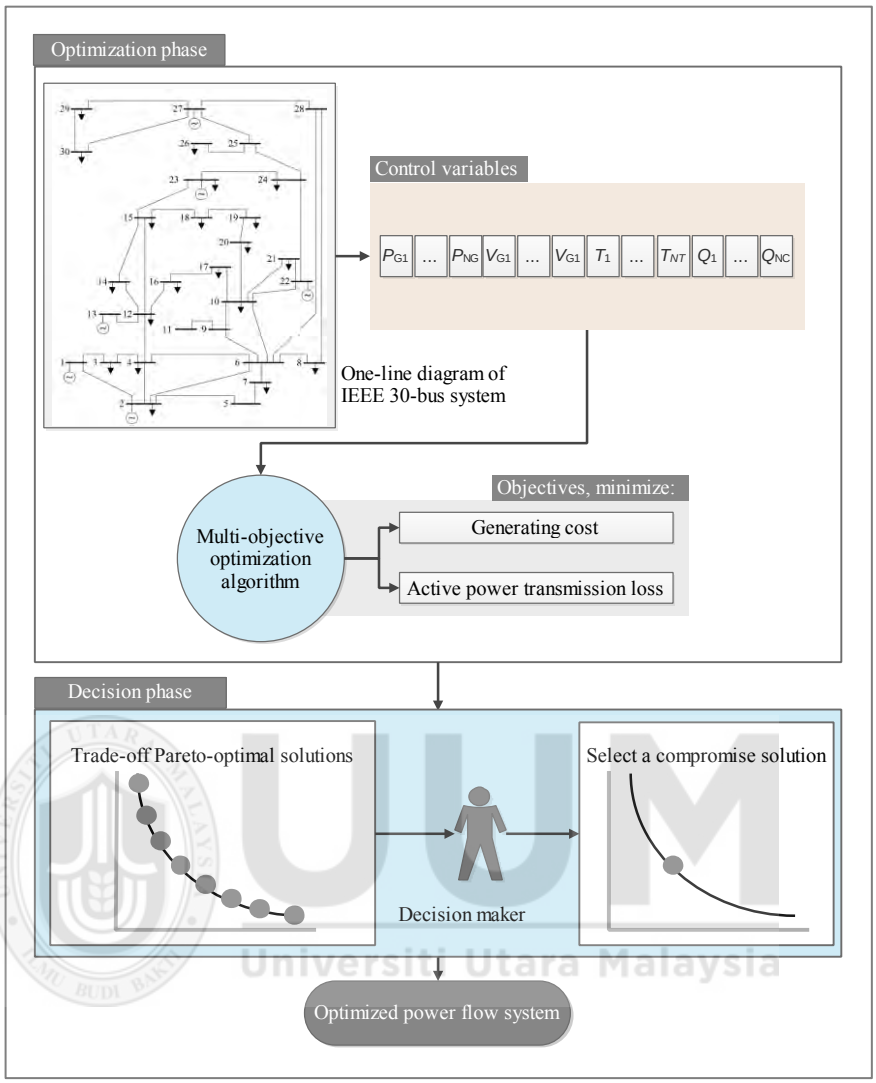


Figure 5.7. Optimization phases of the OPF design MOP

In general, solving the OPF problem through the optimization algorithm requires that the algorithm find the best combination control parameters of the system. The control variables are 24, that is, the active output of five generators (except the balance node), six generator bus voltage amplitudes, four transformer ratios, and nine reactive power compensation devices. The experiments are carried out to compare the 2S-ENDSHHMO algorithm to other algorithms. The mean, SD, best and worst R-IGD and R-HV values of the obtained solutions are calculated for all algorithms, as shown in Table 5.22.

Table 5.22

Mean, SD, best and worst R-IGD and R-HV values of non-dominated solutions, obtained by 2S-ENDSHHMO and MOGWO, MOGOA, MODA, MOSSA, NSABC, NSGSA and R-NSGA-II algorithms, in solving OPF optimization problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
Welded beam	2S-ENDSHHMO	R-IGD	**1.6703E+00**	2.2317E+00	4.3557E-01	5.9707E+00
		R-HV	4.8328E+00	2.7786E+00	6.6653E+00	0.0000E+00
	HHMO	R-IGD	3.6196E+00	6.1134E-01	2.6424E+00	4.7844E+00
		R-HV	2.2100E-01	2.9925E-01	9.6287E-01	0.0000E+00
	MOGWO	R-IGD	1.5468E+00	4.7642E-01	7.7052E-01	2.2992E+00
		R-HV	3.2112E+00	1.2063E+00	5.3959E+00	1.5007E+00
	MOGOA	R-IGD	3.6861E+00	0.0000E+00	3.6861E+00	3.6861E+00
		R-HV	5.8086E-02	0.0000E+00	5.8086E-02	5.8086E-02
	MODA	R-IGD	5.8936E+00	6.9916E-01	9.0426E-01	8.8348E+00
		R-HV	2.3569E+00	3.0931E+00	3.8574E+00	2.8802E-01
	MOSSA	R-IGD	3.1130E+00	2.2062E+00	1.5530E+00	4.6731E+00
		R-HV	1.5430E+00	2.1822E+00	3.0860E+00	0.0000E+00
	NSGSA	R-IGD	4.5185E+00	1.0263E+01	3.5927E-01	3.2689E+01
		R-HV	**5.3563E+00**	2.8275E+00	7.0138E+00	0.0000E+00
	NSABC	R-IGD	3.0522E+00	6.0970E-01	2.6211E+00	3.4834E+00
		R-HV	5.7047E-01	5.9727E-01	9.9280E-01	1.4814E-01
	RNSGAII	R-IGD	2.7747E+00	1.9828E+00	1.7255E+00	6.8035E+00
		R-HV	1.7940E+00	9.6021E-01	2.6728E+00	0.0000E+00

The R-IGD values in Table 5.22 show that the non-dominated solutions obtained by the 2S-ENDSHHMO algorithm has a lower mean R-IGD value of 8.5352E-02, which is 53.85%, 7.39%, 54.69%, 100.00%, 46.34%, 45.28% and 39.80% lower than HHMO, MOGWO, MOGOA, MOSSA, MODA, NSABC and R-NSGA-II algorithms, respectively. This indicates that the non-dominated solutions obtained by the 2S-ENDSHHMO algorithm has better convergence. In terms of diversity of the non-dominated solutions, the mean R-HV values obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSABC and R-NSGA-II algorithms are 95.43%, 33.55%, 98.80%, 100.00%, 68.07%, 88.20% and 62.88% lower than 2S-ENDSHHMO algorithm. The mean R-HV value of the 2S-ENDSHHMO algorithm is 9.77% lower than NSGSA algorithm. The higher mean R-HV value (5.3563E+00) indicates that the NSGSA algorithm has better diversity. However, in terms of convergence, the R-IGD value obtained by the 2S-ENDSHHMO algorithm is 63.03%

lower than NSGSA algorithm, which indicates that the 2S-ENDSHHMO algorithm has better convergence compared to the MODA.

5.9.4 Overall results of the engineering problems

Figures 5.8 shows the average of R-IGD and R-HV ranks obtained by the 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, in solving welded beam, four-bar truss and OPF MOPs.

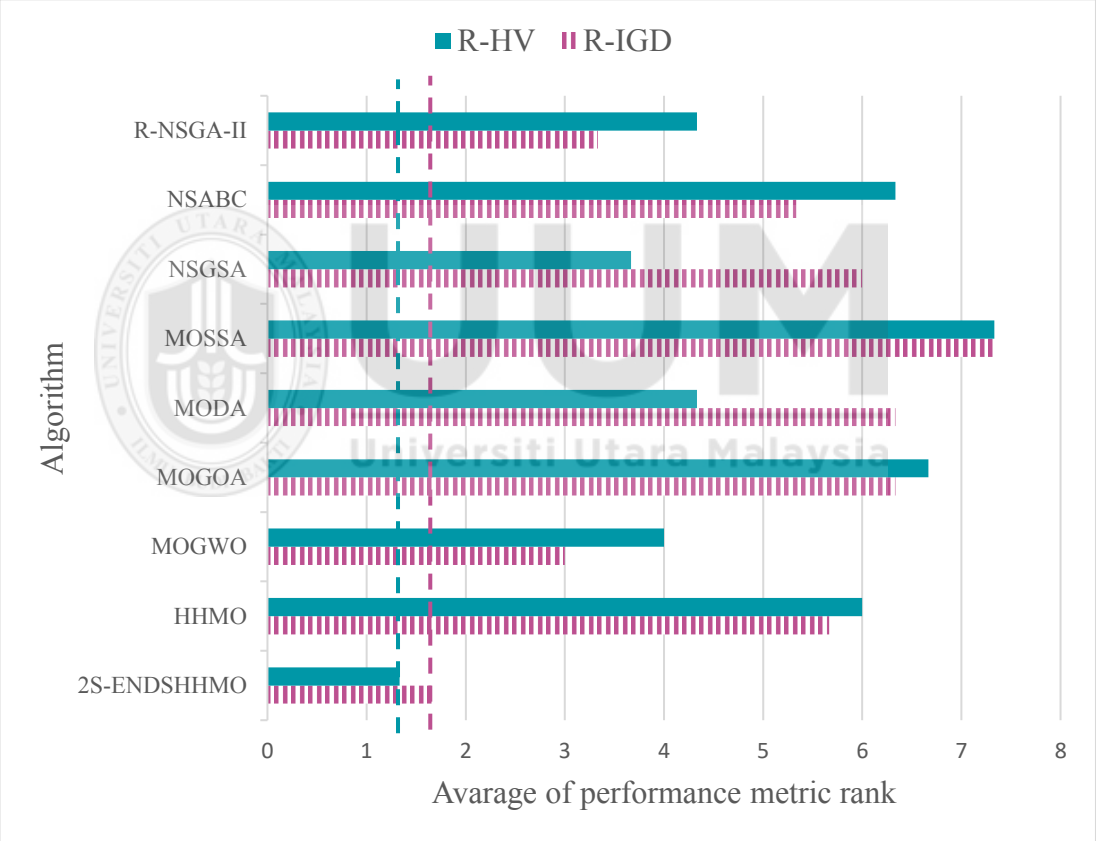


Figure 5.8. Average of R-IGD and R-HV ranks obtained by the 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, in solving welded beam, four-bar truss and OPF optimization problems

In Figure 5.80, the striped and solid bars show the average of R-IGD and R-HV rank, respectively. The lower rank indicates better performance. Based on the results, the 2S-ENDSHHMO algorithm has achieved lowest rank in terms of R-IGD (1.66%) and

R-HV (1.33%), in solving all engineering MOPs. In terms of R-IGD, the second best performance achieved by MOGWO (3%), while the second best R-HV rank (3.66%) achieved by NSGSA. In general, the results demonstrate the search ability of the 2S-ENDSHHMO algorithm in solving complex MOPs with no-convex boundary and several linear and nonlinear constraints.

5.10 Solutions Correspond to the Extreme Points

If the DM is interested in optimizing a single objective, then the decision variables corresponding to the extreme points in the objective space can be used. The extreme points correspond to decisions in which only one of the objectives is optimized. However, based on the definition of MOO, solving MOPs aims to simultaneously optimize all objectives. The results in Appendices D.1-D.27 show the selected solution based on the extreme points which achieves the best value for one objective and worst value for the other in solving welded beam, four-bar truss and OPF MOPs.

5.10.1 Welded Beam Design MOP

In solving the welded beam MOP using the 2S-ENDSHHMO algorithm, the minimum fabrication cost of 1.7916 units is obtained with design variables (h, l, t, b) = (0.2061, 3.3382, 9.4437, 0.2075) (inches), and the end deflection of the beam for these design variables is 0.0126 inch. Similarly, the minimum deflection, found to be 0.0004 inch, is obtained by using these design variables (1.6511, 0.3012, 10.0, 5.0) (inches). For this minimum deflection, the cost is found to be 35.3087 units. Appendix D.1 shows the decision variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving welded beam MOP.

By using the HHMO algorithm, the minimum cost of 1.8964 units is obtained by using these design variables (0.2530, 2.9645, 8.1542, 0.2535) (inches). For this minimum cost, the deflection is found to be 0.0160 inch. Similarly, the minimum deflection is 0.0005 inch for design variables (0.5717, 1.1301, 10.0, 4.7485) (inches). The fabrication cost of this design is 34.9732 units. Appendix D.2 shows the decision variables corresponding to extreme points, obtained by using HHMO algorithm in solving welded beam MOP.

The MOGWO is able to achieve a minimum deflection of 0.00044 inch by using these design variables (1.1416, 1.5903, 10.0, 5.0) (inches). For this minimum deflection, the cost is 39.7919 units. The minimum fabrication cost was obtained by using these design variables (0.2572, 2.7264, 9.1473, 0.2678) (inches) is 2.1702 units. The deflection with this cost is 0.01071 inch. Appendix D.3 shows the decision variables corresponding to extreme points, obtained by using MOGWO algorithm in solving welded beam MOP.

The minimum cost of welded beam design obtained by using the MOGOA algorithm is 2.4140 units with the design parameters to be (0.2575, 3.4483, 6.8986, 0.3732) (inches). With these parameters the deflection of the beam obtained is 0.0179inch. Similarly, the minimum deflection obtained is 0.00044inch with these design variables (3.20830, 0.18628, 10.0, 5.0) (inches) and the cost is 36.2433 units. Appendix D.4 shows the decision variables corresponding to extreme points, obtained by using MOGOA algorithm in solving welded beam MOP.

The solutions obtained by the MODA algorithm show the minimum cost of 3.7029 units, obtained by using these design variables (0.3132, 3.1707, 6.7287, 0.6044) (inches). For this minimum cost, the deflection is found to be 0.01192inch. Similarly, the minimum deflection is 0.0004 inch for design variables (4.7797, 0.1015, 10.0 4.9413). The fabrication cost of this design is 36.0834 units. Appendix D.5 shows the decision variables corresponding to extreme points, obtained by using MODA algorithm in solving welded beam MOP.

The MOSSA achieving a minimum cost of 2.4198 units is obtained with design variables (0.1250, 6.6276, 10.0, 0.2323) (inches). With this set of design variables, the end deflection of the beam is 0.0094 inch. The minimum deflection obtained is 0.0005 inch, with design variables (4.7387, 0.1382, 9.5608, 5.0) (inches) and cost 35.9447 units. Appendix D.6 shows the decision variables corresponding to extreme points, obtained by using MOSSA algorithm in solving welded beam MOP.

The NSGSA algorithm is able to achieve a minimum cost of 2.2046 units by using these design variables (0.1250, 6.3094, 10.0, 0.2145) (inches). For this minimum cost, the deflection is 0.01023 inch. The minimum deflection of 0.00044 inch is obtained by using these design variables (1.4979, 0.3582, 10.0, 5.0) (inches). Appendix D.7 shows the decision variables corresponding to extreme points, obtained by using NSGSA algorithm in solving welded beam MOP.

The minimum fabrication cost of welded beam design obtained using the NSABC algorithm is 4.4035 units with the design variables (0.4721, 1.4589, 8.5756, 0.6341) (inches). With these design variables the deflection of the beam obtained is 0.0055 inch. Similarly, the minimum deflection obtained is 0.0004 inch with these design variables (4.0123, 1.1024, 0.5123, 0.9073) and the cost is 36.0379 units. Appendix D.8 shows the decision variables corresponding to extreme points, obtained by using NSABC algorithm in solving welded beam MOP.

The minimum fabrication cost of welded beam design obtained using the R-NSGA-II algorithm is 1.8924 units with the design variables (0.1937, 3.6330, 10.0,0.2053) (inches). With these design variables the deflection of the beam obtained is 0.0107 inch. Similarly, the minimum deflection obtained is 0.0004 inch with these design variables (1.6125, 0.3090, 10.0, 5.0) (inches) and the cost is 35.3079 units. The decision variables corresponding to extreme points, obtained by using R-NSGA-II algorithm in solving welded beam MOP can be found in Appendix D.9. Figure 5.9 shows the minimum fabrication cost and deflection obtained by each algorithm in solving welded beam MOPs.

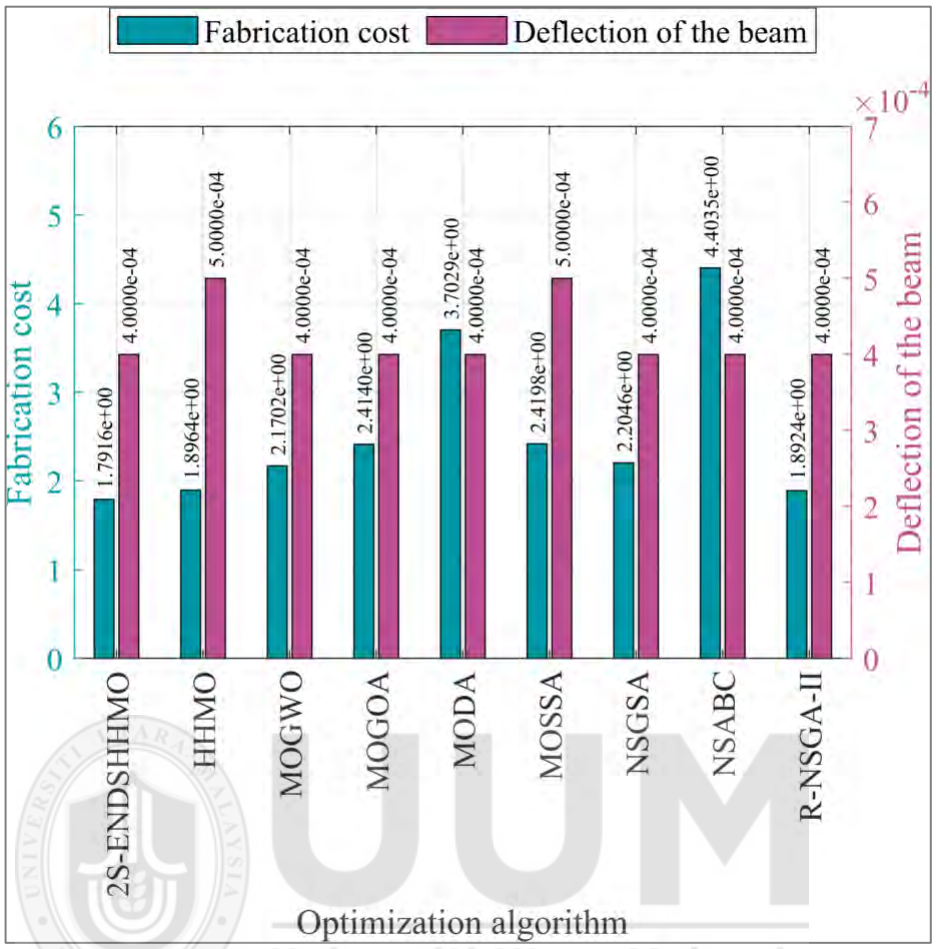


Figure 5.9. Solutions correspond to extreme points obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving welded beam MOP.

In Figure 5.9 the left and right y-axes show the values of the fabrication cost and deflection, respectively, obtained by each algorithm. In solving the welded beam MOP, the solution obtained by the 2S-ENDSHHMO algorithm has a minimum fabrication cost of 1.7916 units, which is 5.53%, 17.45%, 25.78%, 51.62%, 25.96%, 18.74%, 59.31% and 5.33% lower than HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms, respectively. The minimum deflection of 0.0004 inch, obtained by the 2S-ENDSHHMO algorithm, is 5.03%, 1.17% and 12.61% lower than HHMO, MODA and MOSSA algorithms, respectively.

With other algorithms, namely MOGWO, MOGOA, NSGSA, NSABC and R-NSGA-II, the minimum deflection is similar to that obtained by the 2S-ENDSHHMO algorithm.

5.10.2 Four-bar Truss

The minimum volume of the four-bar truss obtained using the 2S-ENDSHHMO algorithm is 1174.20 cm^3, obtained by using the solution (x1, x2, y) = (1.0, 1.4142, 1.4142, 1.0, 1174.200) (cm^2). The deflection for this solution is 0.0741 KPa. The minimum deflection is found to be 0.0322 KPa obtained by using these design variables (1.0, 3.0, 3.0, 3.0) (cm^2). For this minimum deflection, the volume is found to be 1836.3081 cm^3. Appendix D.10 shows the decision variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving four-bar truss MOP.

The HHMO algorithm is able to achieve a minimum volume of 1174.2000 cm^3 by using these design variables (1.0, 1.4142, 1.4142, 1.0) (cm^2). For this minimum volume, the deflection is 0.0741 KPa. The minimum deflection of 0.0322 KPa was obtained by using these design variables (1.0, 3.0, 3.0, 3.0) (cm^2). The volume with this deflection is 1836.3081 cm^3. Appendix D.11 shows the decision variables corresponding to extreme points, obtained by using HHMO algorithm in solving four-bar truss MOP.

The MOGWO algorithm achieving a minimum volume of 1174.2000 cm^3 is obtained with design variables (1.0, 1.4142, 1.4142, 1.0) (cm^2). With this set of design variables, the deflection of the truss is 0.0741KPa. The minimum deflection obtained is 0.0324 KPa, with design variables (1.0397, 3.0, 3.0, 2.9164) (cm^2) and volume 1835.4688 cm^3. Appendix D.12 shows the decision variables corresponding to extreme points, obtained by using MOGWO algorithm in solving four-bar truss MOP.

The solutions obtained by the MOGOA algorithm show that the minimum volume of 1188.4886 cm^3 is obtained by using these design variables (1.0, 1.4142, 1.5892, 1.0) (cm^2). For this minimum volume, the deflection is found to be 0.0719 KPa. Similarly, the minimum deflection is 0.0326 KPa for design variables (1.0, 3.0, 2.8880, 2.9894) (cm^2). The volume of this design is 1827.6524 cm^3. Appendix D.13 shows the decision variables corresponding to extreme points, obtained by using MOGOA algorithm in solving four-bar truss MOP.

The minimum volume obtained by using the MODA algorithm is 1177.7307 cm^3 with the design variables of (1.0005, 1.4142, 1.4540, 1.0) (cm^2). With these design variables, the deflection obtained is 0.0736 KPa. Similarly, the minimum deflection obtained is 0.0322 KPa with these design variables (1.0, 3.0, 3.0, 2.9828) (cm^2) and the volume of this design is 1832.8669 cm^3. Appendix D.14 shows the decision variables corresponding to extreme points, obtained by using MODA algorithm in solving four-bar truss MOP.

The MOSSA was able to achieve a minimum volume of 1183.2001 cm^3 by using these design variables (1.4909, 1.4143, 1.0, 1183.2001) (cm^2). For this minimum volume, the deflection is 0.0724 KPa. The minimum deflection obtained using these design variables (1.0, 3.0, 2.9942, 2.9156) (cm^2) is 0.0324KPa. The volume with this deflection is 1819.0943 cm^3. Appendix D.15 shows the decision variables corresponding to extreme points, obtained by using MOSSA algorithm in solving four-bar truss MOP.

By using NSGSA algorithm, the minimum volume is 1174.2000 cm^3, obtained by using these design variables (1.0, 1.4142, 1.4142, 1.0) (cm^2). For this minimum volume, the deflection is found to be 0.0741 KPa. Similarly, the minimum deflection is 0.0322 KPa for design variables (1.0, 3.0, 3.0, 3.0) (cm^2). The volume of this design is 1836.3081 cm^3. Appendix D.16 shows the decision variables corresponding to extreme points, obtained by using NAGSA algorithm in solving four-bar truss MOP.

In solving the four-bar truss MOP using the NSABC algorithm, the minimum volume of 1245.4801 cm^3 and deflection of 0.0632 KPa are obtained with design variables (1.0, 2.0457, 1.4264, 1.0104) (cm^2). Appendix D.17 shows the decision variables corresponding to extreme points, obtained by using NSABC algorithm in solving four-bar truss MOP.

By using R-NSGA-II algorithm, the minimum volume is 1174.2000 cm3, obtained by using these design variables (1.0, 1.4142, 1.4142, 1.0) (cm^2). For this minimum volume, the deflection is found to be 0.0741 KPa. Similarly, the minimum deflection is 0.0322 KPa for design variables (1.0, 3.0, 3.0, 3.0) (cm^2). The volume of this design is 1836.3081 cm^3. Appendix D.18 shows the decision variables corresponding to extreme points, obtained by using R-NSGA-II algorithm in solving four-bar truss MOP. Figure 5.10 shows the minimum volume and deflection obtained by each algorithm in solving four-bar truss MOPs.

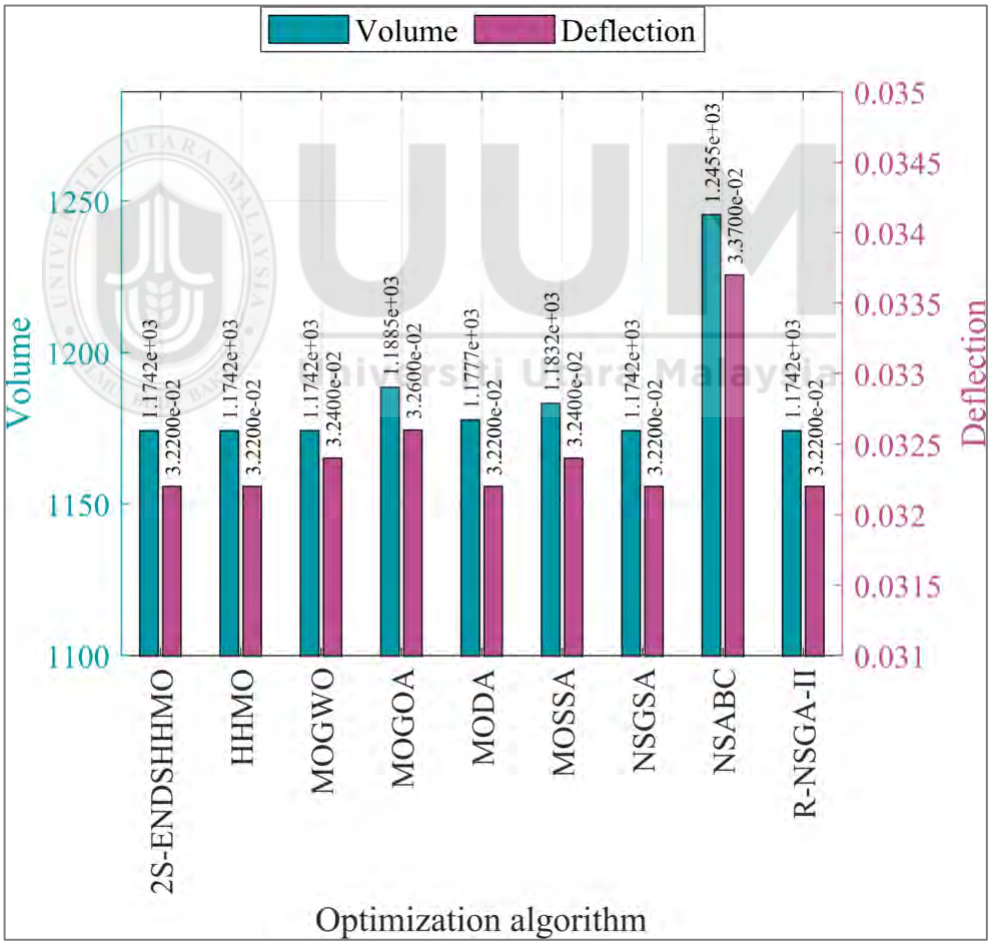


Figure 5.10. Solutions correspond to extreme points obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving four-bar truss MOP

In Figure 5.10 the left and right y-axes show the values of the volume and deflection, respectively, obtained by each algorithm. The minimum volume of 1174.2000 cm^3 achieved by 2S-ENDSHHMO, HHMO, MOGWO, NSGSA and R-NSGA-II algorithms, is 1.20%, 0.30%, 0.76% and 5.72% lower than the volume of MOGOA, MODA, MOSSA and NSABC algorithms. The minimum deflection of 0.0322 KPa is obtained by 2S-ENDSHHMO, HHMO, NSGA and R-NSGA-II algorithms. This value is 0.59%, 1.20%, 0.12%, 0.65% and 4.42% lower than MOGWO, MOGOA, MODA, MOSSA and NSABC algorithms, respectively.

5.10.3 Optimal Power Flow

In solving the OPF by using 2S-ENDSHHMO, the solution *A* has a minimum generating cost of 799.4458 units, and the transmission loss of the line for solution *A* is 8.5986 MW. Similarly, the minimum transmission loss found to be 2.9759 MW is obtained by using solution *B*. For this minimum transmission loss, the generating cost is found to be 967.1640 units. Appendix D.19 shows the control variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving OPF MOP.

By using the HHMO algorithm, the minimum generating cost of 799.5767 units is obtained by using solution *A*. The transmission loss of the line for solution *A* is 8.3032 MW. Similarly, the minimum transmission loss is found to be 3.4145 MW, obtained using solution *B*. For this minimum transmission loss, the generating cost is found to

be 948.2577 units. Appendix D.20 shows the control variables corresponding to extreme points, obtained by using HHMO algorithm in solving OPF MOP.

By using the MOGWO algorithm, the minimum generating cost of 800.7885 units is obtained by using solution *A*. For this minimum cost, the transmission loss is found to be 8.3282 MW. Similarly, the minimum transmission loss is 3.0389 MW for solution *B*. The generating cost of this solution is 967.5126 units. Appendix D.21 for control variables corresponding to extreme points, obtained by using MOGWO algorithm in solving OPF MOP.

The MOGOA algorithm is able to achieve minimum generating cost of 815.8582 units by using solution *A*. For this minimum cost, the transmission loss is 8.3707 MW. The minimum loss is obtained by using the solution *B* is 4.1514 MW. The cost with this solution is 899.4137 MW. Appendix D.22 shows the control variables corresponding to extreme points, obtained by using MOGOA algorithm in solving OPF MOP.

The solutions obtained by the MODA algorithm show that the minimum generating cost of 806.8893 units is obtained by using solution *A*. For this minimum cost, the loss is found to be 8.4853 MW. Similarly, the minimum loss is 4.4897 MW for solution *B*. The generating cost of this design is 880.6957 units. Appendix D.23 shows the control variables corresponding to extreme points, obtained by using MODA algorithm in solving OPF MOP.

In solving the OPF problem, the minimum generating cost of 805.3594 units is obtained by solution *A*. With this solution, the transmission loss is 8.3973 MW. The minimum loss obtained is 3.6307 MW with solution *B* and the cost is 912.3605 units. Appendix D.24 shows the control variables corresponding to extreme points, obtained by using MOSSA algorithm in solving OPF MOP.

The NSGSA algorithm is able to achieve a minimum generating cost of 803.0399 units by using solution *A*. For this minimum cost, the loss is 9.2673 MW. The minimum loss of 2.9666 MW is obtained by using solution *B*. The cost with this loss is 958.5062 units. Appendix D.25 shows the control variables corresponding to extreme points, obtained by using NSGSA algorithm in solving OPF MOP.

In solving OPF by using the NSABC algorithm, the minimum generating cost obtained by using solution *A* is 804.5842 units. With this solution, the loss is found to be 9.7718 MW. Similarly, the minimum loss obtained is 3.3845 MW with solution *B* and the cost of this solution is 960.1727 units. Appendix D.26 shows the control variables corresponding to extreme points, obtained by using NSABC algorithm in solving OPF MOP.

The R-NSGA-II algorithm is able to achieve a minimum generating cost of 799.7188 units by using solution *A*. For this minimum cost, the loss is 8.7950 MW. The minimum loss of 3.0058 MW is obtained by using solution *B*. The cost with this loss is 949.4215 units. Appendix D.27 shows the control variables corresponding to

extreme points, obtained by using R-NSGA-II algorithm in solving OPF MOP. Figure 5.11 shows the minimum generating cost and transmission loss obtained by each algorithm in solving OPF MOPs.

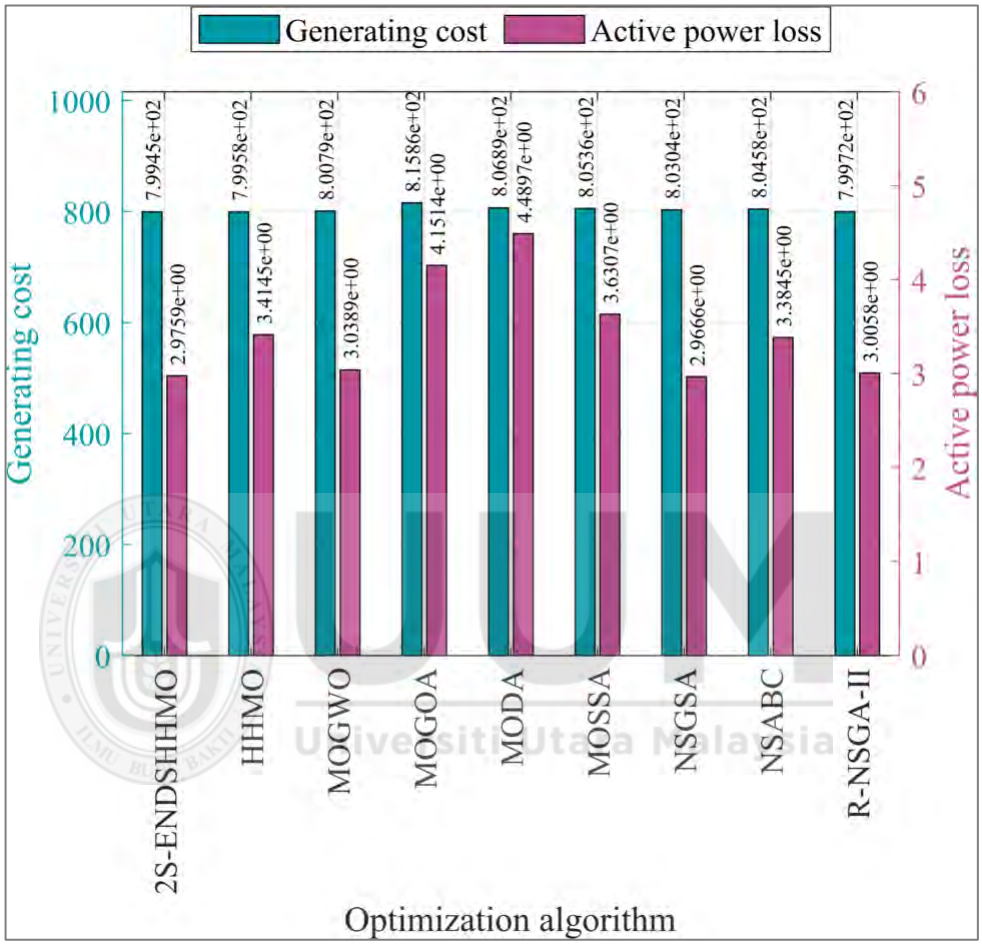


Figure 5.11. Solutions correspond to extreme points obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving OPF MOP.

In Figure 5.11 the left and right y-axes show the values of the minimum generating cost per hour and minimum active power transmission loss of the system, respectively, obtained by each algorithm. The minimum generating cost obtained by the 2S-ENDSHHMO algorithm is 799.4458 units. This cost is 0.02%, 0.17%, 2.01%, 0.92%, 0.73%, 0.45%, 0.64% and 0.03% lower than HHMO, MOGWO, MOGOA, MODA,

MOSSA, NSGSA, NSABC and R-NSGA-II algorithms. The minimum transmission loss of 2.9759 MW, obtained by 2S-ENDSHHMO algorithm, is 12.85%, 2.07%, 28.32%, 33.72%, 18.04%, 12.07% and 1.00 lower than HHMO, MOGWO, MOGOA, MODA, MOSSA, NSABC and R-NSGA-II algorithms, respectively. According to the results, However, the minimum transmission loss of NSGSA is and 0.31% lower than the 2S-ENDSHHMO algorithm. However, the minimum generating cost of this algorithms is higher than 2S-ENDSHHMO algorithm.

The general results of optimizing single objectives in solving welded beam, four-bar truss and OPF MOPs, indicates that the performance of the 2S-ENDSHHMO algorithm is superior compared to most MOSI-based algorithms. This means that solutions corresponding to the extreme points obtained by the 2S-ENDSHHMO algorithm are more attractive than solutions of other algorithms.

5.11 Select a Compromise Solution from the Pareto Set

In MOO, there is no single solution that optimizes all objectives at the same time but, rather, a set of effective solutions in which no solution is better than another for all objectives. In other words, solutions that are good compromises (trade-offs) instead of a single optimal global solution. The decision maker is responsible for choosing a a solution that addresses the overall objectives of the problem. This solution can be selected manually by the DM or by using a particular method, such as the grey relational analysis (GRA) technique for order of preference by similarity to ideal solution, linear programming technique for multidimensional analysis of preference,

simple additive weighting, faire un choix adéquat, which means making an adequate choice, and multiplicative exponent weighting (Wang & Rangaiah, 2017). In this study, the GRA was employed due to its simplicity and, in contrast to other methods such as the technique for order of preference by similarity to ideal solution and simple additive weighting, it does not require any inputs which may affect the selection of the compromise solution.

5.11.1 Grey Relation Analysis

The GRA method is considered as a core branch of the grey system theory (Julong, 1989). In general, the GRA is an effective multi-criterion decision making approach to analyse the degree of correlation between various factors in the system. This method has been widely used in several applications, such as environmental pollution assessment (Shao, Weng, Liou, Lo, & Jiang, 2019), portfolio management (Škrinjarić & Šego, 2019) and process parameter optimization (Mian, Umer, & Alkhalefah, 2019; Mishra, Das, Ukamanal, Routara, & Sahoo, 2015). The steps of the GRA method are as follows:

Step 1: normalizes the objective values of Pareto-optimal solutions based on the smaller-the-better, as shown in Equation (5.1) (Wang & Rangaiah, 2017).

$$F_{ij} = \frac{\max_{i \in m} f_{ij} - f_{ij}}{\max_{i \in m} f_{ij} - \min_{i \in m} f_{ij}} \tag{5.1}$$

Step 2: calculation of grey relational coefficient (GRC), as shown in Equation (5.2) (Panda, Sahoo, & Rout, 2016).

$$\mathrm{GRC}_i(k) = \frac{\Delta_{\min} + \xi \Delta_{\max}}{\Delta_{0i}(k) + \xi \Delta_{\max}} \tag{5.2}$$

where, $\Delta_{0i} = \|x_0(k) - x_i(k)\|$ is the deviation sequence of the reference sequence, $x_0(k)$ and comparability sequence, $x_i(k)$; i=1,2,…,M; k=1,2,…n, M and n are the experimental data and number of responses, respectively. Δ_{min} and Δ_{max} are the minimum and maximum values of the absolute differences (Δ_{0i}) of all comparing sequences. The value of ξ is set to 0.5 (Panda et al., 2016).

Step 3: calculation of grey relational grades (GRG), which is the weighted sum of the GRC, and its calculation formula as shown in Equation (5.3) (Panda et al., 2016).

$$\mathrm{GRG}_i = \frac{1}{n}\sum_{k=1}^{n} \mathrm{GRC}(k) \tag{5.3}$$

Step 4: Find the highest GRG_i, the corresponding solution is the recommended optimal solution.

In this study, the GRA method is applied to select one of the non-dominated solutions obtained by using MOO of the welded beam, four-bar truss design and OPF problems. To perform the GRA, first, the objective matrix (also known as decision matrix) is created. This matrix consists of M objectives (columns) and their values at n Pareto-optimal (non-dominated) solutions (rows), found by an optimization algorithm. This is followed by calculating the GRG values to determine the compromise solution.

5.11.2 Welded Beam Design

The results of the GRA consist of 100 solutions for each algorithm. Each solution is obtained by using 2S-ENDSHHMO, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms and selected according to the GRA value. Table 5.23 shows the selected solutions by using GRA method. The GRA of the

non-dominated solutions obtained by 2S-ENDSHHMO in solving welded beam MOP can be found in Appendix E. The Decision variables corresponding to the compromise solutions is presented in Table 5.24.

Table 5.23
GRA of selected compromise solutions obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving welded beam problem

Algorithm	No.	Objectives		Normalized Objectives		Highest GRG
		Fabrication cost	Deflection of the beam	Fabrication cost	Deflection of the beam	
2S-ENDSHHMO	47	8.5575	0.0019	0.7981	0.8778	0.7580
HHMO	2	6.4658	0.0026	0.8619	0.8600	0.7824
MOGWO	10	8.3400	0.0020	0.8360	0.8491	0.7606
MOGOA	17	6.2355	0.0031	0.8870	0.8466	0.7905
MODA	3	6.9759	0.0026	0.8989	0.8117	0.7791
MOSSA	13	12.2287	0.0015	0.7074	0.8877	0.7237
NSGSA	11	7.3507	0.0023	0.8451	0.8124	0.7453
NSABC	18	4.9464	0.0035	0.9828	0.9060	0.9043
R-NSGA-II	60	8.8183	0.0019	0.7927	0.8608	0.7446

Table 5.24
Decision variables corresponding to compromise solution, obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving welded beam problem

Algorithm	Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
2S-ENDSHHMO	47	0.6350	0.8612	10.0000	1.1433	8.5575	0.0019
HHMO	2	0.5105	1.2494	10.0000	0.8323	6.4658	0.0026
MOGWO	10	0.6637	0.8896	10.0000	1.1038	8.3400	0.0020
MOGOA	17	0.5985	1.8258	9.8579	0.7345	6.2355	0.0031
MODA	3	0.3218	2.5043	10.0000	0.8425	6.9759	0.0026
MOSSA	13	0.5716	2.1890	9.9591	1.4747	12.2287	0.0015
NSGSA	11	0.5407	1.0481	9.9781	0.9707	7.3507	0.0023
NSABC	18	0.4552	1.2652	10.0000	0.6341	4.9464	0.0035
R-NSGA-II	60	0.5712	0.9670	9.9998	1.1763	8.8183	0.0019

In solving welded beam MOPs using 2S-ENDSHHMO, the solution, (0.6350, 0.8612, 10.0, 1.1433) (inches) is selected by using the GRA method. The optimal objectives optimization value and its corresponding decision variable are obtained based on the

highest GRG value (0.7580) (refer Table 5.23). The cost and deflection values corresponding to the highest GRG are 8.5575 units and 0.0019 inch, respectively. By using the HHMO algorithm, the solution (0.5105, 1.2494, 10.0, 0.8323) (inches) obtained by using the GRA method produces a welded beam with the cost = 6.4658 units and deflection = 0.0026 inch. The solution (0.6637, 0.8896, 10.0) (inches) obtained by MOGWO is able to achieve cost = 8.3400 units and deflection = 0.0020 inch. The MOGOA produces the solution (00.5985, 1.8258, 9.8579, 0.7345) (inches), which is able to achieve the cost and deflection = 6.2355 units and 0.0031 inch, respectively. The solution (0.3218, 2.5043, 10.0, 0.8425) (inches) obtained by using MODA is able to achieve cost = 6.9759 units and deflection = 0.0026 inch, respectively. By using MOSSA, the cost and deflection obtained by using the GRA method, using the solution (0.5716, 2.1890, 9.9591, 1.4747) (inches) are 12.2287 units and 0.0015 inch, respectively. Using the GRA solution (0.5407, 1.0481, 9.9781, 0.9707) (inches) obtained by using the NSGSA algorithm, the cost and deflection values are found to be 7.3507 units and 0.0023 inch, respectively. The GRA solution of the NSABC algorithm achieves the cost and deflection values of 4.9464 units and 0.0035 inch, respectively. In the R-NSGA-II, the cost and deflection values obtained by using the GRA method are 8.8183 units and 0.0019 inch, respectively, for the solution (0.5712, 0.9670, 9.9998, 1.1763) (inches).

5.11.3 Four-Bar Truss Design

This section presents the results of selecting a compromise solution for the four-bar truss design problem. The selected compromise solutions by using GRA and the corresponding decision variables are shown in Tables 5.25-5.26, respectively.

Table 5.25
GRA of selected compromise solutions obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving four-bar truss problem

Algorithm	No.	Objectives		Normalized Objectives		Highest GRG
		Volume	Deflection	Volume	Deflection	
2S-ENDSHHMO	1	1174.2000	0.0741	1.0000	0.0000	0.6667
HHMO	80	1836.3081	0.0322	0.0233	1.0000	0.6693
MOGWO	2	1818.6926	0.0324	0.0254	0.9998	0.6693
MOGOA	85	1777.5842	0.0329	0.0783	0.9915	0.6675
MODA	1	1832.8669	0.0322	0.0000	1.0000	0.6667
MOSSA	1	1819.0943	0.0324	0.0000	1.0000	0.6667
NSGSA	1	1174.2000	0.0741	1.0000	0.0000	0.6667
NSABC	2	1245.4801	0.0632	1.0000	0.0000	0.6667
R-NSGA-II	1	1174.2000	0.0741	1.0000	0.0000	0.6667

Table 5.26
Decision variables corresponding to the compromise solution, obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving four-bar truss problem

Algorithm	Solution	x_1(cm^2)	x_2 (cm^2)	x_3(cm^2)	x_4(cm^2)	Volume (cm^3)	Deflection (KPa)
2S-ENDSHHMO	1	1.0000	1.4142	1.4142	1.0000	1174.2000	0.0741
HHMO	80	1.0000	3.0000	3.0000	3.0000	1836.3081	0.0322
MOGWO	2	1.0000	3.0000	3.0000	2.9119	1818.6926	0.0324
MOGOA	85	1.0000	2.9999	3.0000	2.7064	1777.5842	0.0329
MODA	1	1.0000	3.0000	3.0000	2.9828	1832.8669	0.0322
MOSSA	1	1.0000	3.0000	2.9942	2.9156	1819.0943	0.0324
NSGSA	1	1.0000	1.4142	1.4142	1.0000	1174.2000	0.0741
NSABC	2	1.0000	2.0457	1.4264	1.0104	1245.4801	0.0632
R-NSGA-II	1	1.0000	1.4142	1.4142	1.0000	1174.2000	0.0741

From Tables 5.25-5.26, in solving the four-bar truss problem using 2S-ENDSHHMO, the objectives value corresponding to the higher GRG (0.6667) are (1174.20006 cm^3, 0.0741 KPa), and the decision variables are (1.0, 1.4142, 1.4142, 1.0) (cm^2). By using the HHMO algorithm, the solution is (1.0, 3.0, 3.0, 3.0) (cm^2). The volume and

deflection values for this solution are found to be 1836.3081 cm^3 and 0.0322 KPa, respectively. The MOGWO algorithm produces the solution (1.0, 3.0, 3.0, 2.9119) (cm^2). The volume and deflection for this solution are 1818.6926 cm^3 and 0.0324KPa, respectively. The volume and deflection obtained by using the MOGOA algorithm are 1777.5842 cm^2 and 0.0329 KPa, respectively using the solution (1.0, 2.9999, 3.0, 2.7064) (cm^2). By using MODA, the volume and deflection values are found to be 1832.8669 cm^3 and 0.0322 KPa, respectively. By using MOSSA, the volume and deflection values obtained by using the GRA method are found to be 1819.0943 cm^3, 0.0324 KPa, respectively. By using NSGSA, the GRA solution (1.0, 1.4142, 1.4142, 1.0) (cm^2), volume and deflection are found to be 1174.20 cm^3 and 0.0741 Kpa, respectively. By using NSABC, the volume and deflection obtained by using the GRA method are 1245.4801 cm^3, 0.0632 KPa, respectively. By using R-NSGA-II, the GRA solution is (1.0, 1.4142, 1.4142, 1.0) (cm^2). The volume and deflection for this solution are 1174.2000 cm^3 and 0.0741 Kpa, respectively.

5.11.4 Optimal Power Flow

This section presents the results of selecting a compromise solution for the OPF MOP using the GRA method and the corresponding decision variables. The GRA calculation of the selected solutions are shown in Tables 5.28-5.27.

Table 5.27
GRA of selected compromise solutions obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving OPF problem

Algorithm	No.	Objectives		Normalized Objectives		HIGHEST GRG
		Cost	Active Power Loss	Cost	Active Power Loss	

2S-ENDSHHMO	31	800.9602	7.9018	0.9910	0.1239	0.6728
HHMO	23	802.0559	7.4339	0.9833	0.1778	0.6729
MOGWO	78	804.7520	7.0039	0.9762	0.2504	0.6774
MOGOA	15	899.4137	4.1514	0.0000	1.0000	0.6667
MODA	2	806.8893	8.4853	1.0000	0.0000	0.6667
MOSSA	6	808.9586	7.0976	0.9664	0.2727	0.6722
NSGSA	4	809.6164	7.3430	0.9577	0.3054	0.6703
NSABC	34	808.9318	7.7637	0.9721	0.3144	0.6844
R-NSGA-II	34	799.8664	8.5059	0.9990	0.0499	0.6714

Table 5.28

Decision variables corresponding to the compromise solution, obtained by using 2S-ENDSHHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II algorithms in solving OPF problem

		Algorithm								
	Control variables	**2S-ENDSHHMO**	**HHMO**	**MOGWO**	**MOGOA**	**MODA**	**MOSSA**	**NSGSA**	**NSABC**	**R-NSGA-II**
P_G (MW)	P_{G1}	164.80	159.44	152.93	90.63	159.19	150.47	147.53	156.71	173.16
	P_{G2}	49.09	48.83	48.22	67.89	47.89	47.57	52.59	47.83	48.74
	P_{G5}	21.88	22.62	24.02	46.09	23.56	27.33	25.70	28.39	21.81
	P_{G8}	26.11	30.66	33.06	18.02	24.82	28.20	27.21	21.95	23.58
	P_{G11}	14.39	15.19	15.54	30.00	18.49	20.12	18.69	15.87	12.62
	P_{G13}	15.03	14.08	16.63	34.92	17.83	16.82	19.03	20.41	12.00
V (p.u)	V_{G1}	1.10	1.10	1.10	1.10	1.04	1.10	1.06	1.06	1.09
	V_{G2}	1.09	1.09	1.09	1.09	1.02	1.08	1.05	1.04	1.07
	V_{G5}	1.07	1.06	1.06	1.07	0.99	1.04	1.01	1.01	1.04
	V_{G8}	1.07	1.08	1.07	1.07	1.01	1.06	1.03	1.02	1.05
	V_{G11}	1.10	1.10	1.09	1.09	1.04	1.07	1.06	1.06	1.10
	V_{G13}	1.10	1.09	1.09	1.10	1.04	1.05	1.05	1.06	1.10
Tap (p.u)	T_{11}	0.98	1.03	0.97	1.02	0.98	1.04	1.04	1.01	1.00
	T_{12}	0.96	0.98	1.03	0.95	1.00	0.90	0.93	0.96	0.92
	T_{15}	1.00	1.03	1.04	0.99	0.98	0.96	1.00	1.02	0.98
	T_{36}	0.98	0.99	0.99	0.97	0.98	1.02	0.96	0.97	0.95
Capacitor bank (MVar)	QC10	1.50	4.71	4.23	4.17	5.00	4.83	3.09	5.00	4.92
	QC12	0.49	3.68	3.15	0.01	2.94	4.57	2.70	2.77	4.81
	QC15	0.21	1.70	3.44	0.01	2.57	4.60	3.44	4.61	4.83
	QC17	4.40	2.37	4.09	0.75	2.00	1.54	4.80	5.00	4.58
	QC 20	1.16	2.84	3.70	4.91	2.46	1.03	2.92	4.57	4.94
	QC 21	1.33	4.71	1.75	2.94	3.00	0.72	3.86	5.00	5.00
	QC 23	4.87	4.64	2.94	2.79	4.00	4.11	2.26	4.31	4.29
	QC 24	0.02	4.45	4.62	3.88	2.00	0.63	2.88	5.00	5.00
	QC29	4.02	2.35	3.75	4.96	2.95	2.20	3.27	2.45	3.41

Generating cost (Unit/h)	800.96	802.06	804.75	899.41	806.89	808.96	809.62	808.93	799.87
Transmission loss (MW)	7.90	7.43	7.00	4.15	8.49	7.10	7.34	7.76	8.51

In solving the OPF MOP using 2S-ENDSHHMO, the optimal objectives optimization value and its corresponding control variables are obtained based on the highest GRG value (0.6728). The generating cost and transmission loss values corresponding to the selected solution are 800.9602 units and 7.9018 MW, respectively. By using the HHMO algorithm, the optimal objectives optimization value and its corresponding control variables are obtained based on the highest GRG value (0.6729). The generating cost and transmission loss values corresponding to the selected solution are 802.0559 units and 7.4339 MW, respectively. By using MOGWO, the selected solution has a generating cost = 804.7520 units and transmission loss = 7.0039 MW. By using MOGOA, the cost and loss values obtained by using the GRA method are 899.4137 units and 4.1514 MW, respectively. By using MODA, For the GRA solution, the cost and loss are 806.8893 units and 8.4853 MW, respectively. By using the MOSSA, the cost and loss obtained by using GRA solution are 808.9586 units and 7.0976 MW, respectively. By using the NSGSA, the GRA solution, the cost and transmission loss values are found to be 809.6164 units and 7.3430 MW, respectively. By using the NSABC algorithm, for the GRA solution, the cost and loss values are

found to be 808.9318 units and 7.7637 MW, respectively. The solution obtained by using R-NSGA-II produced cost and transmission loss values of 799.8664 units and 8.5059 MW, respectively.

5.11.5 The Compromise Solutions of the Engineering MOPs

Comparison of the compromise solution obtained by the 2S-ENDSHHMO algorithm with solutions obtained by using other algorithms in solving welded beam, four-bar truss and OPF MOPs are shown in Figures 5.12-5.14.

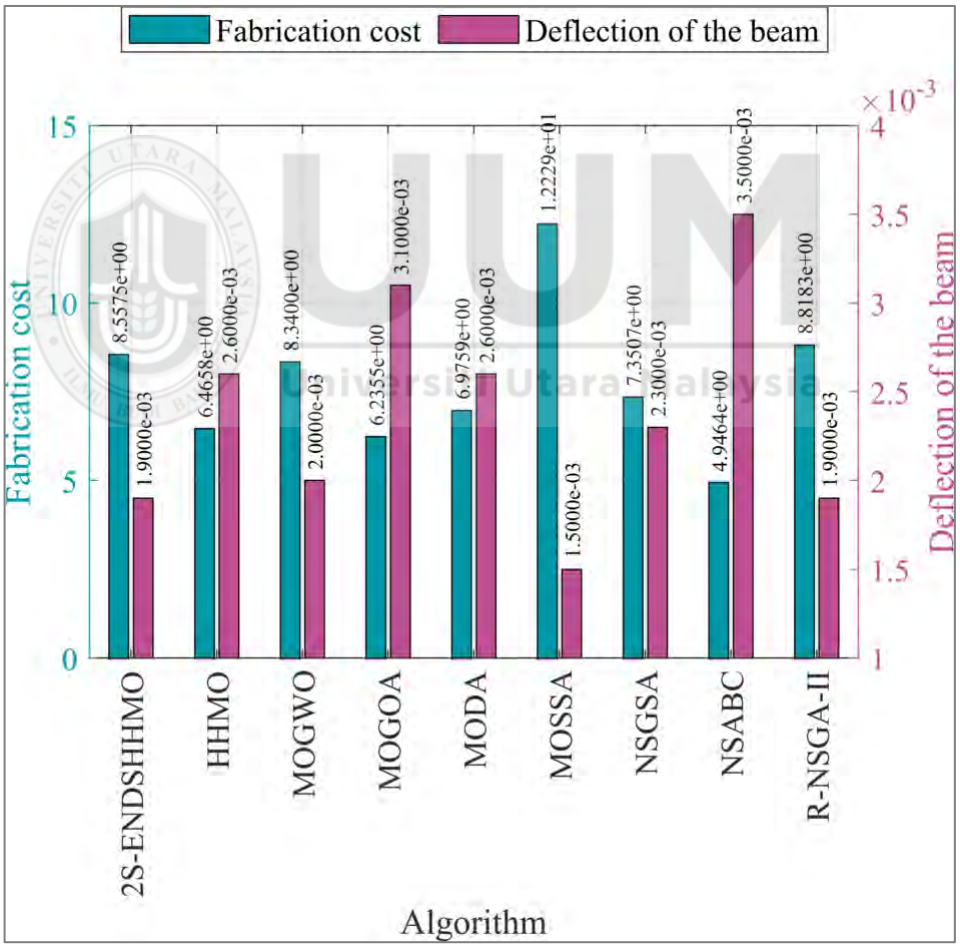


Figure 5.12. Compromise solutions obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving welded beam MOP.

In solving welded beam MOPs, the solution obtained by the 2S-ENDSHHMO algorithm has a fabrication cost of 8.5575 units which is 30.02% and 2.96% lower than the solutions of MOSSA and R-NSGA-II algorithms, respectively. The solutions obtained by HHMO, MOGWO, MOGOA, MODA, NSGSA and NSABC algorithms have a fabrication cost of 24.44%, 2.54%, 27.13%, 18.48%, 14.10% and 42.20% lower than the 2S-ENDSHHMO algorithm. However, the solution obtained by the 2S-ENDSHHMO algorithm has a deflection of 27.20%, 3.45%, 38.45%, 26.31%, 15.65% and 44.54% lower than the solution obtained by using HHMO, MOGWO, MOGOA, MODA, NSGSA and NSABC algorithms, respectively.

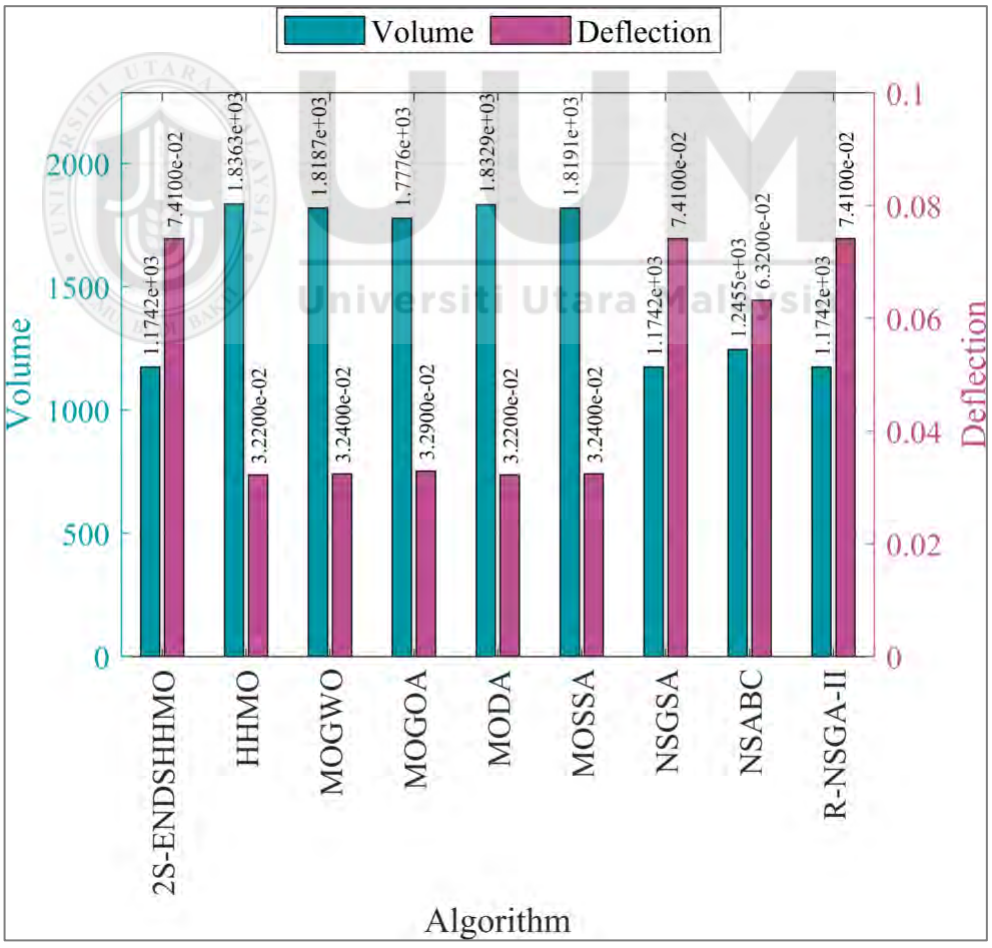


Figure 5.13. Compromise solutions obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving four-bar truss MOP

In solving the four-bar truss MOP, the solution obtained by the 2S-ENDSHHMO algorithm has a volume of 36.06%, 35.44%, 33.94%, 35.94%, 35.45% and 5.72% lower than the solutions of HHMO, MOGWO, MOGOA, MODA and NSABC algorithms. However, the deflection of solutions of these algorithms is 56.58%, 56.31%, 55.61%, 56.53%, 56.30% and 14.72% which is lower than the solution of 2S-ENDSHHMO algorithm.

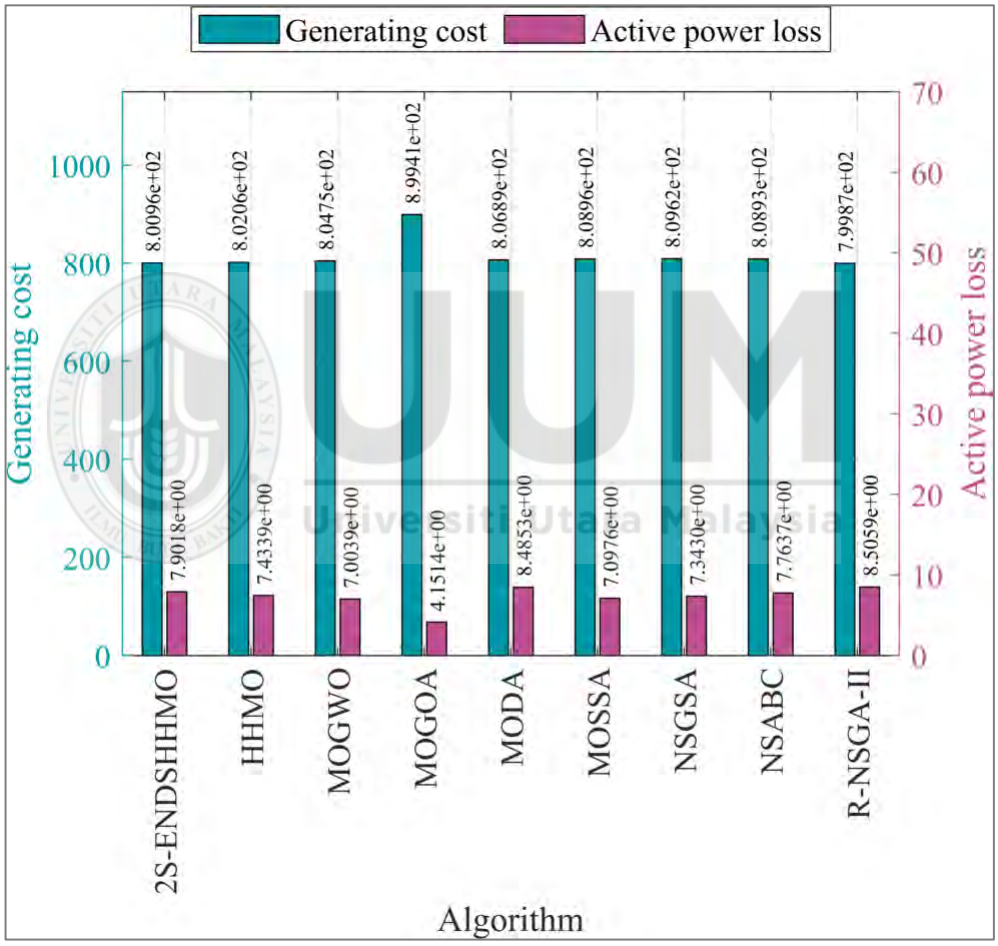


Figure 5.14. Compromise solutions obtained by HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA and NSABC and R-NSGA-II algorithms in solving OPF MOP.

In solving the OPF MOP, the solution obtained by the 2S-ENDSHHMO algorithm has a generating cost of 0.14%, 0.47%, 10.95%, 0.73%, 0.99%, 1.07%, and 0.99% lower

than the solution obtained by HHMO, MOGWO, MODA, MOSSA, NSGSA and NSABC, respectively. The solution obtained by R-NSGA-II algorithm has a generating cost of 0.14% lower than 2S-ENDSHHMO. However, the active transmission loss of this solution is 7.102% higher than the solution of 2S-ENDSHHMO. The transmission loss of HHMO, MOGWO, MODA, NSGSA, NSABC is 5.9219%, 11.3640%, 47.4628%, 6.8764%, 7.0725%, 1.7475% and 7.1020 lower than 2S-ENDSHHMO algorithm, respectively. However, the generating cost of these algorithms is high compared to the 2S-ENDSHHMO. The MOSSA and R-NSGA-II algorithms have a transmission loss of 7.102% and 10.1783% higher than the 2S-ENDSHHMO algorithm. This makes the compromise solution obtained by the 2S-ENDSHHMO algorithm more economically than solutions of other algorithms.

Since the objectives are conflicting with each other, improving one of the them leads to degradation of the other. In Figures 5.1-5.14, the lower value of an objective does not mean the corresponding compromise solution is the better. Because the value of the other objective became higher. In MOO we need to take all objective in consideration. The calculation of compromise solutions helps the DM to choose one of the acceptable solutions belonging to the Pareto front. According to Chiandussi, Codegone, Ferrero, and Varesio (2012), the DM usually selects a compromise solution satisfying all the objectives as better as possible.

5.12 Summary

In this chapter the experiments have been conducted to evaluate the performance of the proposed 2S-ENDSHHMO algorithm and compare it with other algorithms in solving different MOPS. The results show that the 2S-ENDSHHMO algorithm promises to be a competitive metaheuristic in the field of MOO due to its high performance compared to the state-of-the-art algorithms used in the experiments. The 2S-ENDSHHMO algorithm was able to generate a set of non-dominated solutions with greater convergence and diversity to the true Pareto front. of most MOPs used in the experiments and showed a great ability in jumping out of local optima, which indicates the ability of the algorithm in exploration of the search space. Moreover, the 2S-ENDSHHMO algorithm was able to achieve competitive results in solving real-world MOPs compared to other algorithms.

CHAPTER SIX
CONCLUSION AND FUTURE WORKS

Most real-world optimization problems consist of two or more conflicting objectives, in which improving an objective leads to the degradation of others. This type of problem is known as a MOP. Several methods have been proposed to solve MOPs. This study has focused on MOSI-based metaheuristics. In general, the MOSI-based algorithms integrate a single objective optimization algorithm with a MOO approach to handle multiple objectives. In most of the MOSI-based algorithms, the non-dominated solutions are evenly distributed along the Pareto front. However, in MOO, normally, the DM is interested in a particular region of the Pareto front. In this case, the solutions outside the region of interest can be considered as noisy data.

One of the recently proposed reference-point-based MOSI algorithms is the HHMO algorithm. This algorithm is inspired by the social hierarchy and hunting behaviour of the Harris's hawk predator in nature. The HHMO algorithm is developed based on the MOGWO and GWO algorithms, which show superior performance compared to other SI and MOSI-based algorithms. Furthermore, the integration of the preference information with the HHMO algorithm helps in directing the search to the region of interest. In this way, the algorithm, can ignore the calculation of other solutions and only needs to process the optimal solution in the region of interest. Thereby the performance of the algorithm is improved, and computational cost reduced, which

helps decision makers to make better decisions. However, the HHMO algorithm has drawbacks which degrade its performance.

The population of hawks is updated based on the position of the first three best solutions and it does not consider the positions of other hawks. This leads to the loss of population diversity. In this case, the HHMO algorithm will not be able to escape from local optima, especially in solving complex MOPs which leads to premature convergence. To overcome this limitation, a new population update strategy has been proposed. The proposed strategy was integrated with the HHMO to produce the NDSHHMO algorithm. The performance of the NDSHHMO has been evaluated and compared with the HHMO using different MOPs. Results of the experiments showed that the NDSHHMO algorithm outperforms the HHMO algorithm in terms of the convergence toward the Pareto front and diversity of the obtained solutions.

In the HHMO, the exploration and exploitation processes are equally performed according to the value of convergence parameter, *a*. The adjustment strategy of the convergence parameter overlooks the balanced trade-off between exploration and exploitation. This leads to poor approximation of the Pareto front and low diversity of non-dominated solutions. In this study, a new parameter adjustment strategy for the convergence parameter has been proposed. The proposed strategy was integrated with the HHMO to produce the EHHMO algorithm. This algorithm has been evaluated using a set of MOPs and compared with the HHMO. Experimental results showed that the adaptive parameter adjustment strategy can effectively improve the global search

ability of the EHHMO algorithm. This leads to improved convergence toward the Pareto front with better diversity of non-dominated solutions.

The initial population of hawks is generated using the random number generator method. The solution produced by this method is not well distributed in the search space which leads to poor convergence toward the Pareto front. To improve the initial population generator method in the HHMO, a new two-step method has been proposed. This method was integrated with the HHMO to produce the 2S-HHMO algorithm. This algorithm was evaluated by solving MOPs and the performance were compared with the other different initial population generator methods, namely RNG method in the original HHMO, OBL-HHMO, Sobol-HHMO, LHS-HHMO and Hammersley-HHMO algorithms. The experiment results demonstrated that the 2S-HHMO provides a competitive performance compared to the HHMO and other algorithms. This implies that the proposed two-step initial population method can effectively improve the performance of the HHMO algorithm in terms of both convergence and diversity compared to the uniform random number generator method.

This study has focused on improving the performance of the HHMO algorithm by solving these drawbacks. These improvements mainly include the following three aspects: updating the population of hawks, adjusting the convergence parameter value and the initial population of hawks. The results in Chapter Five proved that integrating the proposed population update strategy, convergence parameter adjustment strategy and two-step initial population generator method with HHMO led to better

performance in terms of convergence toward the Pareto font and diversity of nondominated solutions.

6.1 Research Contributions

This study has introduced the 2S-ENDSHHMO algorithm, which is an improved version of the HHMO algorithm in terms of the convergence toward the Pareto front and diversity of the obtained non-dominated solutions. The main contributions of the study are those of a new population update strategy, a new convergence parameter adjustment strategy and a new initial population generator method.

In the first contribution, the update population strategy is introduced. This strategy includes two stages. In the first stage, a new position of hawks is calculated using the new proposed movement strategy which is inspired by the flush-and-ambush technique of Harris's hawks in nature. In this technique, the position of hawks is updated with respect to the positions of all hawks in the population. This leads to an improvement in the hawks' exploration. In the second stage, the non-dominated sorting approach is used to select the best hawks to be used in the next generation, which helped in maintaining population diversity and prevent the premature convergence.

In the second contribution, the transition between exploration and exploitation processes is improved by using the proposed parameter adjustment strategy of the convergence parameter, *a*. In this strategy, the convergence parameter is adaptively

change based on the state of the search space. Furthermore, this strategy allows the algorithm to spend more time on exploration in the early stages of the optimization process and intensify the search within a particular promising region in the later stage.

In the third contribution, the two-step initial population method is proposed to initialize the population of hawks. This method consists of two steps. In the first step, the R-sequence is used to generate quasi-random numbers. In the second step, the partial OBL concept is applied to improve the diversity of the worst half in the population of hawks.

The proposed population update strategy, convergence parameter adjustment strategy and two-step initial population generator method are integrated to produce the 2S-ENDSHHMO algorithm. This algorithm has been tested to solve different well-known MOPs. The experiment results showed that the 2S-ENDSHHMO algorithm is superior compared to other state-of-the-art MOSI-based algorithms. Finally, the proposed 2S-ENDSHHMO algorithm has been verified by solving engineering MOPs. The experiment results showed that the 2S-ENDSHHMO algorithm outperformed other algorithms in terms of convergence toward the Pareto front and diversity of solutions.

6.2 Limitation and Future Work

During the development of this thesis some future lines of research have emerged. These lines remain open and are expected to be addressed in the future. Some are more

directly related to this study and others are more general lines. These lines can be summarised in the following points.

i. The 2S-ENDSHHMO algorithm is mainly developed to work with the continuous search space. It is possible to integrate it with an appropriate discretization method to solve combinatorial MOPs.

ii. The Euclidean distance is used in 2S-ENDSHHMO algorithm as a distance measure. However, there are many other distance measures, which need to be investigated, such as harmonic and Manhattan distances.

iii. In the 2S-ENDSHHMO algorithm, the best three solutions (leaders) plays an important role in guiding the hawks in the search space. Therefore, it will be useful to apply another technique to selection the leaders.

v. The 2S-ENDSHHMO algorithm has been used to solve constraint optimization problems with two objectives. It is possible to apply it to solve constraint optimization problems with many objectives.

iv. In the 2S-ENDSHHMO algorithm, the Pareto dominance approach has been used to compare the solutions and select the non-dominated solutions. It would be interesting to explore the possibility of comparing the solutions using different approach.

Based on the results and discussion presented in Chapter Five, the proposed the 2S-ENDSHHMO is proven to be more superior in terms of convergence and diversity as compared to the other MOSI-based algorithms. This indicates that the 2S-ENDSHHMO possess significant implication to the problem of interest. However, according to no-free-lunch theorem there is no a single algorithm that provides good performance in solving all optimization problems. Therefore, it is possible to explore other MOPs in the future and improve the 2S-ENDSHHMO limitations.

REFERENCES

AbdelAziz, A. M., Soliman, T. H. A., Ghany, K. K. A., & Sewisy, A. A. E.-M. (2019). A pareto-based hybrid whale optimization algorithm with tabu search for multi-objective optimization. *Algorithms, 12*(12), 261.

Abdou, W., & Bloch, C. (2020). Trade-off between diversity and convergence in multi-objective genetic algorithms. In *Proceeding of:* (pp. 37-50). Cham: Springer International Publishing.

Acı, Ç. İ., & Gülcan, H. (2019). A modified dragonfly optimization algorithm for single-and multiobjective problems using brownian motion. *Computational intelligence and neuroscience, 2019*.

Adekanmbi, O., & Green, P. (2015). Conceptual comparison of population based metaheuristics for engineering problems. *The Scientific World Journal, 2015*.

Aguiar e Oliveira Junior, H. (2016). The Many Aspects of Global Optimization. In *Evolutionary Global Optimization, Manifolds and Applications* (pp. 3-12). Cham: Springer International Publishing.

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation, 2*, 39-52.

Akbari, R., & Ziarati, K. (2011). A rank based particle swarm optimization algorithm with dynamic adaptation. *Journal of Computational and Applied Mathematics, 235*(8), 2694-2714.

Al-Betar, M. A., Awadallah, M. A., Faris, H., Aljarah, I., & Hammouri, A. I. (2018). Natural selection methods for grey wolf optimizer. *Expert Systems with Applications, 113*, 481-498.

Al Moubayed, N., Petrovski, A., & McCall, J. (2010). A novel smart multi-objective particle swarm optimisation using decomposition. In *Proceeding of International Conference on Parallel Problem Solving from Nature:* (pp. 1-10). Springer.

Al Moubayed, N., Petrovski, A., & McCall, J. (2014). D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary computation, 22*(1), 47-77.

Aleti, A., & Moser, I. (2016). A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Computing Surveys (CSUR), 49*(3), 56.

Alhammadi, H. Y., & Romagnoli, J. A. (2004). Incorporating environmental, profitability, heat integration and controllability considerations. *The integration of process design and control, 17*, 264.

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., & Mirjalili, S. (2020). Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowledge and Information Systems, 62*(2), 507-539.

Allawi, Z. T., Ibraheem, I. K., & Humaidi, A. J. (2019). Fine-tuning meta-heuristic algorithm for global optimization. *Processes, 7*(10), 657.

Allmendinger, R., Li, X., & Branke, J. (2008). Reference point-based particle swarm optimization using a steady-state approach. In *Proceeding of Asia-Pacific Conference on Simulated Evolution and Learning:* (pp. 200-209). Springer.

Altinoz, O. T., Yilmaz, A. E., & Weber, G.-W. (2014). Improvement of the gravitational search algorithm by means of low-discrepancy sobol quasi random-number sequence based initialization. *Advances in Electrical and Computer Engineering, 14*(3), 55-63.

Amoshahy, M. J., Shamsi, M., & Sedaaghi, M. H. (2016). A novel flexible inertia weight particle swarm optimization algorithm. *PloS one, 11*(8), e0161558.

Anderson, P. A., & Bone, Q. (1980). Communication between individuals in salp chains. II. Physiology. *Proc. R. Soc. Lond. B, 210*(1181), 559-574.

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

Asih, A. M. S., Sopha, B. M., & Kriptaniadewa, G. (2017). Comparison study of metaheuristics: Empirical application of delivery problems. *International Journal of Engineering Business Management, 9*, 1847979017743603.

Ayari, R., Nikdast, M., Hafnaoui, I., Beltrame, G., & Nicolescu, G. (2017). HypAp: a hypervolume-based approach for refining the design of embedded systems. *IEEE Embedded Systems Letters, 9*(3), 57-60.

Babazadeh, A., Poorzahedy, H., & Nikoosokhan, S. (2011). Application of particle swarm optimization to transportation network design problem. *Journal of King Saud University-Science, 23*(3), 293-300.

Back, T. (1996). Evolutionary Algorithms In Theory And Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms: Oxford university press. ISBN: 0195356705

Bai, J., & Liu, H. (2016). Multi-objective artificial bee algorithm based on decomposition by PBI method. *Applied Intelligence, 45*(4), 976-991.

Bairathi, D., & Gopalani, D. (2020). A Novel Swarm Intelligence Based Optimization Method: Harris' Hawk Optimization. In *Intelligent Systems Design and Applications* (Vol. 941, pp. 832-842). Cham: Springer.

Bangyal, W. H., Ahmad, J., Rauf, H. T., & Pervaiz, S. (2018). An improved bat algorithm based on novel initialization technique for global optimization problem. *International Journal of Advanced Computer Science and Applications (IJACSA), 9*(7), 158-166.

Bao, X., Jia, H., & Lang, C. (2019). Dragonfly algorithm with opposition-based learning for multilevel thresholding color image segmentation. *Symmetry, 11*(5), 716.

Barbosa, E. B. d. M., & Senne, E. L. F. (2017). Improving the fine-tuning of metaheuristics: an approach combining design of experiments and racing algorithms. *Journal of Optimization, 2017*.

Bassen, D. M., Vilkhovoy, M., Minot, M., Butcher, J. T., & Varner, J. D. (2017). JuPOETs: a constrained multiobjective optimization approach to estimate biochemical model ensembles in the Julia programming language. *BMC systems biology, 11*(1), 10.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). Nonlinear programming: theory and algorithms: John Wiley & Sons. ISBN: 1118626303

Bednarz, J. C. (1988). Cooperative hunting in Harris' hawks (Parabuteo unicinctus). *science, 239*(4847), 1525.

Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In P. Dario, G. Sandini, & P. Aebischer (Eds.), *Robots and Biological Systems: Towards a New Bionics?* (Vol. 102, pp. 703-712). Berlin, Heidelberg: Springer.

Berrouk, F., Bouchekara, H., Chaib, A., Abido, M., Bounaya, K., & Javaid, M. (2018). A new multi-objective Jaya algorithm for solving the optimal power flow problem. *Journal of Electrical Systems, 14*(3), 165-181.

Bhattacharya, M. (2016). Evolutionary landscape and management of population diversity. In *Combinations of Intelligent Methods and Applications* (pp. 1-18): Springer.

Bhowmik, A. R., & Chakraborty, A. K. (2015). Solution of optimal power flow using non dominated sorting multi objective opposition based gravitational search algorithm. *International journal of electrical power & energy systems, 64*, 1237-1250.

bin Mohd Zain, M. Z., Kanesan, J., Chuah, J. H., Dhanapal, S., & Kendall, G. (2018). A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization. *Applied Soft Computing, 70*, 680-700.

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

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

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

Bouchekara, H. (2014). Optimal power flow using black-hole-based optimization approach. *Applied Soft Computing, 24*, 879-888.

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

Brockhoff, D., Friedrich, T., & Neumann, F. (2008). Analyzing hypervolume indicator based algorithms. In *Proceeding of International Conference on Parallel Problem Solving from Nature:* (pp. 651-660). Springer.

Brück, A., Faßbender, S., & Waffenschmidt, E. (2018). Single- and multi-objective parameter optimization in a tool for designing PV-diesel-battery systems. In *Proceeding of 2018 7th International Energy and Sustainability Conference (IESC):* (pp. 1-5). IEEE. doi:10.1109/IESC.2018.8439998

Cai, X., Li, Y., Fan, Z., & Zhang, Q. (2015). An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on evolutionary computation, 19*(4), 508-523.

Carrese, R., & Li, X. (2015). Preference-Based Multiobjective Particle Swarm Optimization for Airfoil Design. In J. Kacprzyk & W. Pedrycz (Eds.), *Springer*

Handbook of Computational Intelligence (pp. 1311-1331). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cazzaniga, P., Nobile, M. S., & Besozzi, D. (2015). The impact of particles initialization in pso: Parameter estimation as a case in point. In *Proceeding of Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on:* (pp. 1-8). IEEE.

Chakraborty, K., & Chakrabarti, A. (2015). Soft computing techniques in voltage security analysis: Springer. ISBN: 8132223071

Chen, G., Qian, J., Zhang, Z., & Sun, Z. (2019). Multi-objective improved bat algorithm for optimizing fuel cost, emission and active power loss in power system. *IAENG International Journal of Computer Science, 46*(1), 118-133.

Chen, G., Yi, X., Zhang, Z., & Lei, H. (2018). Solving the multi-objective optimal power flow problem using the multi-objective firefly algorithm with a constraints-prior pareto-domination approach. *Energies, 11*(12), 3438.

Chen, H., Zhu, X., Pedrycz, W., Yin, S., Wu, G., & Yan, H. (2018). PEA: Parallel evolutionary algorithm by separating convergence and diversity for large-scale multi-objective optimization. In *Proceeding of 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS):* (pp. 223-232). IEEE.

Cheng, T., Chen, M., Fleming, P. J., Yang, Z., & Gan, S. (2017). A novel hybrid teaching learning based multi-objective particle swarm optimization. *Neurocomputing, 222*, 11-25.

Chiandussi, G., Codegone, M., Ferrero, S., & Varesio, F. E. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications, 63*(5), 912-942.

Christie, R. (1993). Power System Test Case Archive, the IEEE 30-Bus Test System Data. Retrieved from the university of Washington website: https://labs.ece.uw.edu/pstca/pf30/pg_tca30bus.htm

Chu, S.-C., Tsai, P.-W., & Pan, J.-S. (2006). Cat swarm optimization. *Pacific Rim International Conference on Artificial Intelligence*, 854-858.

Coello, C. A. C., Brambila, S. G., Gamboa, J. F., Tapia, M. G. C., & Gómez, R. H. (2019). Evolutionary multiobjective optimization: Open research areas and some challenges lying ahead. *Complex & Intelligent Systems*, 1-16.

Coello, C. A. C., & Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines, 6*(2), 163-190.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary Algorithms For Solving Multi-Objective Problems (Vol. 5): Springer.

Coello, C. A. C., & Lechuga, M. S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceeding of Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on:* (pp. 1051-1056). doi:10.1109/CEC.2002.1004388

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

Coello, C. C., & Pulido, G. T. (2005). Multiobjective structural optimization using a microgenetic algorithm. *Structural and Multidisciplinary Optimization, 30*(5), 388-403.

Coello Coello, C. A. (2000). Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering Systems, 17*(4), 319-346.

Coello Coello, C. A., & Christiansen, A. D. (1999). MOSES: A multiobjective optimization tool for engineering design. *Engineering Optimization, 31*(3), 337-368.

Črepinšek, M., Liu, S.-H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR), 45*(3), 1-33.

Cruz-Chávez, M. A., & Martínez-Oropeza, A. (2016). Feasible initial population with genetic diversity for a population-based algorithm applied to the vehicle routing problem with time windows. *Mathematical Problems in Engineering, 2016*.

Cuevas, E., Echavarría, A., & Ramírez-Ortegón, M. A. (2014). An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. *Applied Intelligence, 40*(2), 256-272.

Dadhich, C., Sharma, N., & Sharma, H. (2017). Howling mechanism based grey wolf optimizer. In *Proceeding of 2017 International Conference on Computer, Communications and Electronics (Comptelix):* (pp. 344-349). doi:10.1109/COMPTELIX.2017.8003991

Dai, C., & Lei, X. (2018). A decomposition-based multiobjective evolutionary algorithm with adaptive weight adjustment. *Complexity, 2018*.

Dai, C., & Lei, X. (2019). A multiobjective brain storm optimization algorithm based on decomposition. *Complexity, 2019*.

Dai, C., Wang, Y., & Ye, M. (2015). A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences, 325*, 541-557.

Darwin, C. (1859). On the origin of species. In. London: John Murray.

Datta, R., & Regis, R. G. (2016). A surrogate-assisted evolution strategy for constrained multi-objective optimization. *Expert Systems with Applications, 57*, 270-284.

Dawson, J. W. (1988). The cooperative breeding system of the Harris' Hawk in Arizona.

de Moura Oliveira, P., Oliveira, J., & Cunha, J. B. (2017). Trends in gravitational search algorithm. In *Proceeding of International Symposium on Distributed Computing and Artificial Intelligence:* (pp. 270-277). Springer.

Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms (Vol. 16): John Wiley & Sons. ISBN: 047187339X

Deb, K. (2014). Multi-objective Optimization. In E. K. Burke & G. Kendall (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 403-449). Boston, MA: Springer US.

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

Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. In *Proceeding of Proceedings of the 8th annual conference on Genetic and evolutionary computation:* (pp. 1629-1636).

Deb, K., & Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceeding of Proceedings of the 8th annual conference on Genetic and evolutionary computation:* (pp. 635-642). ACM.

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

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization* (pp. 105-145): Springer.

DeBruyne, A. S., & Kaur, B. D. (2016). Harris's hawk multi-objective optimizer for reference point problems. In *Proceeding of Proceedings on the International Conference on Artificial Intelligence (ICAI):* (pp. 287). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

DeBruyne, S. (2018). *Bio-Inspired Evolutionary Algorithms for Multi-Objective Optimization Applied to Engineering Applications*. University of Toledo,

Dede, T., Grzywiński, M., & Venkata Rao, R. (2020). Jaya: A new meta-heuristic algorithm for the optimization of braced dome structures. In *Proceeding of Advanced Engineering Optimization Through Intelligent Techniques:* (pp. 13-20). Singapore: Springer Singapore. doi:10.1007/978-981-13-8196-6_2

Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., . . . Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation, 48*, 220-250.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3-18.

Dewangan, R. K., Shukla, A., & Godfrey, W. W. (2019). Three dimensional path planning using Grey wolf optimizer for UAVs. *Applied Intelligence, 49*(6), 2201-2217.

Díaz-Manríquez, A., Toscano, G., Barron-Zambrano, J. H., & Tello-Leal, E. (2016). R2-based multi/many-objective particle swarm optimization. *Computational intelligence and neuroscience, 2016*.

Digehsara, P. A., Chegini, S. N., Bagheri, A., & Roknsaraei, M. P. (2020). An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled Halton sequence. *Cogent Engineering, 7*(1), 1737383.

Ding, M., Chen, H., Lin, N., Jing, S., Liu, F., Liang, X., & Liu, W. (2017). Dynamic population artificial bee colony algorithm for multi-objective optimal power flow. *Saudi journal of biological sciences, 24*(3), 703-710.

Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*.

Dorigo, M., & Gambardella, L. M. (1996). A study of some properties of Ant-Q. In *Proceeding of International Conference on Parallel Problem Solving from Nature:* (pp. 656-665). Springer.

Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). Metaheuristics for Hard Optimization: Methods and Case Studies: Springer Science & Business Media. ISBN: 3540309667

Du, P., Wang, J., Hao, Y., Niu, T., & Yang, W. (2019). A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2. 5 and PM10 forecasting. *arXiv preprint arXiv:1905.13550*.

Ebeed, M., Kamel, S., & Jurado, F. (2018). Optimal power flow using recent optimization techniques. In *Classical and Recent Aspects of Power System Optimization* (pp. 157-183): Elsevier.

Eiben, A. E., Michalewicz, Z., Schoenauer, M., & Smith, J. E. (2007). Parameter Control In Evolutionary Algorithms. In *Parameter setting in evolutionary algorithms* (pp. 19-46): Springer.

Elarbi, M., Bechikh, S., Ben Said, L., & Datta, R. (2017). Multi-objective Optimization: Classical and Evolutionary Approaches. In S. Bechikh, R. Datta, & A. Gupta (Eds.), *Recent Advances in Evolutionary Multi-objective Optimization* (pp. 1-30). Cham: Springer International Publishing.

Emary, E., Zawbaa, H. M., & Grosan, C. (2017). Experienced gray wolf optimization through reinforcement learning and neural networks. *IEEE transactions on neural networks and learning systems, 29*(3), 681-694.

Emmerich, M. T., & Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing, 17*(3), 585-609.

Ergul, E. U., & Eminoglu, I. (2014). DOPGA: A new fitness assignment scheme for multi-objective evolutionary algorithms. *International Journal of Systems Science, 45*(3), 407-426.

Falehi, A. D. (2020). An innovative optimal RPO-FOSMC based on multi-objective grasshopper optimization algorithm for DFIG-based wind turbine to augment MPPT and FRT capabilities. *Chaos, Solitons & Fractals, 130*, 109407.

Falehi, A. D., & Rafiee, M. (2019). Optimal control of novel fuel cell-based DVR using ANFISC-MOSSA to increase FRT capability of DFIG-wind turbine. *Soft Computing, 23*(15), 6633-6655.

Fan, S.-K. S., Chang, J.-M., & Chuang, Y.-C. (2015). A new multi-objective particle swarm optimizer using empirical movement and diversified search strategies. *Engineering Optimization, 47*(6), 750-770.

Fan, Z., Wang, T., Cheng, Z., Li, G., & Gu, F. (2017). An improved multiobjective particle swarm optimization algorithm using minimum distance of point to line. *Shock and Vibration, 2017*.

Farhang-Mehr, A., & Azarm, S. (2002). Entropy-based multi-objective genetic algorithm for design optimization. *Structural and Multidisciplinary Optimization, 24*(5), 351-361.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications, 30*(2), 413-435.

Farnad, B., & Jafarian, A. (2018). A new nature-inspired hybrid algorithm with a penalty method to solve constrained problem. *International Journal of Computational Methods, 15*(08), 1850069.

Farooq, M. U., Ahmad, A., & Hameed, A. (2017). Opposition-based initialization and a modified pattern for Inertia Weight (IW) in PSO. In *Proceeding of INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on:* (pp. 96-101). IEEE.

Faure, H., Kritzer, P., & Pillichshammer, F. (2015). From van der Corput to modern constructions of sequences for quasi-Monte Carlo rules. *Indagationes Mathematicae, 26*(5), 760-822.

Feo, T. A., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters, 8*(2), 67-71.

Filatovas, E., Kurasova, O., Redondo, J., & Fernández, J. (2019). A reference point-based evolutionary algorithm for approximating regions of interest in multiobjective problems. *TOP*, 1-22.

Filatovas, E., Kurasova, O., & Sindhya, K. (2015). Synchronous R-NSGA-II: an extended preference-based evolutionary algorithm for multi-objective optimization. *Informatica, 26*(1), 33-50.

Filatovas, E., Lančinskas, A., Kurasova, O., & Žilinskas, J. (2017). A preference-based multi-objective evolutionary algorithm R-NSGA-II with stochastic local search. *Central European Journal of Operations Research, 25*(4), 859-878.

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*.

Frank, S., & Rebennack, S. (2016). An introduction to optimal power flow: Theory, formulation, and examples. *IIE Transactions, 48*(12), 1172-1197.

Ganchev, I., Garcia, N. M., Dobre, C., Mavromoustakis, C. X., & Goleva, R. (2019). Enhanced Living Environments: Algorithms, Architectures, Platforms, and Systems (Vol. 11369): Springer. ISBN: 3030107523

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

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures, 89*(23-24), 2325-2336.

Gandomi, A. H., 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.

Gao, W.-f., Huang, L.-l., Wang, J., Liu, S.-y., & Qin, C.-d. (2016). Enhanced artificial bee colony algorithm through differential evolution. *Applied Soft Computing, 48*, 137-150.

Gao, W.-f., Liu, S.-y., & Huang, L.-l. (2012). Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. *Communications in Nonlinear Science and Numerical Simulation, 17*(11), 4316-4327.

García, I. C., Coello, C. A. C., & Arias-Montano, A. (2014). Mopsohv: A new hypervolume-based multi-objective particle swarm optimizer. In *Proceeding*

of 2014 IEEE Congress on Evolutionary Computation (CEC): (pp. 266-273). IEEE.

Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing, 1*(3), 190-206.

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

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning, 3*(2), 95-99.

Got, A., Moussaoui, A., & Zouache, D. (2020). A guided population archive whale optimization algorithm for solving multiobjective optimization problems. *Expert Systems with Applications, 141*, 112972.

Guo, M., Wang, J., Zhu, L., Guo, S., & Xie, W. (2020). An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems. *IEEE Access, 8*, 69861-69893.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik, 2*(1), 84-90.

Hamdan, M., & Qudah, O. (2015). The Initialization of Evolutionary Multi-objective Optimization Algorithms. In *Proceeding of International Conference in Swarm Intelligence:* (pp. 495-504). Springer.

Han, F., Sun, Y.-W.-T., & Ling, Q.-H. (2018). An improved multiobjective quantum-behaved particle swarm optimization based on double search strategy and circular transposon mechanism. *Complexity, 2018*.

Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. *Information Sciences, 422*, 462-479.

Hansen, P., & Mladenović, N. (2018). Variable Neighborhood Search. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.), *Handbook of Heuristics* (pp. 759-787). Cham: Springer International Publishing.

Hassanzadeh, H. R., & Rouhani, M. (2010). A multi-objective gravitational search algorithm. In *Proceeding of Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on:* (pp. 7-12). IEEE.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems, 97*, 849-872.

Hemelrijk, C. (2005). Self-organisation and evolution of biological and social systems: Cambridge University Press. ISBN: 0521846552

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing, 2*(2), 88-105.

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence: MIT press. ISBN: 0262581116

Hoque, M. T., Hoque, M. K., & Sinha, A. (2019). Smart Coordination Approach for Power Management with Modern PEV Technology. In *Advances in Computer, Communication and Control* (pp. 59-73): Springer.

Hu, W., & Yen, G. G. (2013). Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on evolutionary computation, 19*(1), 1-18.

Hu, Z., Bao, Y., & Xiong, T. (2014). Partial opposition-based adaptive differential evolution algorithms: Evaluation on the CEC 2014 benchmark set for real-parameter optimization. In *Proceeding of 2014 IEEE Congress on Evolutionary Computation (CEC):* (pp. 2259-2265). IEEE.

Huang, C., Li, Y., & Yao, X. (2019). A Survey of Automatic Parameter Tuning Methods for Metaheuristics. *IEEE Transactions on evolutionary computation*.

Huang, F.-z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation, 186*(1), 340-356.

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

Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., & Vivas-Valencia, C. (2019). An introduction to multiobjective simulation optimization. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 29*(1), 7.

Hussain, A., & Muhammad, Y. S. (2019). Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. *Complex & Intelligent Systems*, 1-14.

Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications, 31*(11), 7665-7683.

Ibrahim, R. A., Elaziz, M. A., & Lu, S. (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Systems with Applications, 108*, 1-27.

Isiet, M., & Gadala, M. (2019). Self-adapting control parameters in particle swarm optimization. *Applied Soft Computing, 83*, 105653.

Jacobs, D. S., & Bastian, A. (2017). Predator-prey Interactions: Co-evolution Between Bats and Their Prey: Springer. ISBN: 331932490X

Jain, P., & Saxena, A. (2019). An opposition theory enabled moth flame optimizer for strategic bidding in uniform spot energy market. *Engineering science and technology, an international journal, 22*(4), 1047-1067.

Jakob, W., & Blume, C. (2014). Pareto optimization or cascaded weighted sum: A comparison of concepts. *Algorithms, 7*(1), 166-185.

Jakubovski Filho, H. L., Ferreira, T. N., & Vergilio, S. R. (2019). Preference based multi-objective algorithms applied to the variability testing of software product lines. *Journal of Systems and Software, 151*, 194-209.

Jana, N. D., Das, S., & Sil, J. (2018). Landscape Characterization and Algorithms Selection for the PSP Problem. In *A Metaheuristic Approach to Protein Structure Prediction: Algorithms and Insights from Fitness Landscape Analysis* (pp. 87-150). Cham: Springer International Publishing.

Janga Reddy, M., & Nagesh Kumar, D. (2007). An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Engineering Optimization, 39*(1), 49-68.

Jangir, P., & Jangir, N. (2018). A new non-dominated sorting grey wolf optimizer (ns-gwo) algorithm: Development and application to solve engineering designs and economic constrained emission dispatch problem with integration of wind power. *Engineering Applications of Artificial Intelligence, 72*, 449-467.

Jangir, P., & Trivedi, I. N. (2018). Non-dominated sorting Moth Flame optimizer: A novel multi-objective optimization algorithm for solving engineering design problems. *Engineering Technology Open Access Journal*, 1-15.

Jia, C., & Zhu, H. (2017). An Improved Multiobjective Particle Swarm Optimization Based on Culture Algorithms. *Algorithms, 10*(2), 46.

Jia, H., Peng, X., Song, W., Lang, C., Xing, Z., & Sun, K. (2019). Multiverse optimization algorithm based on lévy flight improvement for multithreshold color image segmentation. *IEEE Access, 7*, 32805-32844.

Jiang, P., Li, R., & Li, H. (2019). Multi-objective algorithm for the design of prediction intervals for wind power forecasting model. *Applied Mathematical Modelling, 67*, 101-122.

Jiang, S., Wang, Y., & Ji, Z. (2014). Convergence analysis and performance of an improved gravitational search algorithm. *Applied Soft Computing, 24*, 363-384.

Joshi, H., & Arora, S. (2017). Enhanced grey wolf optimization algorithm for global optimization. *Fundamenta Informaticae, 153*(3), 235-264.

Julong, D. (1989). Introduction to grey system theory. *The Journal of grey system, 1*(1), 1-24.

Kahourzade, S., Mahmoudi, A., & Mokhlis, H. B. (2015). A comparative study of multi-objective optimal power flow based on particle swarm, evolutionary programming, and genetic algorithm. *Electrical Engineering, 97*(1), 1-12.

Kannan, B., & Kramer, S. N. (1994). An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design.

Kar, A. K. (2016). Bio inspired computing–A review of algorithms and scope of applications. *Expert Systems with Applications, 59*, 20-32.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Retrieved from

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*(1), 108-132.

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.

Kasdirin, H. A., Yahya, N., Aras, M., & Tokhi, M. (2017). Hybridizing invasive weed optimization with firefly algorithm for unconstrained and constrained optimization problems. *Journal of Theoretical and Applied Information Technology, 95*(4), 912-927.

Kaswan, K. S., Choudhary, S., & Sharma, K. (2015). Applications of artificial bee colony optimization technique: Survey. In *Proceeding of Computing for*

Sustainable Global Development (INDIACom), 2015 2nd International Conference on: (pp. 1660-1664). IEEE.
Kaveh, A., & Mahdavi, V. R. (2019). Multi-objective colliding bodies optimization algorithm for design of trusses. *Journal of computational design and engineering, 6*(1), 49-59.
Kazemzadeh Azad, S. (2018). Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. *Engineering Optimization, 50*(1), 89-105.
Kazimipour, B., Li, X., & Qin, A. K. (2013). Initialization methods for large scale global optimization. In *Proceeding of Evolutionary Computation (CEC), 2013 IEEE Congress on:* (pp. 2750-2757). IEEE.
Kazimipour, B., Li, X., & Qin, A. K. (2014). A review of population initialization techniques for evolutionary algorithms. In *Proceeding of Evolutionary Computation (CEC), 2014 IEEE Congress on:* (pp. 2585-2592). IEEE.
Keller, A. A. (2019). Multi-Objective Optimization in Theory and Practice II: Metaheuristic Algorithms: Bentham Science Publishers. ISBN: 1681087065
Kennedy, J., & Eberhar, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, 4*, 1942–1948.
Khan, T. A., & Li, W. (2017). Optimal design of plate-fin heat exchanger by combining multi-objective algorithms. *International journal of heat and mass transfer, 108*, 1560-1572.
Khanum, R. A., Jan, M. A., Aldegheishem, A., Mehmood, A., Alrajeh, N., & Khanan, A. (2019). Two New Improved Variants of Grey Wolf Optimizer for Unconstrained Optimization. *IEEE Access, 8*, 30805-30825.
Khatib, W., & Fleming, P. J. (1998). The stud GA: A mini revolution? In *Proceeding of:* (pp. 683-691). Berlin, Heidelberg: Springer Berlin Heidelberg.
Kheiri, F. (2018). A review on optimization methods applied in energy-efficient building geometry and envelope design. *Renewable and Sustainable Energy Reviews, 92*, 897-920.
Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220*(4598), 671-680.
Kishor, A., Singh, P. K., & Prakash, J. (2016). NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering. *Neurocomputing, 216*, 514-533.
Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of computational design and engineering, 5*(4), 458-472.
Kotinis, M. (2010). A particle swarm optimizer for constrained multi-objective engineering design problems. *Engineering Optimization, 42*(10), 907-926.
Kramer, O. (2010). Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evolutionary Intelligence, 3*(2), 51-65.
Kuang, F., Jin, Z., Xu, W., & Zhang, S. (2014). A novel chaotic artificial bee colony algorithm based on tent map. In *Proceeding of Evolutionary Computation (CEC), 2014 IEEE Congress on:* (pp. 235-241). IEEE.

Kulkarni, A. A. (2017). Using optimality theory and reference points to improve the diversity and convergence of a fuzzy-adaptive multi-objective particle swarm optimizer.

Kumar, Y., & Sahoo, G. (2016). A hybridise approach for data clustering based on cat swarm optimisation. *International Journal of Information and Communication Technology, 9*(1), 117-141.

Kumawat, I. R., Nanda, S. J., & Maddila, R. K. (2017). Multi-objective whale optimization. In *Proceeding of TENCON 2017-2017 IEEE Region 10 Conference:* (pp. 2747-2752). IEEE.

Kuo, R., & Huang, C. (2009). Application of particle swarm optimization algorithm for solving bi-level linear programming problem. *Computers & Mathematics with Applications, 58*(4), 678-685.

Kuo, R., & Zulvia, F. E. (2015). The gradient evolution algorithm: A new metaheuristic. *Information Sciences, 316*, 246-265.

Lampinen, J., & Zelinka, I. Mixed integer-discrete-continuous optimization by differential evolution, Part 2: A practical example. In *Proceeding of Proceedings of MENDEL:* (pp. 77-81).

Land, A. H., & Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497-520.

Larcher, G., & Niederreiter, H. (1993). Kronecker-type sequences and nonarchimedean diophantine approximations. *Acta Arithmetica, 63*(4), 379-396.

Larsen, R. B., Jouffroy, J., & Lassen, B. (2016). On the premature convergence of particle swarm optimization. In *Proceeding of Control Conference (ECC), 2016 European:* (pp. 1922-1927). IEEE.

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

Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research, 169*(3), 932-942.

Leung, M.-F., Ng, S.-C., Cheung, C.-C., & Lui, A. K. (2014). A new strategy for finding good local guides in MOPSO. In *Proceeding of 2014 IEEE Congress on Evolutionary Computation (CEC):* (pp. 1990-1997). IEEE.

Levy, G. (2002). An introduction to quasi-random numbers. *Numerical Algorithms Group Ltd., http://www. nag. co. uk/IndustryArticles/introduction_to_quasi_random_numbers. pdf (last accessed in April 10, 2012)*, 143.

Li, F., Liu, J., Tan, S., & Yu, X. (2015). R2-MOPSO: A multi-objective particle swarm optimizer based on R2-indicator and decomposition. In *Proceeding of 2015 IEEE Congress on Evolutionary Computation (CEC):* (pp. 3148-3155). IEEE.

Li, J.-q., Han, Y.-q., Duan, P.-y., Han, Y.-y., Niu, B., Li, C.-d., . . . Liu, Y.-p. (2020). Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems. *Journal of Cleaner Production, 250*, 119464.

Li, J., & Dong, N. (2017). Gravitational Search Algorithm with a new technique. In *Proceeding of Computational Intelligence and Security (CIS), 2017 13th International Conference on:* (pp. 516-519). IEEE.

Li, K., Chen, R., Min, G., & Yao, X. (2018a). Integration of preferences in decomposition multiobjective optimization. *IEEE transactions on cybernetics, 48*(12), 3359-3370.

Li, K., & Deb, K. (2016). Performance assessment for preference-based evolutionary multi-objective optimization using reference points. *COIN Report, 1*(1), 1-23.

Li, K., Deb, K., & Yao, X. (2018b). R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on evolutionary computation, 22*(6), 821-835.

Li, L., Yevseyeva, I., Basto-Fernandes, V., Trautmann, H., Jing, N., & Emmerich, M. (2016). An Ontology of Preference-Based Multiobjective Metaheuristics. *arXiv preprint arXiv:1609.08082*.

Li, Z., & Zheng, X. (2017). Review of design optimization methods for turbomachinery aerodynamics. *Progress in Aerospace Sciences, 93*, 1-23.

Li, Z., Zhou, Y., Zhang, S., & Song, J. (2016). Lévy-flight moth-flame algorithm for function optimization and engineering design problems. *Mathematical Problems in Engineering, 2016*.

Lin, Q., Li, J., Du, Z., Chen, J., & Ming, Z. (2015). A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research, 247*(3), 732-744.

Liu, H., Xu, G., Ding, G., & Li, D. (2015). Integrating opposition-based learning into the evolution equation of bare-bones particle swarm optimization. *Soft Computing, 19*(10), 2813-2836.

Liu, J., Li, F., Kong, X., & Huang, P. (2019). Handling many-objective optimisation problems with R2 indicator and decomposition-based particle swarm optimiser. *International Journal of Systems Science, 50*(2), 320-336.

Liu, R., Wang, R., Feng, W., Huang, J., & Jiao, L. (2016). Interactive reference region based multi-objective evolutionary algorithm through decomposition. *IEEE Access, 4*, 7331-7346.

Liu, T., Jiao, L., Ma, W., Ma, J., & Shang, R. (2016). A new quantum-behaved particle swarm optimization based on cultural evolution mechanism for multiobjective problems. *Knowledge-Based Systems, 101*, 90-99.

Liu, Y., Lu, H., Cheng, S., & Shi, Y. (2019). An Adaptive Online Parameter Control Algorithm for Particle Swarm Optimization Based on Reinforcement Learning. In *Proceeding of 2019 IEEE Congress on Evolutionary Computation (CEC):* (pp. 815-822). IEEE.

Lones, M. A. (2020). Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms. *SN Computer Science, 1*(1), 49.

Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence, 68*, 63-80.

Long, W., Wu, T., Cai, S., Liang, X., Jiao, J., & Xu, M. (2019). A novel grey wolf optimizer algorithm with refraction learning. *IEEE Access, 7*, 57805-57819.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2019). Iterated Local Search: Framework and Applications. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 129-168). Cham: Springer International Publishing.

Lu, Q., Ren, Y., Jin, H., Meng, L., Li, L., Zhang, C., & Sutherland, J. W. (2020). A hybrid metaheuristic algorithm for a profit-oriented and energy-efficient disassembly sequencing problem. *Robotics and Computer-Integrated Manufacturing, 61*, 101828.

Luo, J., Huang, X., Li, X., & Gao, K. (2019). A novel particle swarm optimizer for many-objective optimization. In *Proceeding of 2019 IEEE Congress on Evolutionary Computation (CEC):* (pp. 958-965). IEEE.

Luo, J., Liu, Q., Yang, Y., Li, X., Chen, M.-r., & Cao, W. (2017). An artificial bee colony algorithm for multi-objective optimisation. *Applied Soft Computing, 50*, 235-251.

Lv, L., Zhao, J., Wang, J., & Fan, T. (2019). Multi-objective firefly algorithm based on compensation factor and elite learning. *Future generation computer systems, 91*, 37-47.

Ma, R.-J., Yu, N.-Y., & Hu, J.-Y. (2013). Application of particle swarm optimization algorithm in the heating system planning problem. *The Scientific World Journal, 2013*.

Maaranen, H., Miettinen, K., & Mäkelä, M. M. (2004). Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications, 47*(12), 1885-1895.

Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. *Swarm and Evolutionary Computation, 39*, 1-23.

Mahmoodabadi, M., Bagheri, A., Nariman-Zadeh, N., & Jamali, A. (2012). A new optimization algorithm based on a combination of particle swarm optimization, convergence and divergence operators for single-objective and multi-objective problems. *Engineering Optimization, 44*(10), 1167-1186.

Mahmoodabadi, M. J., & Rezaee Babak, N. (2020). Fuzzy adaptive robust proportional–integral–derivative control optimized by the multi-objective grasshopper optimization algorithm for a nonlinear quadrotor. *Journal of Vibration and Control*, 1077546319901019.

Mahmoodabadi, M. J., & Shahangian, M. M. (2019). A new multi-objective artificial bee colony algorithm for optimal adaptive robust controller design. *IETE Journal of Research*, 1-14.

Majeed, M. M., & Patri, S. R. (2018). An enhanced grey wolf optimization algorithm with improved exploration ability for analog circuit design automation. *Turkish Journal of Electrical Engineering & Computer Sciences, 26*(5), 2605-2617.

Man-Im, A., Ongsakul, W., Singh, J., & Boonchuay, C. (2015). Multi-objective optimal power flow using stochastic weight trade-off chaotic NSPSO. In *Proceeding of 2015 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA):* (pp. 1-8). IEEE.

Marinakis, Y., Marinaki, M., & Migdalas, A. (2015). Adaptive Tunning of All Parameters in a Multi-Swarm Particle Swarm Optimization Algorithm: An

Application to the Probabilistic Traveling Salesman Problem. In *Optimization, Control, and Applications in the Information Age* (pp. 187-207): Springer.
McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*(2), 239-245.
Mellal, M. A., & Salhi, A. (2020). Parallel–Series System Optimization by Weighting Sum Methods and Nature-Inspired Computing. In N. Dey, A. S. Ashour, & S. Bhattacharyya (Eds.), *Applied Nature-Inspired Computing: Algorithms and Case Studies* (pp. 231-251). Singapore: Springer Singapore.
Mellal, M. A., & Zio, E. (2019). An adaptive particle swarm optimization method for multi-objective system reliability optimization. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 233*(6), 990-1001.
Meneghini, I. R., Alves, M. A., Gaspar-Cunha, A., & Guimarães, F. G. (2020). Scalable and customizable benchmark problems for many-objective optimization. *Applied Soft Computing, 90*, 106139.
Meng, Z., Shen, R., & Jiang, M. (2014). A penalty function algorithm with objective parameters and constraint penalty parameter for multi-objective programming. *American Journal of Operations Research, 4*(06), 331.
Mian, S. H., Umer, U., & Alkhalefah, H. (2019). Optimization of scanning parameters in coordinate metrology using grey relational analysis and fuzzy logic. *Mathematical Problems in Engineering, 2019*.
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., & Dong, J. S. (2020). Multi-objective Grey Wolf Optimizer. In *Multi-Objective Optimization using Artificial Intelligence Techniques* (pp. 47-58). Cham: Springer International Publishing.
Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163-191.
Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51-67.
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.
Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. d. S. (2016). Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Systems with Applications, 47*, 106-119.
Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence, 48*(4), 805-820.
Mishra, P., Das, D., Ukamanal, M., Routara, B., & Sahoo, A. (2015). Multi-response optimization of process parameters using Taguchi method and grey relational analysis during turning AA 7075/SiC composite in dry and spray cooling

environments. *International Journal of Industrial Engineering Computations, 6*(4), 445-456.

Mladenovic, N. (1995). A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization. In *Proceeding of papers presented at Optimization Days:* (pp.

Moghaddam, F. F., Moghaddam, R. F., & Cheriet, M. (2012). Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214*.

Mohamed, A.-A. A., El-Gaafary, A. A., Mohamed, Y. S., & Hemeida, A. M. (2016). Multi-objective modified grey wolf optimizer for optimal power flow. In *Proceeding of Power Systems Conference (MEPCON), 2016 Eighteenth International Middle East:* (pp. 982-990). IEEE. doi:10.1109/MEPCON.2016.7837016

Mohammadi, A., Omidvar, M. N., & Li, X. (2012). Reference point based multi-objective optimization through decomposition. In *Proceeding of Evolutionary Computation (CEC), 2012 IEEE Congress on:* (pp. 1-8). IEEE.

Mohammadi, A., Omidvar, M. N., & Li, X. (2013). A new performance metric for user-preference based multi-objective evolutionary algorithms. In *Proceeding of 2013 IEEE Congress on Evolutionary Computation:* (pp. 2825-2832). IEEE.

Mohan, N., Sivaraj, R., & Priya, R. D. (2016). A comprehensive review of bat algorithm and its applications to various optimization problems. *Asian Journal of Research in Social Sciences and Humanities, 6*(11), 676-690.

Momoh, J. A. (2015). Adaptive stochastic optimization techniques with applications: CRC Press. ISBN: 1439829799

Monsef, H., Naghashzadegan, M., Jamali, A., & Farmani, R. (2019). Comparison of evolutionary multi objective optimization algorithms in optimum design of water distribution network. *Ain Shams Engineering Journal, 10*(1), 103-111.

Mustaffa, Z., Sulaiman, M. H., & Yusof, Y. (2015). An application of grey wolf optimizer for commodity price forecasting. *Applied Mechanics & Materials, 785*.

Nenortaite, J., & Butleris, R. (2008). Application of particle swarm optimization algorithm to decision making model incorporating cluster analysis. In *Proceeding of 2008 Conference on Human System Interactions:* (pp. 88-93). doi:10.1109/HSI.2008.4581414

Neumann, A., Gao, W., Doerr, C., Neumann, F., & Wagner, M. (2018). Discrepancy-based evolutionary diversity optimization. In *Proceeding of Proceedings of the Genetic and Evolutionary Computation Conference:* (pp. 991-998).

Nezami, O. M., Bahrampour, A., & Jamshidlou, P. (2013). Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) algorithm for preventing from premature convergence. *Procedia Computer Science, 24*, 54-65.

Ng, K., Lee, C., Zhang, S., Wu, K., & Ho, W. (2017). A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion. *Computers & Industrial Engineering, 109*, 151-168.

Ni, K., Wang, J., Tang, G., & Wei, D. (2019). Research and application of a novel hybrid model based on a deep neural network for electricity load forecasting: A case study in australia. *Energies, 12*(13), 2467.

Ni, Q., & Deng, J. (2014). Analysis of population diversity of dynamic probabilistic particle swarm optimization algorithms. *Mathematical Problems in Engineering, 2014*.

Nie, X., Wang, W., & Nie, H. (2017). Chaos quantum-behaved cat swarm optimization algorithm and its application in the pv mppt. *Computational intelligence and neuroscience, 2017*.

Niu, B., Wang, H., Wang, J., & Tan, L. (2013). Multi-objective bacterial foraging optimization. *Neurocomputing, 116*, 336-345.

Niyomubyeyi, O., Sicuaio, T. E., González, J. I. D., Pilesjö, P., & Mansourian, A. (2020). A comparative study of four metaheuristic algorithms, amosa, moabc, mspso, and nsga-ii for evacuation planning. *Algorithms, 13*(1), 16.

Ochoa, G., Harvey, I., & Buxton, H. (2000). Optimal mutation rates and selection pressure in genetic algorithms. In *Proceeding of Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation:* (pp. 315-322). Citeseer.

Oliva, D., & Elaziz, M. A. (2020). An improved brainstorm optimization using chaotic opposite-based learning with disruption operator for global optimization and feature selection. *Soft Computing*.

Orouskhani, M., Teshnehlab, M., & Nekoui, M. A. (2018). EMCSO: An Elitist Multi-Objective Cat Swarm Optimization. *Journal of Optimization in Industrial Engineering, 11*(2), 107-117.

Owen, A. B. (2019). Monte Carlo Book: the Quasi-Monte Carlo parts.

Padhy, S., Panda, S., & Mahapatra, S. (2017). A modified GWO technique based cascade PI-PD controller for AGC of power systems in presence of Plug in Electric Vehicles. *Engineering science and technology, an international journal, 20*(2), 427-442.

Pan, A., Wang, L., Guo, W., & Wu, Q. (2018). A diversity enhanced multiobjective particle swarm optimization. *Information Sciences, 436*, 441-465.

Panda, A., Sahoo, A., & Rout, R. (2016). Multi-attribute decision making parametric optimization and modeling in hard turning using ceramic insert through grey relational analysis: A case study. *Decision Science Letters, 5*(4), 581-592.

Panda, G., Pradhan, P. M., & Majhi, B. (2011). IIR system identification using cat swarm optimization. *Expert Systems with Applications, 38*(10), 12671-12683.

Panniem, A., & Puphasuk, P. (2018). A modified artificial bee colony algorithm with firefly algorithm strategy for continuous optimization problems. *Journal of Applied Mathematics, 2018*.

Pardo, D., Möller, L., Neunert, M., Winkler, A. W., & Buchli, J. (2016). Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *IEEE Robotics and Automation Letters, 1*(2), 946-953.

Parpinelli, R. S., Plichoski, G., Da Silva, R. S., & Narloch, P. H. (2019). A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms. *IJBIC, 13*(1), 1-20.

Parsopoulos, K. E., & Vrahatis, M. N. (2005). Unified particle swarm optimization for solving constrained engineering optimization problems. In *Proceeding of International conference on natural computation:* (pp. 582-591). Springer.

Pavlov, D., & Kasumyan, A. (2000). Patterns and mechanisms of schooling behavior in fish: a review. *Journal of Ichthyology, 40*(2), S163.

Peng, W., & Zhang, Q. (2008). A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *Proceeding of Granular Computing, 2008. GrC 2008. IEEE International Conference on:* (pp. 534-537). IEEE. doi:10.1109/GRC.2008.4664724

Pharr, M., Jakob, W., & Humphreys, G. (2016). Physically based rendering: From theory to implementation: Morgan Kaufmann. ISBN: 0128007095

Pluhacek, M., Senkerik, R., Viktorin, A., Kadavy, T., & Zelinka, I. (2017). A review of real-world applications of particle swarm optimization algorithm. In *Proceeding of International Conference on Advanced Engineering Theory and Applications:* (pp. 115-122). Springer.

Poles, S., Fu, Y., & Rigoni, E. (2009). The effect of initial population sampling on the convergence of multi-objective genetic algorithms. In *Multiobjective programming and goal programming* (pp. 123-133): Springer.

Prakash, S., Trivedi, V., & Ramteke, M. (2016). An elitist non-dominated sorting bat algorithm NSBAT-II for multi-objective optimization of phthalic anhydride reactor. *International Journal of System Assurance Engineering and Management, 7*(3), 299-315.

Precup, R.-E., David, R.-C., Szedlak-Stinean, A.-I., Petriu, E. M., & Dragan, F. (2017). An Easily Understandable Grey Wolf Optimizer and Its Application to Fuzzy Controller Tuning. *Algorithms, 10*(2), 68.

Prugel-Bennett, A. (2010). Benefits of a population: five mechanisms that advantage population-based algorithms. *IEEE Transactions on evolutionary computation, 14*(4), 500-517.

Qi, Y., Li, X., Yu, J., & Miao, Q. (2019). User-preference based decomposition in MOEA/D without using an ideal point. *Swarm and Evolutionary Computation, 44*, 597-611.

Qiao, J., Zhou, H., & Yang, C. (2018). Bare-bones multiobjective particle swarm optimization based on parallel cell balanceable fitness estimation. *IEEE Access, 6*, 32493-32506.

Radulescu, A., López-Ibánez, M., & Stützle, T. (2013). Automatically improving the anytime behaviour of multiobjective evolutionary algorithms. In *Proceeding of International Conference on Evolutionary Multi-Criterion Optimization:* (pp. 825-840). Springer.

Ragsdell, K., & Phillips, D. (1976). Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry, 98*(3), 1021-1025.

Rahman, C. M., & Rashid, T. A. (2020). A Survey on Dragonfly Algorithm and its Applications in Engineering.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences, 183*(1), 1-15.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences, 179*(13), 2232-2248.

Rashid, M., & Baig, A. R. (2010). Improved opposition-based PSO for feedforward neural network training. In *Proceeding of Information Science and Applications (ICISA), 2010 International Conference on:* (pp. 1-6). IEEE.

Rechenberg, I. (1989). Evolution Strategy: Nature's Way of Optimization. In H. W. Bergmann (Ed.), *Optimization: Methods and Applications, Possibilities and Limitations: Proceedings of an International Seminar Organized by Deutsche Forschungsanstalt für Luft- und Raumfahrt, Bonn, June 1989* (pp. 106-126). Berlin, Heidelberg: Springer Berlin Heidelberg.

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

Roberts, M. (2018). The unreasonable effectiveness of quasirandom sequences. http://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/.

Roque, L. A., Fontes, D. B., & Fontes, F. A. (2017). A Metaheuristic Approach to the Multi-Objective Unit Commitment Problem Combining Economic and Environmental Criteria. *Energies, 10*(12), 2029.

Sadatsakkak, S. A., Ahmadi, M. H., Bayat, R., Pourkiaei, S. M., & Feidt, M. (2015). Optimization density power and thermal efficiency of an endoreversible Braysson cycle by using non-dominated sorting genetic algorithm. *Energy conversion and management, 93*, 31-39.

Sahib, M. A., Abdulnabi, A. R., & Mohammed, M. A. (2018). Improving bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution. *Alexandria engineering journal, 57*(4), 3341-3349.

Santosa, B., & Ningrum, M. K. (2009). Cat swarm optimization for clustering. In *Proceeding of Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of:* (pp. 54-59). IEEE.

Sapre, S., & Mini, S. (2020). Moth flame optimization algorithm based on decomposition for placement of relay nodes in WSNs. *Wireless Networks, 26*(2), 1473-1492.

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

Savsani, V. (2014). Implementation of modified artificial bee colony (ABC) optimization technique for minimum cost design of welded structures. *International Journal for Simulation and Multidisciplinary Design Optimization, 5*, A11.

Savsani, V., & Tawhid, M. A. (2017). Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Engineering Applications of Artificial Intelligence, 63*, 20-32.

Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. *Journal of Experimental & Theoretical Artificial Intelligence, 30*(2), 293-317.

Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, 1-20.

Schwefel, H.-P. P. (1993). Evolution and Optimum Seeking: The Sixth Generation: John Wiley \& Sons, Inc. ISBN: 0471571482

Sedarous, S., El-Gokhy, S. M., & Sallam, E. (2018). Multi-swarm multi-objective optimization based on a hybrid strategy. *Alexandria engineering journal, 57*(3), 1619-1629.

Shao, Q., Weng, S.-S., Liou, J. J., Lo, H.-W., & Jiang, H. (2019). Developing a sustainable urban-environmental quality evaluation system in china based on a hybrid model. *International journal of environmental research and public health, 16*(8), 1434.

Si, T., & Dutta, R. (2019). Partial Opposition-Based Particle Swarm Optimizer in ArtiŻcial Neural Network Training for Medical Data ClassiŻcation. *International Journal of Information Technology & Decision Making, 18*(5), 1717-1750.

Siavash, M., Pfeifer, C., Rahiminejad, A., & Vahidi, B. (2017). An application of grey wolf optimizer for optimal power flow of wind integrated power systems. In *Proceeding of Electric Power Engineering (EPE), 2017 18th International Scientific Conference on:* (pp. 1-6). IEEE.

Sierra, M. R., & Coello, C. A. C. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and ∈-dominance. In *Proceeding of International Conference on Evolutionary Multi-Criterion Optimization:* (pp. 505-519). Springer.

Simon, D. (2008). Biogeography-Based Optimization. *IEEE Transactions on evolutionary computation, 12*(6), 702-713.

Škrinjarić, T., & Šego, B. (2019). Using grey incidence analysis approach in portfolio selection. *International Journal of Financial Studies, 7*(1), 1.

Slater, P. B. (2019). Extensions of generalized two-qubit separability probability analyses to higher dimensions, additional measures and new methodologies. *Quantum Information Processing, 18*(4), 121.

Slowik, A., & Kwasnicka, H. (2018). Nature inspired methods and their industry applications—swarm intelligence algorithms. *IEEE Transactions on Industrial Informatics, 14*(3), 1004-1015.

Sobol', I. y. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 7*(4), 784-802.

Song, Z., Liu, B., & Cheng, H. (2019). Adaptive particle swarm optimization with population diversity control and its application in tandem blade optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233*(6), 1859-1875.

Sörensen, K., Sevaux, M., & Glover, F. (2018). A History of Metaheuristics. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.), *Handbook of Heuristics* (pp. 791-808). Cham: Springer International Publishing.

Stadler, W., & Dauer, J. (1993). Multicriteria optimization in engineering: A tutorial and survey. *Structural optimization: Status and promise, 150*.

Stanger-Hall, K. F., Lloyd, J. E., & Hillis, D. M. (2007). Phylogeny of north american fireflies (coleoptera: Lampyridae): Implications for the evolution of light signals. *Molecular phylogenetics and evolution, 45*(1), 33-49.

Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., . . . Di Stefano, D. (2008). Real-World Applications of Multiobjective Optimization. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (pp. 285-327). Berlin, Heidelberg: Springer Berlin Heidelberg.

Stojanović, I., Brajević, I., Stanimirović, P. S., Kazakovtsev, L. A., & Zdravev, Z. (2017). Application of heuristic and metaheuristic algorithms in solving constrained weber problem with feasible region bounded by arcs. *Mathematical Problems in Engineering, 2017*.

Stützle, T. (1998). Local search algorithms for combinatorial problems. *Darmstadt University of Technology PhD Thesis, 20*.

Stützle, T., López-Ibáñez, M., Pellegrini, P., Maur, M., Montes de Oca, M., Birattari, M., & Dorigo, M. (2012). Parameter adaptation in ant colony optimization. In Y. Hamadi, E. Monfroy, & F. Saubion (Eds.), *Autonomous Search* (pp. 191-215). Berlin, Heidelberg: Springer Berlin Heidelberg.

Su, W., Chen, H., Liu, F., Lin, N., Jing, S., Liang, X., & Liu, W. (2017). A novel comprehensive learning artificial bee colony optimizer for dynamic optimization biological problems. *Saudi journal of biological sciences, 24*(3), 695-702.

Sun, Y., & Gao, Y. (2019). A multi-objective particle swarm optimization algorithm based on gaussian mutation and an improved learning strategy. *Mathematics, 7*(2), 148.

Talbi, E.-G. (2009). Metaheuristics: From Design To Implementation (Vol. 74). Hoboken, NJ: John Wiley & Sons. ISBN: 978-0-470-27858-1

Talbi, E.-G. (2013). A Unified Taxonomy of Hybrid Metaheuristics with Mathematical Programming, Constraint Programming and Machine Learning. In E.-G. Talbi (Ed.), *Hybrid Metaheuristics* (pp. 3-76). Berlin, Heidelberg: Springer Berlin Heidelberg.

Tamura, K., & Gallagher, M. (2019). Quantitative measure of nonconvexity for black-box continuous functions. *Information Sciences, 476*, 64-82.

Tan, Y., Lu, X., Liu, Y., Wang, Q., & Zhang, H. (2019). Decomposition-based multiobjective optimization with invasive weed colonies. *Mathematical Problems in Engineering, 2019*.

Tanabe, R., & Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 106078.

Tang, R., Li, K., Ding, W., Wang, Y., Zhou, H., & Fu, G. (2020). Reference point based multi-objective optimization of reservoir operation: A comparison of three algorithms. *Water resources management*, 1-16.

Tawhid, M. A., & Savsani, V. (2018a). ∈-constraint heat transfer search (∈-HTS) algorithm for solving multi-objective engineering design problems. *Journal of computational design and engineering, 5*(1), 104-119.

Tawhid, M. A., & Savsani, V. (2018b). A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems. *Applied Intelligence, 48*(10), 3762-3781.

Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation, 17*(3), 411-436.

Tian, D. (2017). Particle swarm optimization with chaos-based initialization for numerical optimization. *Intelligent Automation & Soft Computing*, 1-12.

Tian, Y., Wang, H., Zhang, X., & Jin, Y. (2017). Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization. *Complex & Intelligent Systems, 3*(4), 247-263.

Tian, Y., Zhang, X., Cheng, R., & Jin, Y. (2016). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *Proceeding of Evolutionary Computation (CEC), 2016 IEEE Congress on:* (pp. 5222-5229). IEEE.

Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In *Proceeding of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06):* (pp. 695-701). IEEE.

Tomassetti, G. (2010). A cost-effective algorithm for the solution of engineering problems with particle swarm optimization. *Engineering Optimization, 42*(5), 471-495.

Torbaghan, S. S., & Gibescu, M. (2017). Optimum transmission system expansion offshore considering renewable energy sources. In *Optimization in Renewable Energy Systems* (pp. 177-231): Elsevier.

Tsai, C.-W., Chiang, M.-C., Ksentini, A., & Chen, M. (2016). Metaheuristic algorithms for healthcare: Open issues and challenges. *Computers & Electrical Engineering, 53*, 421-434.

Tsai, C.-W., Huang, Y.-T., & Chiang, M.-C. (2014). A non-dominated sorting firefly algorithm for multi-objective optimization. In *Proceeding of 2014 14th International Conference on Intelligent Systems Design and Applications:* (pp. 62-67). IEEE. doi:10.1109/ISDA.2014.7066269

Tu, Q., Chen, X., & Liu, X. (2019). Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection. *IEEE Access, 7*, 78012-78028.

Tuba, M., Bacanin, N., & Stanarevic, N. (2012). Adjusted artificial bee colony (ABC) algorithm for engineering problems. *WSEAS Transaction on Computers, 11*(4), 111-120.

Ünal, A. N., & Kayakutlu, G. (2020). Multi-objective particle swarm optimization with random immigrants. *Complex & Intelligent Systems*.

Vachhani, V. L., Dabhi, V. K., & Prajapati, H. B. (2016). Improving NSGA-II for solving multi objective function optimization problems. In *Proceeding of 2016 International Conference on Computer Communication and Informatics (ICCCI):* (pp. 1-6). IEEE. doi:10.1109/ICCCI.2016.7479921

van der Plas, C. C., Tervonen, T. T., & Dekker, R. R. (2012). *Evaluation of scalarization methods and NSGA-II/SPEA2 genetic algorithms for multi-*

objective optimization of green supply chain design (1566-7294). Retrieved from

Vasuki, A. (2020). Nature Inspired Optimization for Image Processing.

Verma, O. P., Aggarwal, D., & Patodi, T. (2016). Opposition and dimensional based modified firefly algorithm. *Expert Systems with Applications, 44*, 168-176.

Vikram, K. A., Ratnam, C., Lakshmi, V., Kumar, A. S., & Ramakanth, R. (2018). Application of dragonfly algorithm for optimal performance analysis of process parameters in turn-mill operations-A case study. In *Proceeding of IOP Conference Series: Materials Science and Engineering:* (pp. 012154). IOP Publishing.

Vira, C., & Haimes, Y. Y. (1983). Multiobjective decision making: theory and methodology. In *North Holland series in system science and engineering*: North-Holland.

Voudouris, C. (1997). *Guided local search for combinatorial optimisation problems.* University of Essex,

Wang, G.-G., Deb, S., & Cui, Z. (2015). Monarch butterfly optimization. *Neural Computing and Applications*.

Wang, G.-G., & Tan, Y. (2017). Improving metaheuristic algorithms with information feedback models. *IEEE transactions on cybernetics, 49*(2), 542-555.

Wang, H., Jin, Y., & Yao, X. (2016). Diversity assessment in many-objective optimization. *IEEE transactions on cybernetics, 47*(6), 1510-1522.

Wang, J.-S., & Li, S.-X. (2019). An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific reports, 9*(1), 1-21.

Wang, J. (2015). A new cat swarm optimization with adaptive parameter control. In *Genetic and Evolutionary Computing* (pp. 69-78): Springer.

Wang, W. L., Li, W., & Wang, Y. L. (2019a). An Opposition-Based Evolutionary Algorithm for Many-Objective Optimization with Adaptive Clustering Mechanism. *Computational intelligence and neuroscience, 2019*.

Wang, W. L., Li, W. K., Wang, Z., & Li, L. (2019b). Opposition-based multi-objective whale optimization algorithm with global grid ranking. *Neurocomputing, 341*, 41-59.

Wang, Y., Liu, B., Ma, Z., Wong, K.-C., & Li, X. (2019). Nature-inspired multiobjective cancer subtype diagnosis. *IEEE journal of translational engineering in health and medicine, 7*, 1-12.

Wang, Y., & Sun, X. (2018). A many-objective optimization algorithm based on weight vector adjustment. *Computational intelligence and neuroscience, 2018*.

Wang, Z., & Rangaiah, G. P. (2017). Application and analysis of methods for selecting an optimal solution from the Pareto-optimal front obtained by multiobjective optimization. *Industrial & Engineering Chemistry Research, 56*(2), 560-574.

Weerasinghe, G., Chi, H., & Cao, Y. (2016). Particle swarm optimization simulation via optimal Halton sequences. *Procedia Computer Science, 80*, 772-781.

Wei, L.-X., Li, X., Fan, R., Sun, H., & Hu, Z.-Y. (2018). A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator. *IEEE Access, 6*, 14710-14721.

Wei, L., Li, X., & Fan, R. (2019). A new multi-objective particle swarm optimisation algorithm based on R2 indicator selection mechanism. *International Journal of Systems Science, 50*(10), 1920-1932.

Weiszer, M., Chen, J., Stewart, P., & Zhang, X. (2018). Preference-based evolutionary algorithm for airport surface operations. *Transportation Research Part C: Emerging Technologies, 91*, 296-316.

Wen, L. (2016). Grey wolf optimizer based on nonlinear adjustment control parameter. In *Proceeding of International Conference on Sensors, Mechatronics and Automation:* (pp. 643-648).

Wickramasinghe, U. K., Carrese, R., & Li, X. (2010). Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *Proceeding of IEEE congress on evolutionary computation:* (pp. 1-8). IEEE.

Wickramasinghe, U. K., & Li, X. (2008). Integrating user preferences with particle swarms for multi-objective optimization. In *Proceeding of Proceedings of the 10th annual conference on Genetic and evolutionary computation:* (pp. 745-752). ACM. doi:10.1145/1389095.1389237

Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In *Multiple criteria decision making theory and application* (pp. 468-486): Springer.

Wilcoxon, F. (1992). Individual Comparisons by Ranking Methods. In S. Kotz & N. L. Johnson (Eds.), *Breakthroughs in Statistics: Methodology and Distribution* (pp. 196-202). New York, NY: Springer New York.

Xia, X., Ji, J., Li, C.-s., Xue, X., Wang, X., & Zhang, C. (2019). Multiobjective optimal control for hydraulic turbine governing system based on an improved MOGWO algorithm. *Complexity, 2019*.

Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., & Liu, B. (2018). Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access*.

Xu, G., Wu, Z.-H., & Jiang, M.-Z. (2015). Premature convergence of standard particle swarm optimisation algorithm based on Markov chain analysis. *International Journal of Wireless and Mobile Computing, 9*(4), 377-382.

Yan, F., Xu, J., & Yun, K. (2019). Dynamically Dimensioned Search Grey Wolf Optimizer Based on Positional Interaction Information. *Complexity, 2019*.

Yang, C., Gao, W., Liu, N., & Song, C. (2015). Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight. *Applied Soft Computing, 29*, 386-394.

Yang, C., & Ji, J. (2016). Multiobjective bacterial foraging optimization using archive strategy. In *Proceeding of ICPRAM:* (pp. 185-192).

Yang, K., Li, L., Deutz, A., Back, T., & Emmerich, M. (2016). Preference-based multiobjective optimization using truncated expected hypervolume improvement. In *Proceeding of 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD):* (pp. 276-281). doi:10.1109/FSKD.2016.7603186

Yang, X.-S., Deb, S., Hanne, T., & He, X. (2019). Attraction and diffusion in nature-inspired optimization algorithms. *Neural Computing and Applications, 31*(7), 1987-1994.

Yang, X.-S., Karamanoglu, M., & He, X. (2014). Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering Optimization, 46*(9), 1222-1237.

Yang, X., You, G., Zhao, C., Dou, M., & Guo, X. (2019). An Improved multi-objective genetic algorithm based on orthogonal design and adaptive clustering pruning strategy. *arXiv preprint arXiv:1901.00577*.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Proceeding of Stochastic Algorithms: Foundations and Applications, SAGA 2009:* (pp. 169-178). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-04944-6_14

Yang, X. S. (2010a). Engineering optimization: an introduction with metaheuristic applications. Hoboken, New Jersey: John Wiley & Sons. ISBN: 9780470582466

Yang, X. S. (2010b). Nature-inspired metaheuristic algorithms: Luniver press. ISBN: 1905986289

Yang, X. S. (2010c). A new metaheuristic bat-inspired algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature inspired cooperative strategies for optimization* (pp. 65-74). Berlin, Heidelberg: Springer.

Yang, X. S. (2012a). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation, 3*(5), 267-274.

Yang, X. S. (2012b). Flower pollination algorithm for global optimization. In *Proceeding of:* (pp. 240-249). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yang, X. S. (2012c). Swarm-based metaheuristic algorithms and no-free-lunch theorems. In P. Rafael & S. Heitor (Eds.), *Theory and New Applications of Swarm Intelligence.* (Vol. 9, pp. 1-16). Janeza Trdine: InTech.

Yang, X. S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with computers, 29*(2), 175-184.

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

Yang, X. S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation, 5*(3), 141-149.

Yang, X. S., Karamanoglu, M., & He, X. (2014b). Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering Optimization, 46*(9), 1222-1237.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on evolutionary computation, 3*(2), 82-102.

Yi, J., Huang, D., Fu, S., He, H., & Li, T. (2015). Multi-objective bacterial foraging optimization algorithm based on parallel cell entropy for aluminum electrolysis production process. *IEEE Transactions on Industrial Electronics, 63*(4), 2488-2500.

Yu, H., Wang, Y., & Xiao, S. (2020). Multi-objective particle swarm optimization based on cooperative hybrid strategy. *Applied Intelligence, 50*(1), 256-269.

Yu, X., & Zhang, X. (2017). Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems. *PloS one, 12*(2).

Zapotecas-Martínez, S., Coello, C. A. C., Aguirre, H. E., & Tanaka, K. (2018). A review of features and limitations of existing scalable multiobjective test suites. *IEEE Transactions on evolutionary computation, 23*(1), 130-142.

Zapotecas Martínez, S., & Coello Coello, C. A. (2011). A multi-objective particle swarm optimizer based on decomposition. In *Proceeding of Proceedings of the 13th annual conference on Genetic and evolutionary computation:* (pp. 69-76). ACM. doi:10.1145/2001576.2001587

Zellagui, M., Hassan, H. A., & Abdelaziz, A. Y. (2017). Non-dominated sorting gravitational search algorithm for multi-objective optimization of power transformer design. *Engineering Review, 37*(1), 27-37.

Zhang, J., Chen, W.-N., Zhan, Z.-H., Yu, W.-J., Li, Y.-L., Chen, N., & Zhou, Q. (2012). A survey on algorithm adaptation in evolutionary computation. *Frontiers of Electrical and Electronic Engineering, 7*(1), 16-31.

Zhang, L., Liu, L., Yang, X.-S., & Dai, Y. (2016). A novel hybrid firefly algorithm for global optimization. *PloS one, 11*(9).

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation, 11*(6), 712-731.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., & Tiwari, S. (2009). Multi-objective optimization test instances for the congress on evolutionary computation (CEC 2009) special session & competition. *Working Report CES-887. University of Essex, UK*.

Zhang, X., Zheng, X., Cheng, R., Qiu, J., & Jin, Y. (2018). A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Information Sciences, 427*, 63-76.

Zhang, Y., Zeng, B., Li, Y., & Li, J. (2018). A mullti-or many-objective evolutionary algorithm with global loop update. *arXiv preprint arXiv:1803.06282*.

Zhao, M., Wang, X., Yu, J., Bi, L., Xiao, Y., & Zhang, J. (2020). Optimization of Construction Duration and Schedule Robustness Based on Hybrid Grey Wolf Optimizer with Sine Cosine Algorithm. *Energies, 13*(1), 215.

Zheng, H., & Zhou, Y. (2013). A cooperative coevolutionary cuckoo search algorithm for optimization problem. *Journal of Applied Mathematics, 2013*.

Zhong, Y.-B., Xiang, Y., & Liu, H.-L. (2014). A multi-objective artificial bee colony algorithm based on division of the searching space. *Applied Intelligence, 41*(4), 987-1011.

Zhou, J., & Yao, X. (2017). Multi-objective hybrid artificial bee colony algorithm enhanced with Lévy flight and self-adaption for cloud manufacturing service composition. *Applied Intelligence, 47*(3), 721-742.

Zhou, Z., Dai, G., Fang, P., Chen, F., & Tan, Y. (2008). An improved hybrid multi-objective particle swarm optimization algorithm. In *Proceeding of International Symposium on Intelligence Computation and Applications:* (pp. 181-188). Springer.

Zhu, Q., Lin, Q., Chen, W., Wong, K.-C., Coello, C. A. C., Li, J., . . . Zhang, J. (2017). An external archive-guided multiobjective particle swarm optimization algorithm. *IEEE transactions on cybernetics, 47*(9), 2794-2808.

Zhu, X., Gao, Z., Du, Y., Cheng, S., & Xu, F. (2018). A decomposition-based multi-objective optimization approach considering multiple preferences with robust performance. *Applied Soft Computing, 73*, 263-282.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation, 8*(2), 173-195.

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Proceeding of International conference on parallel problem solving from nature:* (pp. 832-842). Springer. doi:10.1007/978-3-540-30217-9_84

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on evolutionary computation, 3*(4), 257-271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation, 7*(2), 117-132.

Zou, J., Fu, L., Yang, S., Zheng, J., Ruan, G., Pei, T., & Wang, L. (2019). An adaptation reference-point-based multiobjective evolutionary algorithm. *Information Sciences, 488*, 41-57.

Appendix A
Formulations and Characteristics of The Benchmark Functions

A.1 Formulations and Characteristics of The ZDT family.

Characteristics of ZDT problems

Problem	Pareto border	Formulation
ZDT1	Convex	$f_1(x) = x_1$ $f_2(x) = g(x) \times h(x)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g}$
ZDT2	Concave	$f_1(x) = x_1$ $f_2(x) = g(x) \times h(x)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - (f_1/g)^2$
ZDT3	disconnected	$f_1(x) = x_1$ $f_2 = g(x) \times h(x)$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g(x)} - (f_1/g(x))\sin(10\pi f_1)$
ZDT4	Multimodal	$f_1(x) = x_1$ $f_2(x) = g(x) \times h(x)$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{f_1/g}$
ZDT6	Concave	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x) \times h(x)$ $g(x) = 1 + 9\left[\left(\sum_{i=2}^{10} x_i\right)/9\right]^{0.25}$ $h(f_1, g) = 1 - (f_1/g)^2$

A.2 Formulations and Characteristics of The DTLZ family.

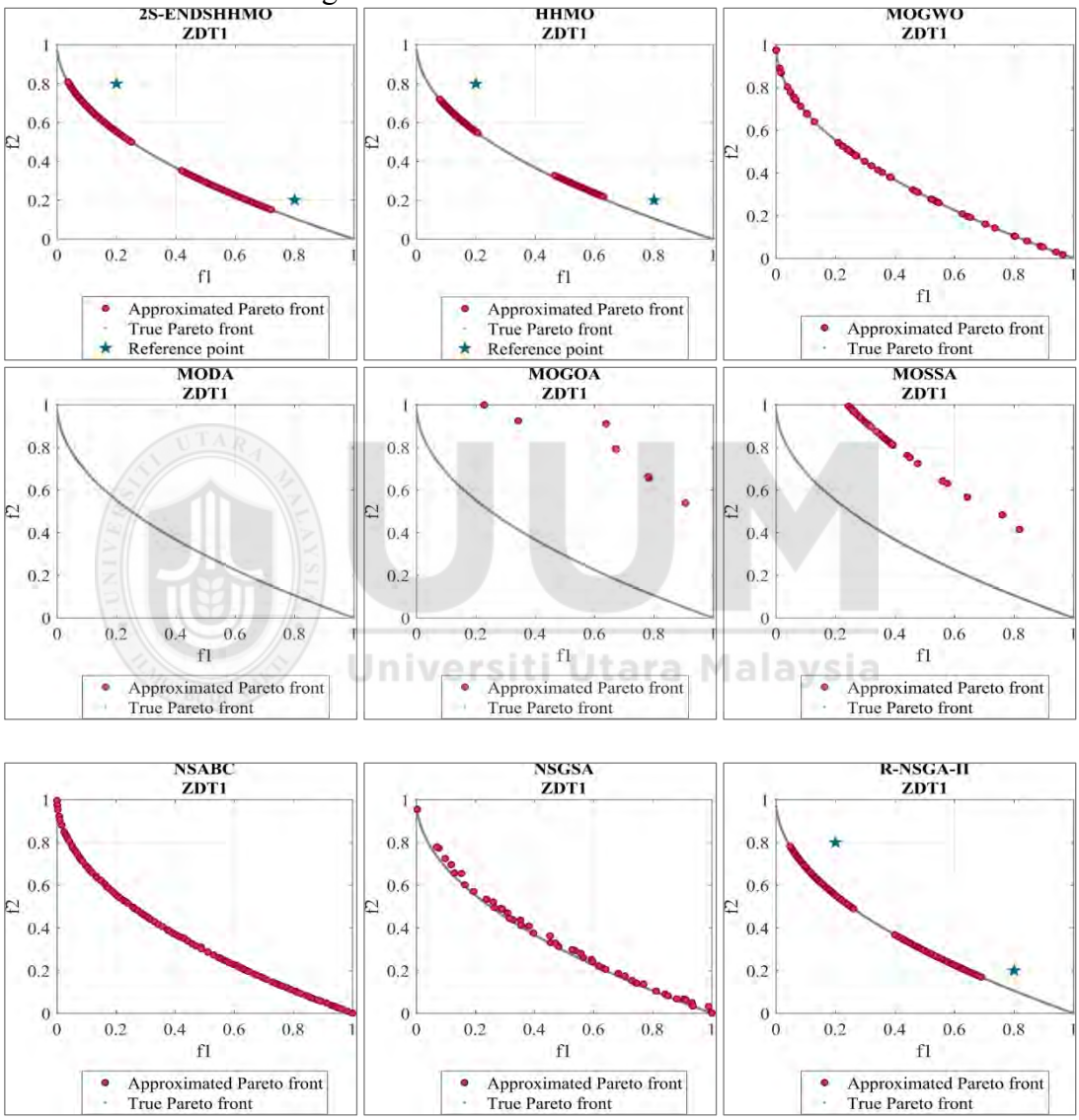
Characteristics of DTLZ problems

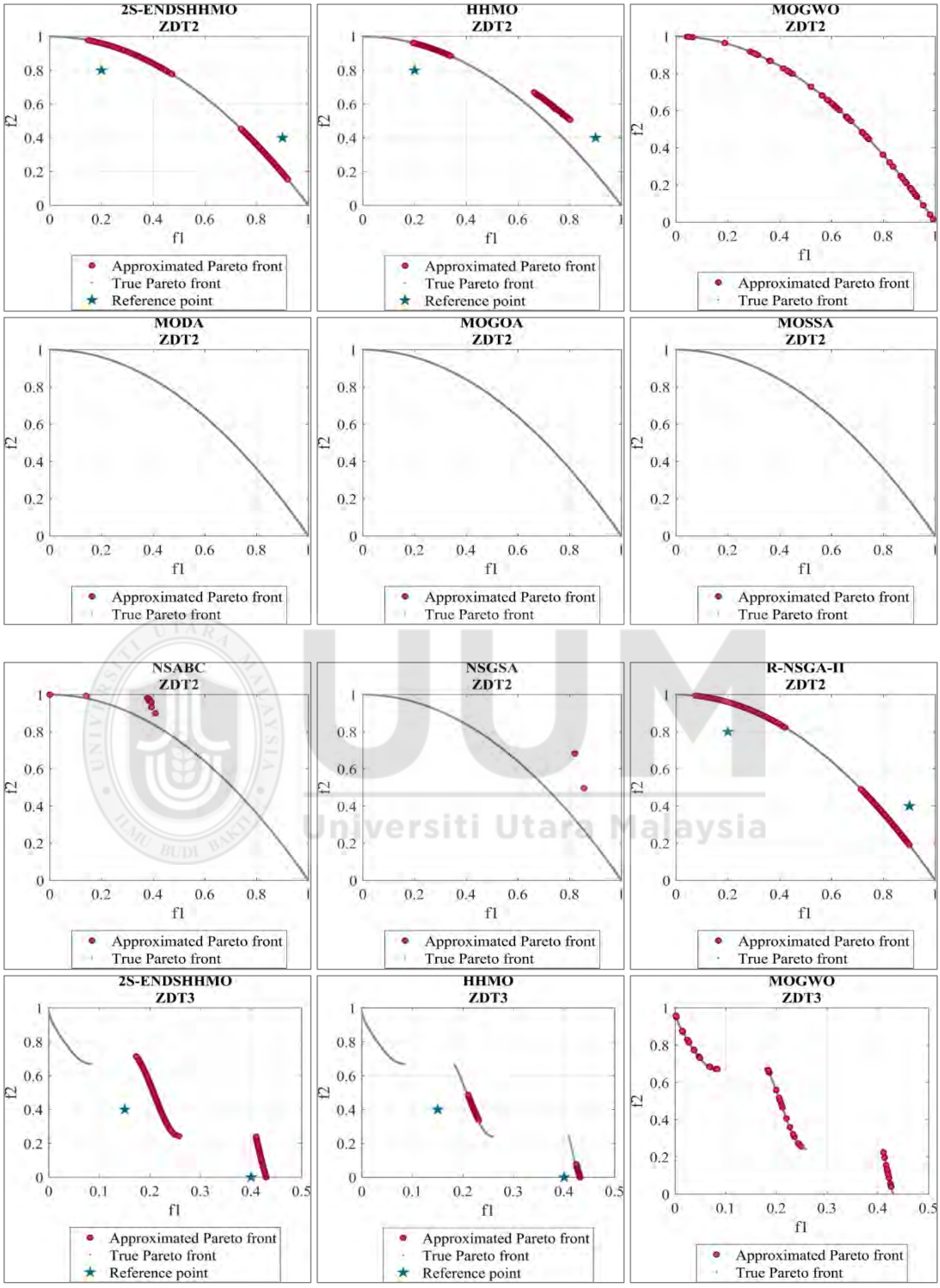
Problem	Pareto border	Formulation
DTLZ2	Concave	$f1(x) = 1 + g(x)\cos\left(x_1\frac{\pi}{2}\right)$ $f2(x) = 1 + g(x)\sin\left(x_1\frac{\pi}{2}\right)$

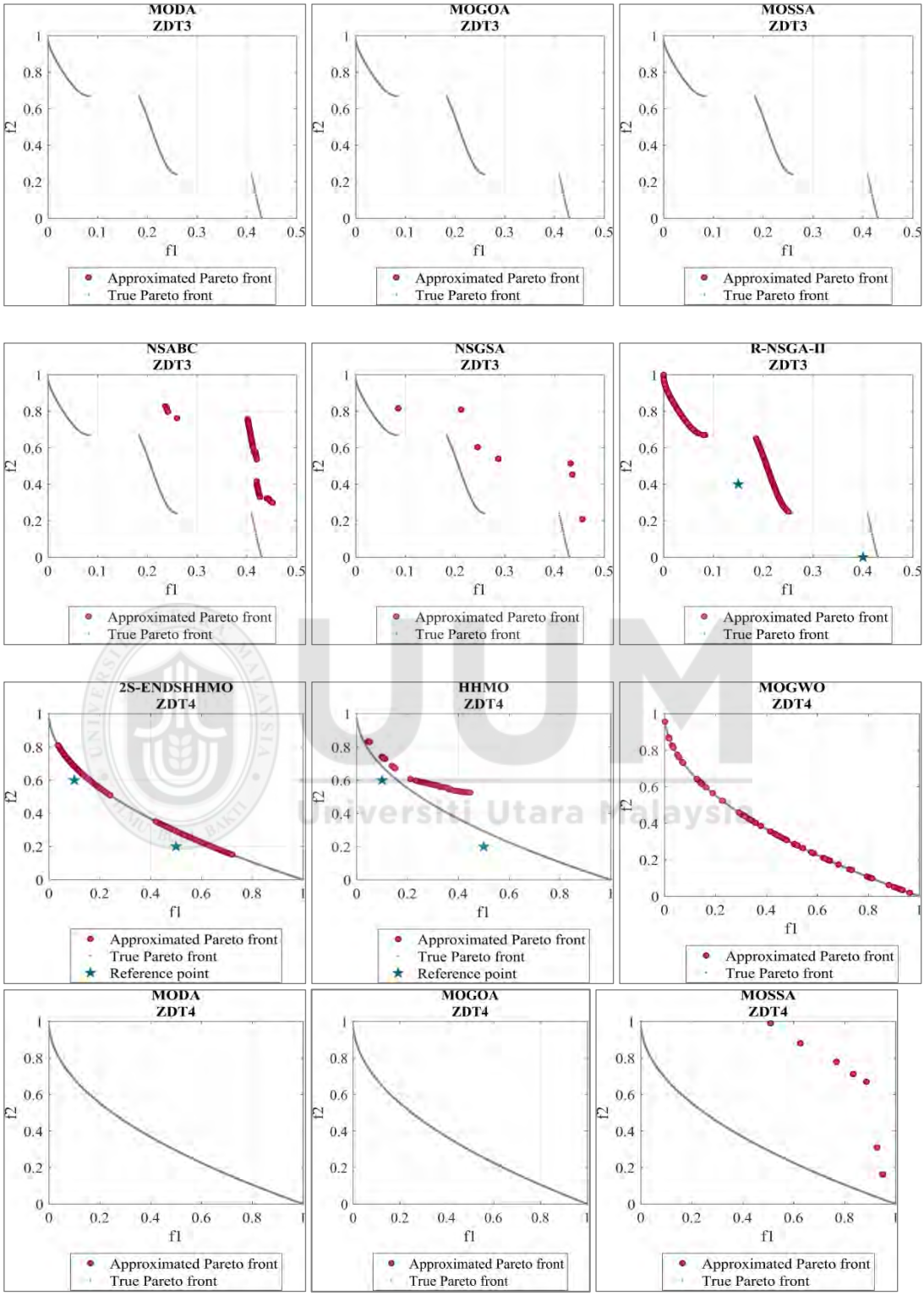
DTLZ4		$g(x) = \sum_{x_i \in x} (x_i - 0.5)^2$ $f1(x) = 1 + g(x) \cos\left(x_1^{\alpha} \frac{\pi}{2}\right)$ $f2(x) = 1 + g(x) \sin\left(x_1^{\alpha} \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} (x_i^{\alpha} - 0.5)^2$ $\alpha = 100$
DTLZ5	Degenerated	$f1(x) = 1 + g(x) \cos\left(\frac{1 + 2gx}{4(1 + g(x))} . \frac{\pi}{2}\right)$ $f2(x) = 1 + g(x) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{x_i \in x} (x_i - 0.5)^2$
DTLZ6	Degenerated	$f1(x) = 1 + g(x) \cos\left(\frac{1 + 2gx}{4(1 + g(x))} . \frac{\pi}{2}\right)$ $f2(x) = 1 + g(x) \sin\left(x_1 \frac{\pi}{2}\right)$ $g(x) = \sum_{i=m}^{n} x_i^{0.1}$
DTLZ7	Disconnected	$f1(x) = x_1$ $f2(x) = x_2$ $f_M(x) = 1 + g(x_M)\,).\, h(f_1, f_2, \dots, f_{M-1} g(x))$ $g(x) = 1 + \frac{9}{\lvert xM \rvert} \sum_{x_i \in x_m} x_i$ $\mathrm{h}(f_1, f_1, \dots, f_{M-1}, g) = \mathrm{M} - \sum_{i=1}^{M-1} \left(\frac{f_i}{\lvert 1 + g(x) \rvert} (1 + \sin(3\pi f_i))\right)$

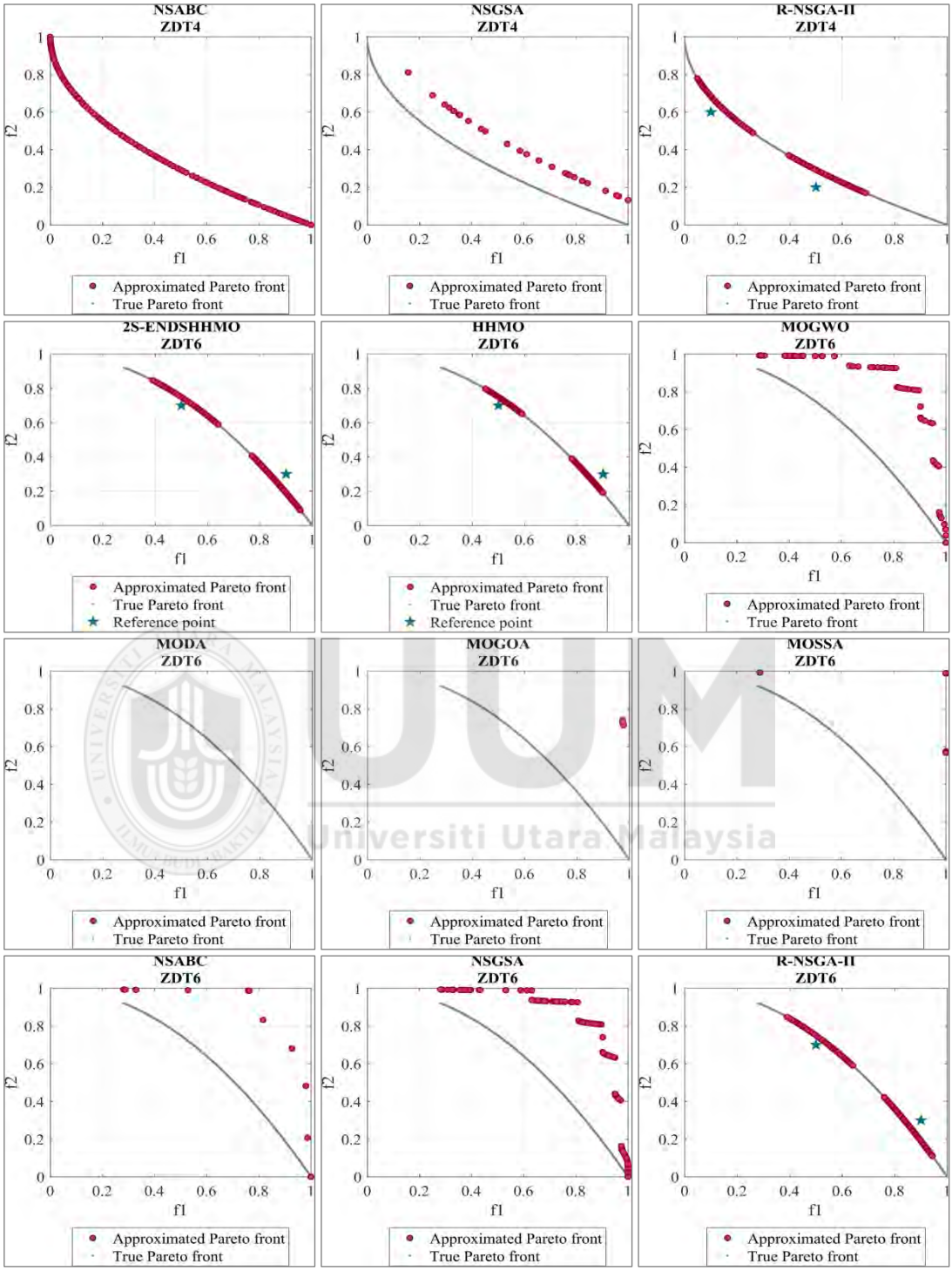
Appendix B
Distribution of the Non-dominated Solutions Obtained by MOSI-based Algorithms

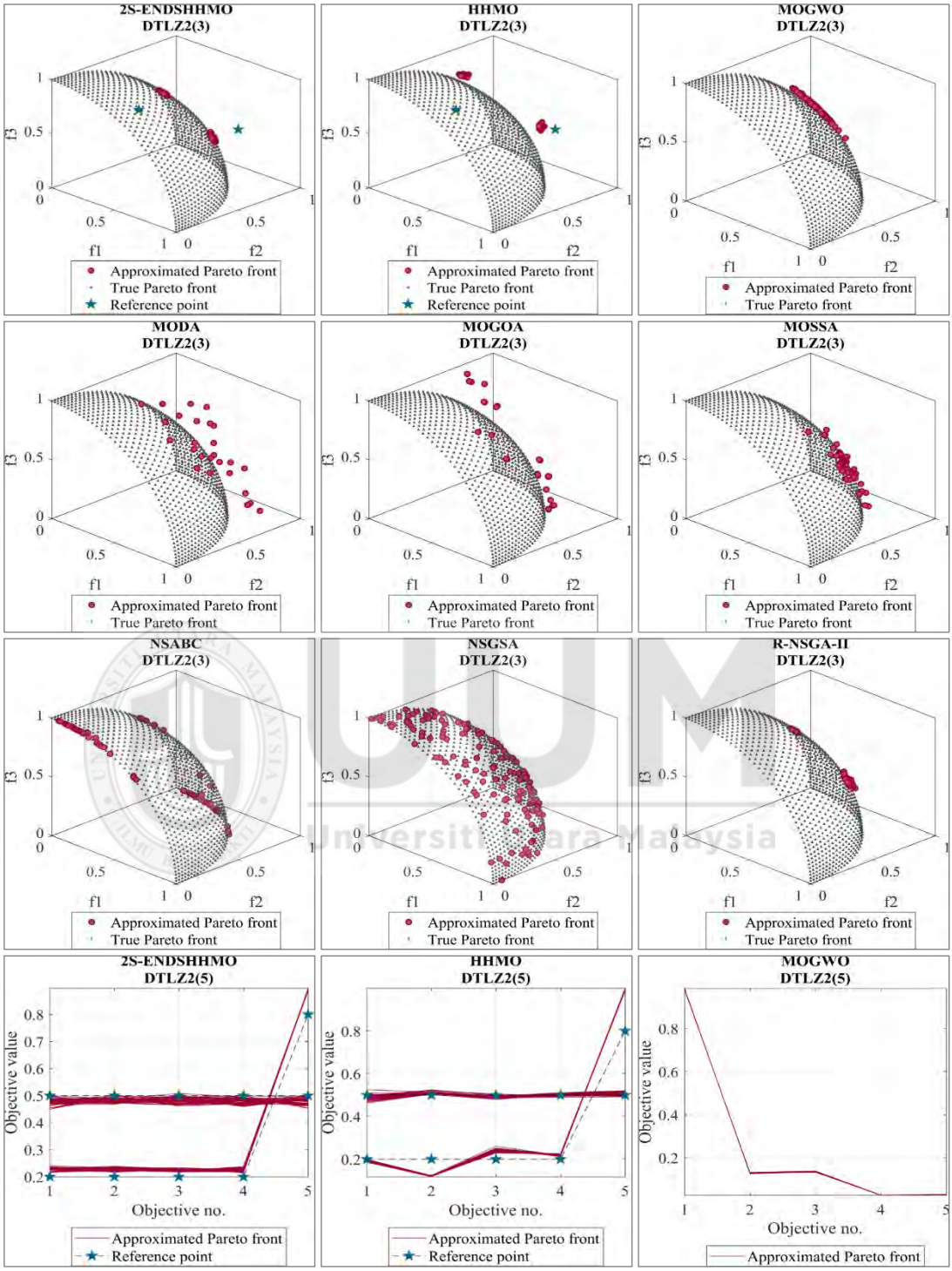
B.1: The distribution of the approximated Pareto front obtained by using 2S-ENDSHHMO, HHMO, MOGWO, MOGOA, MODA, MOSSA, NSGSA, NSABC and R-NSGA-II in solving the test functions

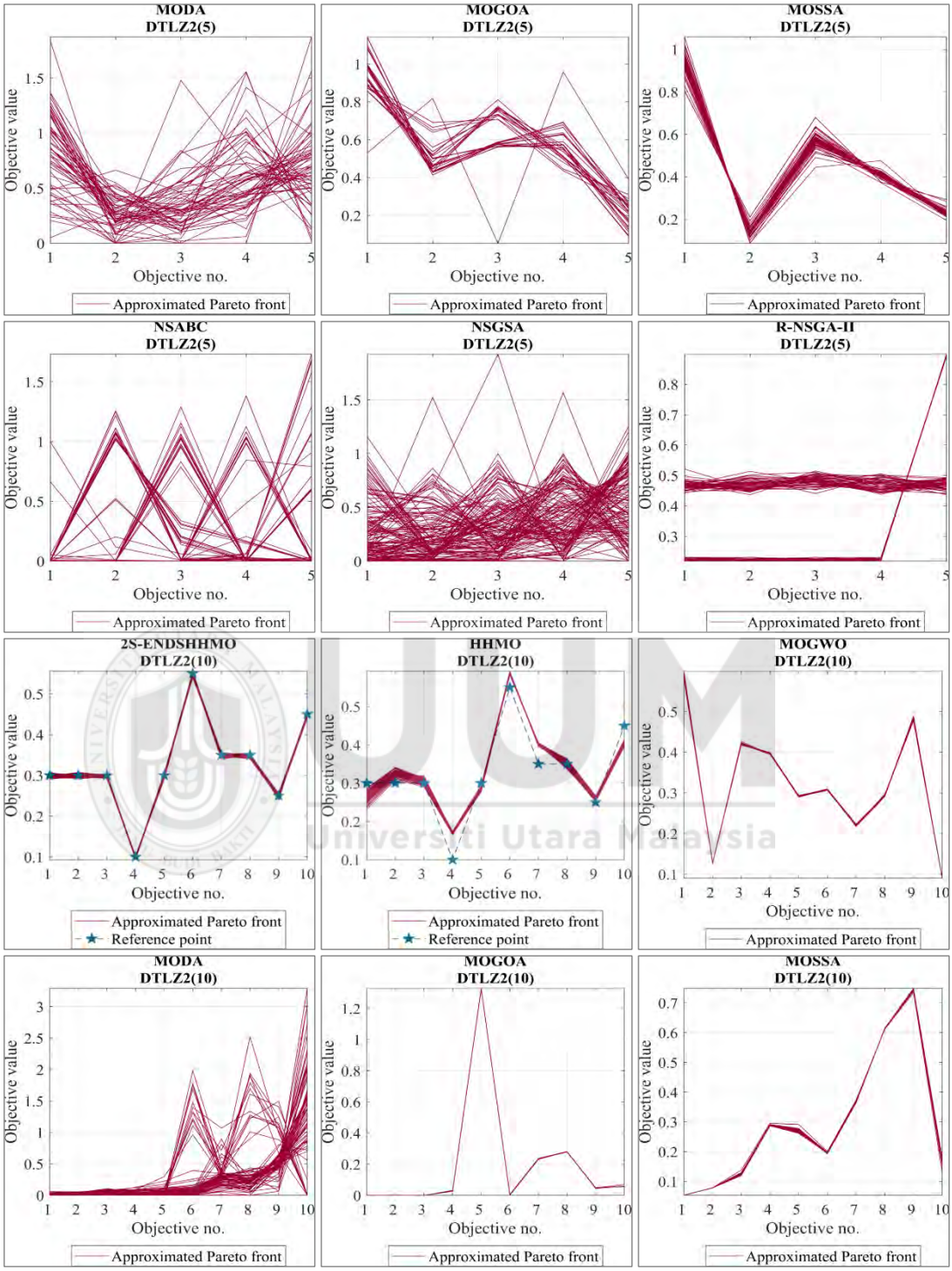


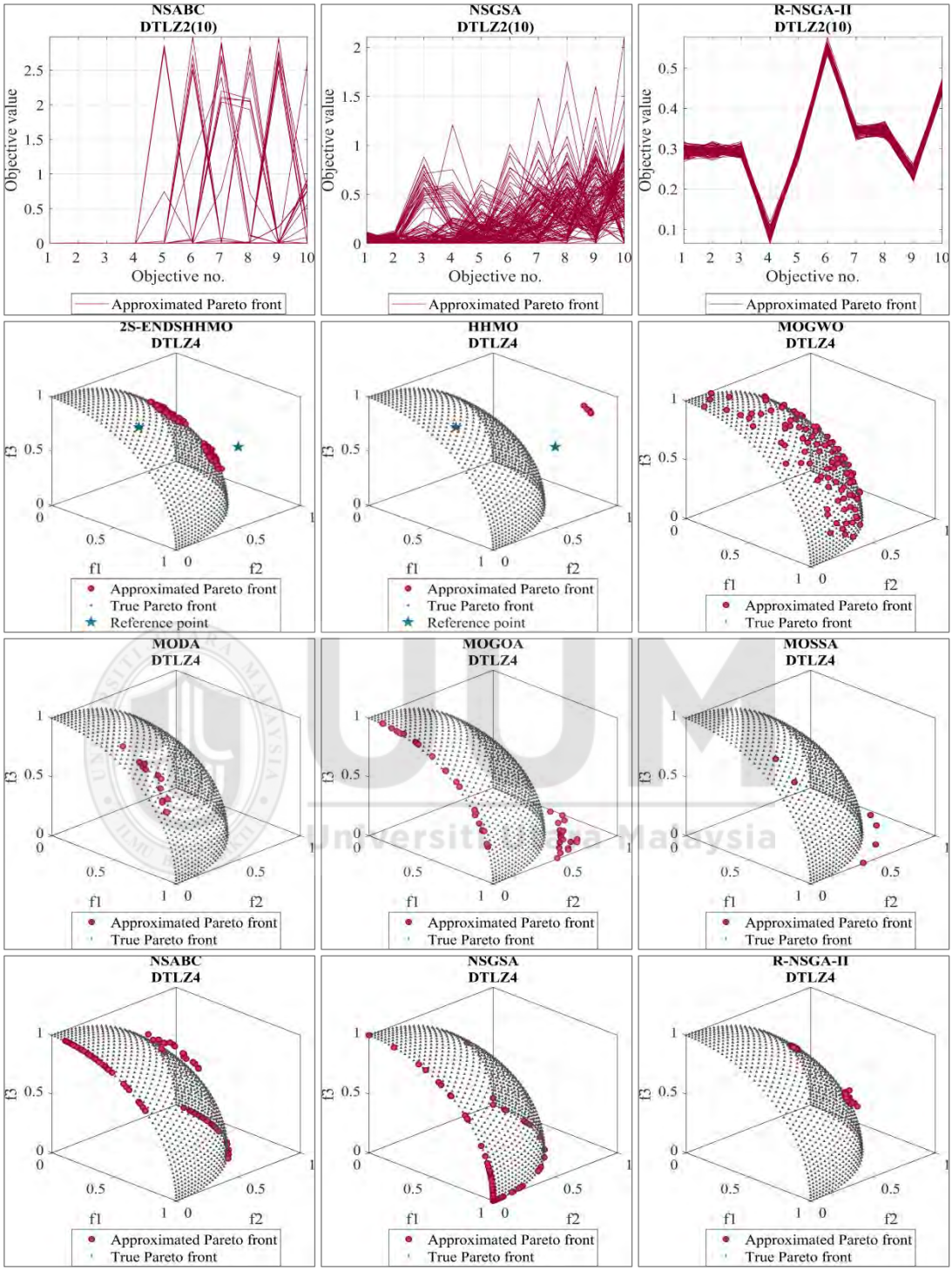


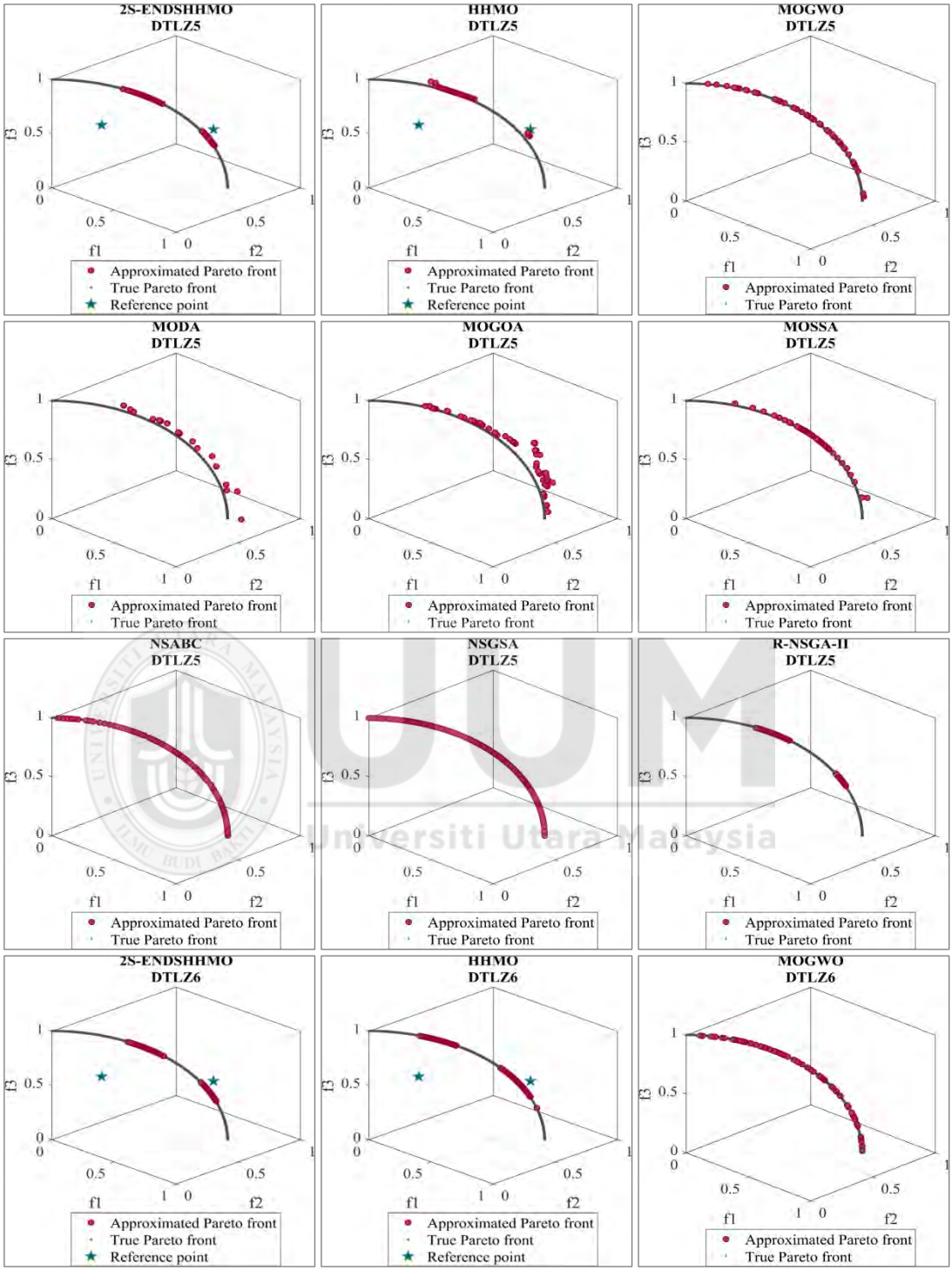


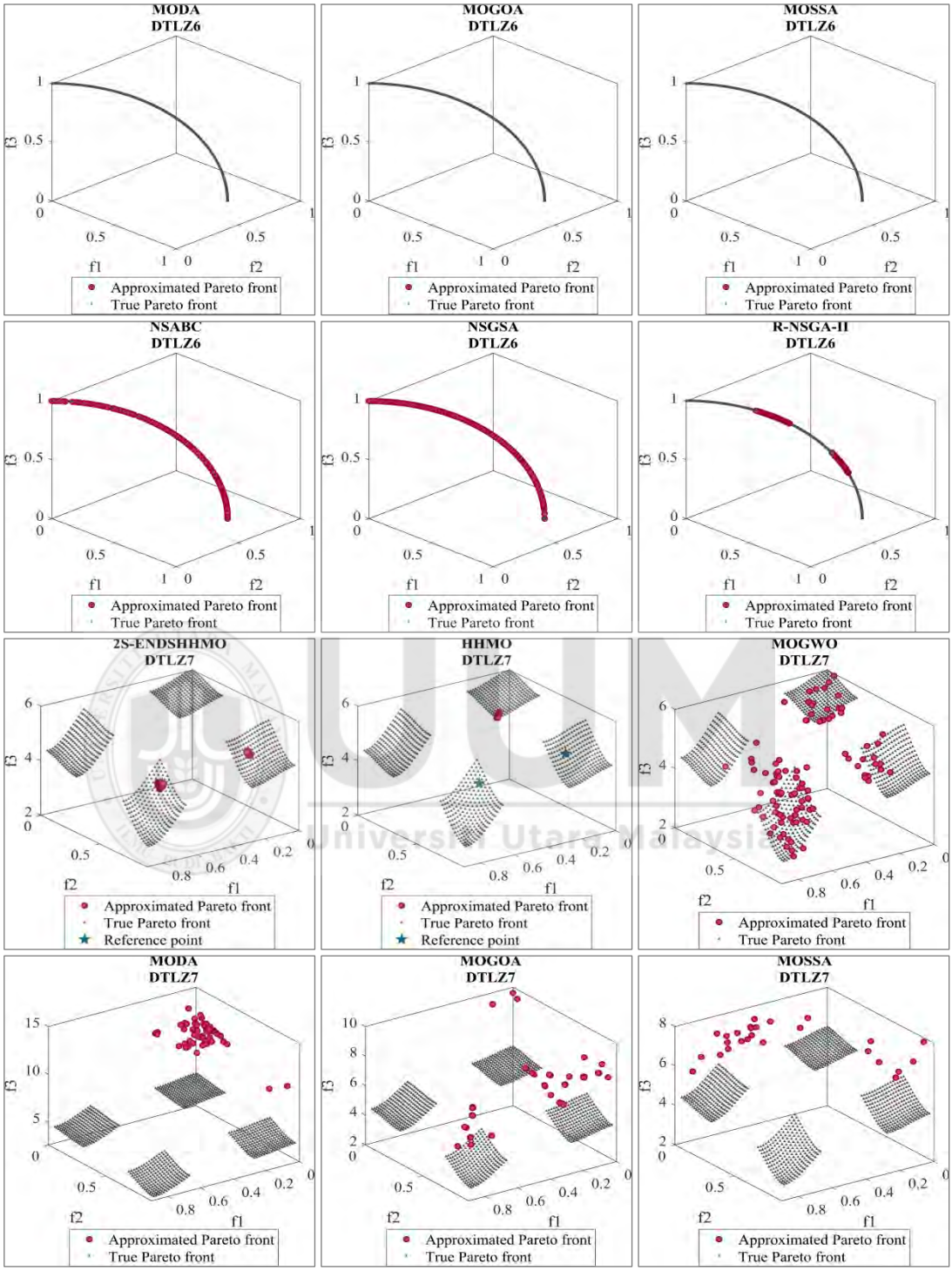


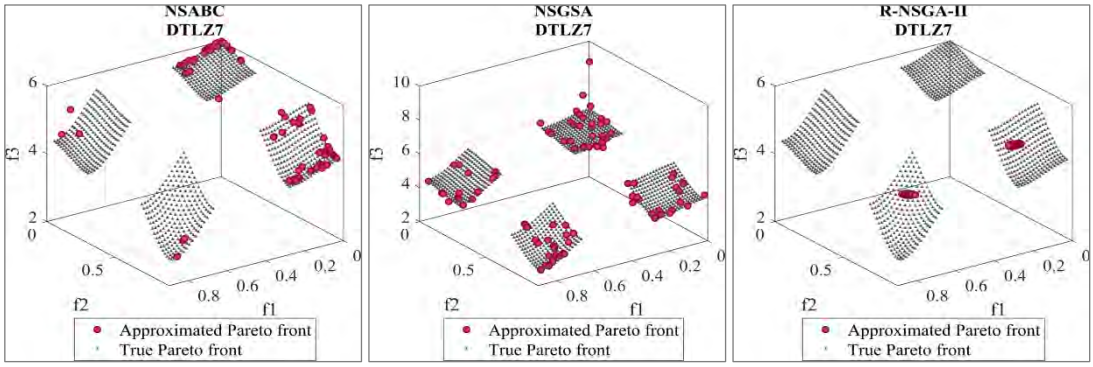




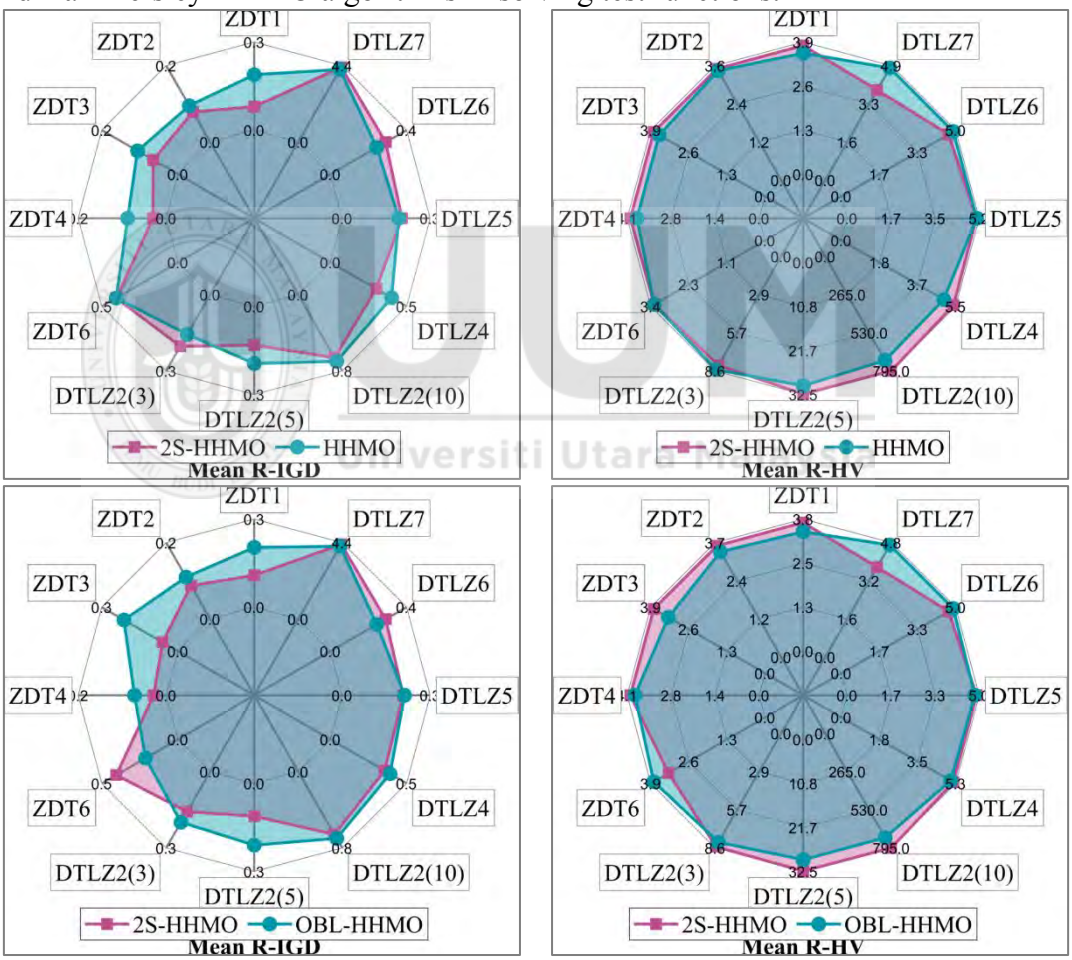


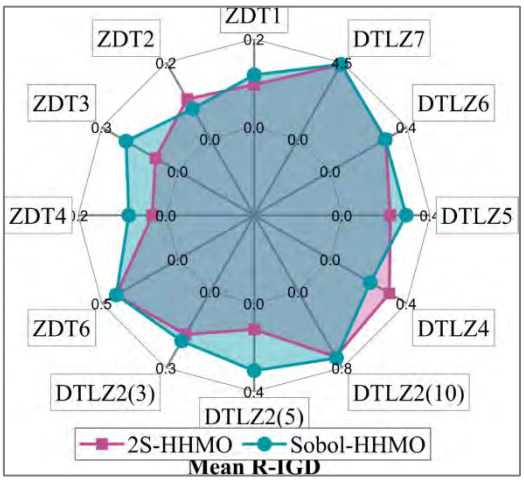


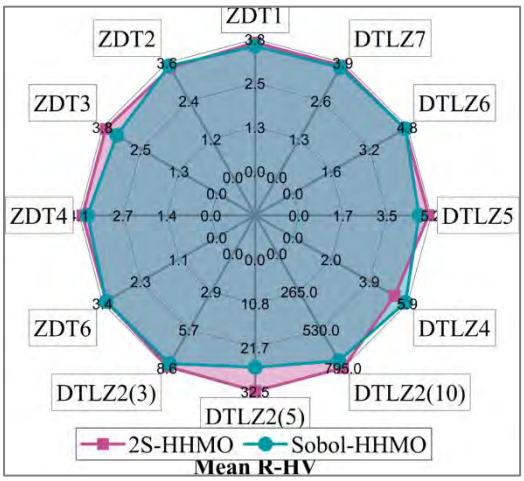


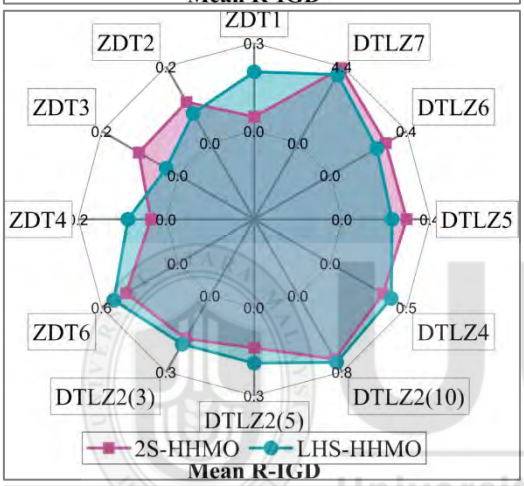


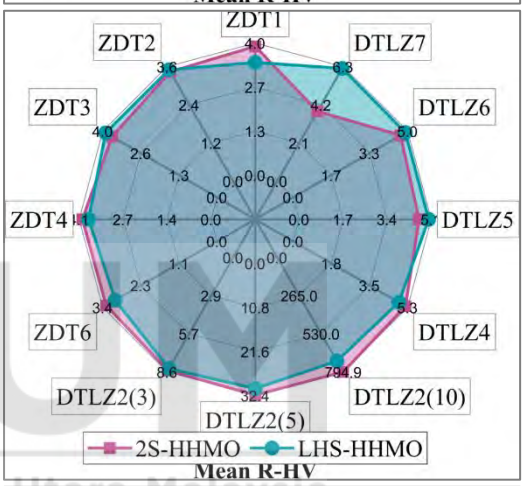
B.2: The results of HHMO, 2S-HHMO, OBL-HHMO, Sobol-HHMO, LHS-HHMO and Hammersley-HHMO algorithms in solving test functions.

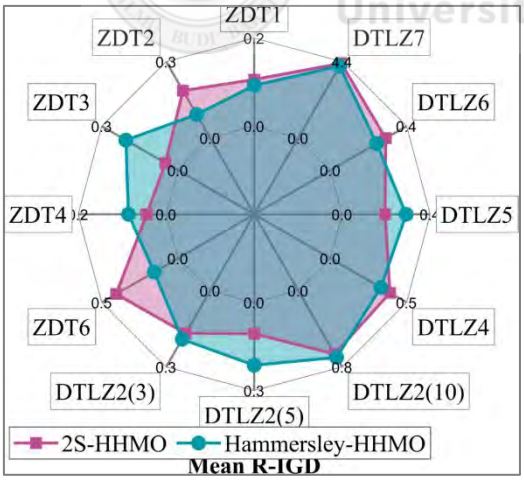


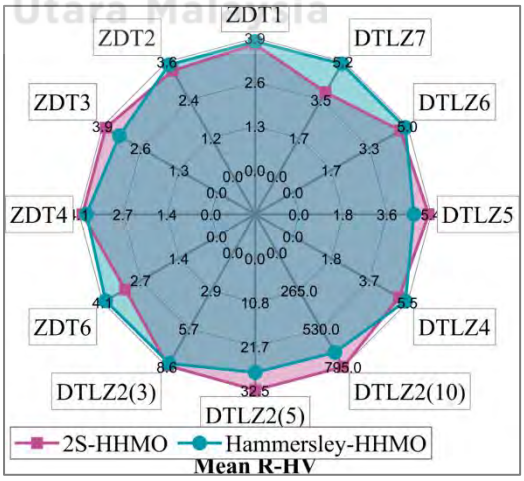












Appendix C
The Results of Solving Test Function Obtained by Using MOSI-Based Algorithms

C.1: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the ZDT1 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT1	2S-ENDSHHMO	R-IGD	5.6533E-03	6.2151E-05	5.5497E-03	5.7316E-03
		R-HV	4.0560E+00	7.7556E-03	4.0707E+00	4.0519E+00
		Epsilon	**1.2489E-02**	1.7361E-03	9.1853E-03	1.3537E-02
	HHMO	R-IGD	1.7578E-01	2.0731E-01	3.9839E-03	4.1901E-01
		R-HV	3.5367E+00	5.8494E-01	4.0465E+00	2.8585E+00
		Epsilon	1.6344E-01	1.6673E-01	1.9181E-02	3.5811E-01
	MOGWO	R-IGD	2.1932E-02	6.9049E-03	1.1696E-02	3.6331E-02
		R-HV	4.0017E+00	4.4822E-02	4.0424E+00	3.9110E+00
		Epsilon	3.9243E-02	9.8950E-03	2.2672E-02	5.7439E-02
	MOGOA	R-IGD	1.8471E+00	6.5981E-01	9.9156E-01	2.9222E+00
		R-HV	6.1001E-01	6.0848E-01	1.5677E+00	0.0000E+00
		Epsilon	1.3938E+00	4.7224E-01	7.8519E-01	2.1644E+00
	MODA	R-IGD	3.9270E+00	7.6875E-01	2.2648E+00	4.6178E+00
		R-HV	1.3736E-02	4.1208E-02	1.2362E-01	0.0000E+00
		Epsilon	2.8725E+00	5.4841E-01	1.6817E+00	3.3654E+00
	MOSSA	R-IGD	1.9638E+00	2.6132E-01	1.6923E+00	2.5700E+00
		R-HV	3.4141E-01	1.6552E-01	5.6947E-01	1.6349E-02
		Epsilon	1.4866E+00	1.8669E-01	1.2834E+00	1.9171E+00
	NSGSA	R-IGD	1.3675E-02	9.3208E-03	6.7432E-03	3.1761E-02
		R-HV	4.0226E+00	4.7315E-02	4.0783E+00	3.9228E+00
		Epsilon	3.0952E-02	1.7982E-02	1.5093E-02	7.2797E-02
	NSABC	R-IGD	1.3679E-01	4.4568E-02	7.5396E-02	1.8018E-01
		R-HV	3.5691E+00	1.0270E-01	3.7721E+00	3.4600E+00
		Epsilon	1.2766E-02	3.8367E-03	7.7160E-03	2.0508E-02
	R-NSGA-II	R-IGD	**5.2558E-03**	5.8714E-04	4.8554E-03	6.5586E-03
		R-HV	**4.1561E+00**	1.3436E-02	4.1679E+00	4.1212E+00
		Epsilon	1.3114E-02	1.7344E-03	9.7937E-03	1.5524E-02

C.2: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the ZDT2 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
ZDT2	2S-ENDSHHMO	R-IGD	5.9105E-03	1.5228E-04	5.6203E-03	6.0517E-03
		R-HV	3.8653E+00	9.6061E-03	3.8769E+00	3.8577E+00
		Epsilon	**1.2445E-02**	1.5170E-03	1.0505E-02	1.3679E-02
	HHMO	R-IGD	1.1792E-01	1.4844E-01	1.4071E-02	4.9398E-01
		R-HV	3.4469E+00	4.2979E-01	3.8114E+00	2.5232E+00
		Epsilon	1.4848E-01	1.2708E-01	4.9511E-02	4.4751E-01
	MOGWO	R-IGD	3.1017E-02	2.0845E-02	7.9890E-03	6.8484E-02
		R-HV	3.7312E+00	1.1214E-01	3.8857E+00	3.5361E+00
		Epsilon	6.0671E-02	3.7037E-02	2.0442E-02	1.1122E-01
	MOGOA	R-IGD	3.6416E+00	3.4330E+00	5.0505E-01	1.0077E+01
		R-HV	7.3508E-01	8.2651E-01	2.3759E+00	0.0000E+00
		Epsilon	2.6678E+00	2.4315E+00	4.5227E-01	7.2229E+00
	MODA	R-IGD	6.1061E+01	3.3025E+01	5.4427E+00	1.0267E+02

		R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Epsilon	4.3275E+01	2.3352E+01	3.9465E+00	7.2694E+01
		R-IGD	2.9267E+00	1.7482E+00	1.8378E-01	6.2258E+00
	MOSSA	R-HV	4.8290E-01	1.0066E+00	3.1470E+00	0.0000E+00
		Epsilon	2.1660E+00	1.2375E+00	2.1970E-01	4.5002E+00
		R-IGD	8.6036E-02	5.9997E-02	7.5630E-03	1.4984E-01
	NSGSA	R-HV	3.5109E+00	2.5676E-01	3.8635E+00	3.2400E+00
		Epsilon	1.2188E-01	7.2425E-02	1.3998E-02	1.9368E-01
		R-IGD	4.9043E-01	1.5503E-01	2.3447E-01	7.8454E-01
	NSABC	R-HV	2.4358E+00	3.4429E-01	3.0570E+00	1.8027E+00
		Epsilon	1.7627E-02	5.6174E-03	1.2766E-02	3.0807E-02
		R-IGD	**3.9049E-03**	1.1619E-04	3.7084E-03	4.0528E-03
	R-NSGA-II	R-HV	**4.6164E+00**	4.5989E-03	4.6257E+00	4.6129E+00
		Epsilon	1.3568E-02	1.1140E-04	1.3311E-02	1.3650E-02

C.3: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the ZDT3 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		R-IGD	**2.4479E-03**	6.7881E-05	2.4006E-03	2.6238E-03
	2S-ENDSHHMO	R-HV	3.9684E+00	5.9268E-03	3.9711E+00	3.9516E+00
		Epsilon	**2.6548E-03**	2.8317E-03	1.5941E-03	1.0660E-02
		R-IGD	1.9888E-01	2.0830E-01	1.5067E-02	5.8388E-01
	HHMO	R-HV	3.3788E+00	6.0141E-01	3.9525E+00	2.3330E+00
		Epsilon	1.9080E-01	1.8764E-01	1.0340E-02	5.1088E-01
		R-IGD	1.6595E-02	9.0443E-03	5.5936E-03	3.3225E-02
	MOGWO	R-HV	3.8859E+00	7.2366E-02	3.9639E+00	3.7743E+00
		Epsilon	4.1603E-02	3.5429E-02	3.8871E-03	9.6867E-02
		R-IGD	3.2415E+00	2.1386E+00	7.6807E-01	7.6261E+00
	MOGOA	R-HV	3.8803E-01	6.8126E-01	1.9528E+00	0.0000E+00
		Epsilon	2.3910E+00	1.5125E+00	6.4145E-01	5.4917E+00
		R-IGD	6.8899E+01	1.3639E+01	4.1858E+01	8.3183E+01
ZDT3	MODA	R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Epsilon	4.8818E+01	9.6442E+00	2.9697E+01	5.8919E+01
		R-IGD	2.9327E+00	2.1638E+00	8.6916E-01	5.9600E+00
	MOSSA	R-HV	6.4365E-01	7.3252E-01	1.7567E+00	0.0000E+00
		Epsilon	2.1726E+00	1.5303E+00	7.1304E-01	4.3136E+00
		R-IGD	1.8155E-01	2.8549E-01	7.2512E-03	8.2928E-01
	NSGSA	R-HV	3.4398E+00	7.3321E-01	3.9429E+00	1.8536E+00
		Epsilon	1.7585E-01	2.2105E-01	1.3602E-02	6.3721E-01
		R-IGD	5.4984E-01	8.1710E-02	3.7493E-01	6.2900E-01
	NSABC	R-HV	2.4123E+00	1.8528E-01	2.8087E+00	2.2363E+00
		Epsilon	3.0175E-02	2.6325E-02	1.1979E-02	1.0251E-01
		R-IGD	3.5202E-02	1.0165E-01	2.8174E-03	3.2450E-01
	R-NSGA-II	R-HV	**3.9875E+00**	3.1224E-01	4.0905E+00	3.0989E+00
		Epsilon	9.2610E-03	4.2335E-04	8.6339E-03	9.8482E-03

C.4: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the ZDT4 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		R-IGD	**5.0082E-03**	7.7101E-06	4.9952E-03	5.0185E-03
ZDT4	2S-ENDSHHMO	R-HV	**4.1564E+00**	2.3014E-04	4.1569E+00	4.1562E+00
		Epsilon	**7.7062E-03**	5.1414E-05	7.6552E-03	7.8243E-03

		R-IGD	4.3117E-02	5.7142E-02	2.7345E-03	1.5176E-01
	HHMO	R-HV	3.9498E+00	2.4825E-01	4.1342E+00	3.4984E+00
		Epsilon	7.4713E-02	7.3575E-02	1.3438E-02	1.9567E-01
		R-IGD	2.7842E-02	1.2659E-02	1.0843E-02	5.3222E-02
	MOGWO	R-HV	4.0476E+00	8.7156E-02	4.1602E+00	3.8900E+00
		Epsilon	5.2254E-02	2.3044E-02	2.2817E-02	8.9581E-02
		R-IGD	6.7458E+00	1.9786E+00	4.2296E+00	8.9919E+00
	MOGOA	R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Epsilon	4.8702E+00	1.3991E+00	3.0909E+00	6.4585E+00
		R-IGD	5.7661E+00	2.3054E+00	3.2902E+00	1.0999E+01
	MODA	R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Epsilon	4.1774E+00	1.6303E+00	2.4264E+00	7.8776E+00
		R-IGD	7.4892E-01	1.0744E+00	2.9453E-02	2.9653E+00
	MOSSA	R-HV	2.7139E+00	1.6781E+00	4.0189E+00	0.0000E+00
		Epsilon	5.9747E-01	7.8088E-01	4.8498E-02	2.1966E+00
		R-IGD	1.0291E-02	8.1070E-03	4.9933E-03	3.2113E-02
	NSGSA	R-HV	4.1361E+00	2.6953E-02	4.1704E+00	4.0782E+00
		Epsilon	2.0131E-02	1.0115E-02	9.6700E-03	4.3893E-02
		R-IGD	5.1382E-01	4.3659E-01	3.3444E-02	1.2071E+00
	NSABC	R-HV	2.7297E+00	1.0458E+00	4.0554E+00	1.2487E+00
		Epsilon	3.6123E-02	3.1670E-02	9.4745E-03	1.1284E-01
		R-IGD	3.8590E-02	1.0436E-01	5.5525E-03	3.3561E-01
	R-NSGA-II	R-HV	3.9603E+00	3.1055E-01	4.0636E+00	3.0766E+00
		Epsilon	7.1269E-02	1.3427E-01	7.4856E-03	3.2604E-01

C.5: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the ZDT6 problem

MOP	**Algorithm**	**Metric**	**Mean**	**SD**	**Best**	**Worst**
		R-IGD	**2.5867E-03**	7.7699E-05	2.4822E-03	2.6690E-03
	2S-ENDSHHMO	R-HV	**4.4545E+00**	6.2963E-03	4.4620E+00	4.4494E+00
		Epsilon	9.7824E-03	1.3105E-04	9.5260E-03	9.9504E-03
		R-IGD	4.2760E-01	4.8118E-01	4.5574E-03	1.0499E+00
	HHMO	R-HV	3.2860E+00	1.2621E+00	4.4118E+00	1.6868E+00
		Epsilon	3.6691E-01	3.7357E-01	3.3946E-02	8.4102E-01
		R-IGD	1.7475E-01	1.1547E-02	1.6181E-01	1.9315E-01
	MOGWO	R-HV	3.7302E+00	5.0942E-02	3.7922E+00	3.6555E+00
		Epsilon	2.0099E-01	6.2944E-03	1.9610E-01	2.1585E-01
		R-IGD	1.3721E+00	9.2924E-01	1.6912E-01	2.6192E+00
	MOGOA	R-HV	1.5504E+00	1.5514E+00	3.7503E+00	3.5517E-02
		Epsilon	1.0740E+00	6.7052E-01	1.9868E-01	1.9516E+00
		R-IGD	7.5319E+00	5.6976E+00	9.7107E-01	1.3229E+01
ZDT6	MODA	R-HV	7.2620E-01	9.3780E-01	1.8391E+00	0.0000E+00
		Epsilon	5.4403E+00	4.0103E+00	8.3098E-01	9.4541E+00
		R-IGD	1.3112E+00	1.7543E+00	1.5985E-01	5.9299E+00
	MOSSA	R-HV	2.1829E+00	1.4531E+00	3.8069E+00	0.0000E+00
		Epsilon	1.0207E+00	1.2447E+00	1.9488E-01	4.2930E+00
		R-IGD	3.3436E-01	3.3599E-01	1.6009E-01	9.7127E-01
	NSGSA	R-HV	3.3564E+00	8.0277E-01	3.8045E+00	1.8384E+00
		Epsilon	3.3224E-01	2.6346E-01	1.9429E-01	8.3101E-01
		R-IGD	1.6359E-01	2.3880E-03	1.6090E-01	1.6812E-01
	NSABC	R-HV	3.7813E+00	1.4058E-02	3.7979E+00	3.7553E+00
		Epsilon	1.9684E-01	9.3502E-04	1.9565E-01	1.9845E-01
		R-IGD	2.8213E-03	1.7476E-05	2.8024E-03	2.8514E-03
	R-NSGA-II	R-HV	4.2682E+00	1.0119E-04	4.2683E+00	4.2680E+00

	Epsilon	**8.8925E-03**	1.8335E-03	6.2315E-03	1.0216E-02

C.6: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ2 with three objective problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
DTLZ2 (3)	2S-ENDSHHMO	R-IGD	**5.9549E-02**	2.3553E-03	5.6725E-02	6.2936E-02
		R-HV	**9.5813E+00**	3.4719E-02	9.6307E+00	9.5159E+00
		Epsilon	**9.3292E-02**	4.7400E-03	8.7793E-02	1.0443E-01
	HHMO	R-IGD	6.0026E-01	2.1995E-01	2.1479E-01	8.0993E-01
		R-HV	5.6807E+00	1.2562E+00	8.0106E+00	4.5061E+00
		Epsilon	4.4202E-01	1.3158E-01	2.1195E-01	5.7064E-01
	MOGWO	R-IGD	1.0205E-01	4.0249E-02	3.7206E-02	1.8057E-01
		R-HV	9.5112E+00	4.0018E-01	1.0049E+01	8.9110E+00
		Epsilon	1.4109E-01	2.6621E-02	8.7672E-02	1.8525E-01
	MOGOA	R-IGD	2.9055E-01	7.0896E-02	1.5081E-01	3.8570E-01
		R-HV	7.5175E+00	5.2667E-01	8.4426E+00	6.7484E+00
		Epsilon	2.5980E-01	4.6728E-02	1.7181E-01	3.1871E-01
	MODA	R-IGD	2.5222E-01	5.6295E-02	1.4158E-01	3.1011E-01
		R-HV	7.7335E+00	4.4477E-01	8.7542E+00	7.2628E+00
		Epsilon	2.1343E-01	3.2858E-02	1.7374E-01	2.6945E-01
	MOSSA	R-IGD	1.5843E-01	7.3946E-02	6.2990E-02	3.0614E-01
		R-HV	8.9404E+00	7.3812E-01	9.9391E+00	7.6117E+00
		Epsilon	1.7325E-01	5.0972E-02	1.0279E-01	2.6363E-01
	NSGSA	R-IGD	1.5671E-01	2.9675E-02	1.1367E-01	2.1107E-01
		R-HV	8.6085E+00	3.0912E-01	8.9140E+00	7.9518E+00
		Epsilon	1.7697E-01	2.1174E-02	1.4449E-01	2.1208E-01
	NSABC	R-IGD	9.3907E-02	2.5399E-02	6.5924E-02	1.4782E-01
		R-HV	9.3911E+00	3.7595E-01	9.8107E+00	8.7152E+00
		Epsilon	1.2847E-01	2.0941E-02	1.0489E-01	1.7087E-01
	R-NSGA-II	R-IGD	6.8791E-02	1.9271E-04	6.8488E-02	6.9211E-02
		R-HV	6.7764E+00	3.3680E-03	6.7841E+00	6.7709E+00
		Epsilon	9.8999E-02	4.7192E-03	9.3145E-02	1.0855E-01

C.7: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ2 with five objective problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
DTLZ2 (5)	2S-ENDSHHMO	R-IGD	**7.6619E-02**	1.4910E-02	3.9975E-02	9.3158E-02
		R-HV	**3.4787E+01**	6.3405E-01	3.5930E+01	3.3869E+01
		Epsilon	4.5117E-01	2.0801E-01	2.7440E-01	6.9515E-01
	HHMO	R-IGD	7.5021E-01	4.9654E-01	1.5725E-01	1.1974E+00
		R-HV	1.7566E+01	1.1041E+01	3.1327E+01	8.1699E+00
		Epsilon	5.9848E-01	2.2674E-01	3.2685E-01	8.0195E-01
	MOGWO	R-IGD	5.0390E-01	3.3021E-01	1.9233E-01	1.1900E+00
		R-HV	2.1884E+01	7.4073E+00	2.9984E+01	8.1736E+00
		Epsilon	4.8267E-01	1.4808E-01	3.4274E-01	7.9020E-01
	MOGOA	R-IGD	9.2398E-01	2.4648E-01	6.3277E-01	1.4459E+00
		R-HV	1.2796E+01	3.8648E+00	1.8854E+01	6.1193E+00
		Epsilon	6.7128E-01	1.1300E-01	5.3950E-01	9.1598E-01
	MODA	R-IGD	5.6361E-01	2.2428E-01	1.6053E-01	8.8449E-01
		R-HV	1.9573E+01	5.6045E+00	3.0508E+01	1.2500E+01
		Epsilon	4.9022E-01	8.0117E-02	3.6950E-01	6.5527E-01
	MOSSA	R-IGD	6.2771E-01	3.0728E-01	2.1831E-01	1.0895E+00

		R-HV	1.9881E+01	8.1059E+00	3.2297E+01	9.4042E+00
		Epsilon	5.4123E-01	1.3804E-01	3.5415E-01	7.4377E-01
		R-IGD	7.6097E-01	1.3977E-01	5.8356E-01	1.0552E+00
	NSGSA	R-HV	1.4904E+01	2.5729E+00	1.8470E+01	9.8739E+00
		Epsilon	5.9684E-01	6.2505E-02	5.1750E-01	7.2843E-01
		R-IGD	2.1415E-01	6.4732E-02	8.9990E-02	3.1737E-01
	NSABC	R-HV	2.9652E+01	2.7828E+00	3.5493E+01	2.6142E+01
		Epsilon	3.3848E-01	4.4323E-02	2.5862E-01	3.9845E-01
		R-IGD	3.7859E-01	2.3095E-04	3.7830E-01	3.7893E-01
	R-NSGA-II	R-HV	2.5925E+01	1.4108E-02	2.5950E+01	2.5898E+01
		Epsilon	**2.8268E-01**	2.1529E-03	2.7973E-01	2.8530E-01

C.8: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ2 with 10 objective problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		R-IGD	5.6874E-01	2.5620E-03	5.6503E-01	5.7213E-01
	2S-ENDSHHMO	R-HV	1.0563E+03	6.0745E+00	1.0644E+03	1.0495E+03
		Epsilon	4.4814E-01	3.6988E-03	4.4218E-01	4.5381E-01
		R-IGD	6.9929E-01	1.0747E-01	6.0993E-01	9.6582E-01
	HHMO	R-HV	7.2147E+02	1.4911E+02	8.8498E+02	3.8534E+02
		Epsilon	5.1777E-01	4.6457E-02	4.7342E-01	6.2113E-01
		R-IGD	1.2643E+00	3.8482E-01	8.8009E-01	2.1023E+00
	MOGWO	R-HV	2.5682E+02	1.5359E+02	4.6246E+02	3.3480E+01
		Epsilon	7.4469E-01	1.3402E-01	6.0589E-01	1.0299E+00
		R-IGD	2.7689E+00	9.0655E-01	1.6289E+00	4.5560E+00
	MOGOA	R-HV	2.5465E+01	3.4631E+01	9.3847E+01	9.9364E-03
		Epsilon	1.2450E+00	2.9265E-01	8.7179E-01	1.8181E+00
		R-IGD	1.3347E+00	1.6284E-01	1.1595E+00	1.6843E+00
DTLZ2 (10)	MODA	R-HV	1.8650E+02	5.8551E+01	2.6693E+02	8.3273E+01
		Epsilon	7.5077E-01	8.7501E-02	5.3867E-01	8.3335E-01
		R-IGD	1.8581E+00	7.1830E-01	9.9909E-01	2.8972E+00
	MOSSA	R-HV	1.3480E+02	1.4522E+02	3.7465E+02	4.6708E+00
		Epsilon	9.1429E-01	2.4828E-01	5.5960E-01	1.2351E+00
		R-IGD	5.0583E+00	6.1075E-01	3.5306E+00	5.7540E+00
	NSGSA	R-HV	6.5051E-02	2.0399E-01	6.4563E-01	9.1424E-07
		Epsilon	1.9794E+00	1.9428E-01	1.4928E+00	2.2010E+00
		R-IGD	8.6878E-01	9.8349E-02	6.1804E-01	9.6330E-01
	NSABC	R-HV	4.7561E+02	1.2133E+02	7.9915E+02	3.7452E+02
		Epsilon	4.9936E-01	7.3545E-02	4.0907E-01	6.3009E-01
		R-IGD	**5.6030E-01**	6.9578E-03	5.4934E-01	5.7162E-01
	R-NSGA-II	R-HV	**1.1331E+03**	6.9625E+00	1.1438E+03	1.1222E+03
		Epsilon	**4.3551E-01**	5.8464E-03	4.2780E-01	4.4685E-01

C.9: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ4 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		R-IGD	**3.8053E-02**	8.8330E-03	2.7184E-02	5.5236E-02
	2S-ENDSHHMO	R-HV	7.3875E+00	1.1352E-01	7.5483E+00	7.2267E+00
		Epsilon	**2.6143E-01**	1.7387E-02	2.3765E-01	2.9575E-01
DTLZ4		R-IGD	5.7950E-01	3.4731E-01	1.3918E-01	1.1994E+00
	HHMO	R-HV	4.1206E+00	1.4373E+00	6.0592E+00	1.7029E+00
		Epsilon	6.3221E-01	2.0358E-01	3.6165E-01	9.9056E-01

		R-IGD	1.2659E-01	1.2395E-01	5.8057E-02	4.7243E-01
	MOGWO	R-HV	6.5341E+00	8.0114E-01	7.0502E+00	4.4304E+00
		Epsilon	2.8752E-01	1.0754E-01	2.0477E-01	5.6700E-01
		R-IGD	3.2835E-01	9.8024E-02	1.1637E-01	4.6030E-01
	MOGOA	R-HV	5.1429E+00	6.2712E-01	6.6862E+00	4.5712E+00
		Epsilon	4.8586E-01	6.1939E-02	3.5177E-01	5.7331E-01
		R-IGD	6.2549E-01	1.6480E-01	3.8460E-01	9.2714E-01
	MODA	R-HV	3.6139E+00	6.7265E-01	4.6480E+00	2.4711E+00
		Epsilon	6.5886E-01	9.4707E-02	5.1585E-01	8.3277E-01
		R-IGD	5.3334E-01	1.6000E-01	2.5805E-01	8.2889E-01
	MOSSA	R-HV	4.0771E+00	6.8682E-01	5.3146E+00	2.9401E+00
		Epsilon	6.0461E-01	9.4156E-02	4.3961E-01	7.7694E-01
		R-IGD	9.9507E-02	3.1256E-02	6.2453E-02	1.7586E-01
	NSGSA	R-HV	6.6878E+00	3.6003E-01	6.9985E+00	5.8068E+00
		Epsilon	3.1995E-01	3.6730E-02	2.6412E-01	3.8729E-01
		R-IGD	3.1561E-01	4.8231E-02	2.1301E-01	3.8375E-01
	NSABC	R-HV	5.0484E+00	2.5911E-01	5.5758E+00	4.6522E+00
		Epsilon	4.7701E-01	3.0485E-02	4.1147E-01	5.1535E-01
		R-IGD	2.2783E-01	6.9314E-03	2.1622E-01	2.3768E-01
	R-NSGA-II	R-HV	**9.6024E+00**	8.4958E-02	9.7438E+00	9.4849E+00
		Epsilon	3.0701E-01	1.3558E-03	3.0457E-01	3.0825E-01

C.10: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ5 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
		R-IGD	**1.6876E-01**	6.6029E-04	1.6793E-01	1.6985E-01
	2S-ENDSHHMO	R-HV	5.4548E+00	1.7573E-02	5.4946E+00	5.4279E+00
		Epsilon	3.9142E-01	1.3145E-02	3.6415E-01	4.0844E-01
		R-IGD	3.3653E-01	3.1601E-02	3.0120E-01	4.0039E-01
	HHMO	R-HV	4.1184E+00	1.6014E-01	4.2984E+00	3.8123E+00
		Epsilon	5.6209E-01	3.5722E-02	4.9721E-01	6.2420E-01
		R-IGD	2.3035E-01	2.7160E-02	1.8966E-01	2.7284E-01
	MOGWO	R-HV	4.9623E+00	2.2504E-01	5.3912E+00	4.6664E+00
		Epsilon	2.5849E-01	3.2786E-02	2.1593E-01	3.1466E-01
		R-IGD	3.1746E-01	4.5262E-02	2.8066E-01	4.4190E-01
	MOGOA	R-HV	4.3132E+00	2.6629E-01	4.5156E+00	3.5993E+00
		Epsilon	3.5447E-01	9.4083E-02	2.7092E-01	5.2824E-01
		R-IGD	3.5391E-01	2.2466E-02	3.1868E-01	3.9900E-01
DTLZ5	MODA	R-HV	4.0891E+00	1.5334E-01	4.2919E+00	3.7923E+00
		Epsilon	5.1701E-01	1.1219E-01	3.4092E-01	6.2134E-01
		R-IGD	2.9086E-01	3.1527E-02	2.5321E-01	3.4472E-01
	MOSSA	R-HV	4.5039E+00	2.3231E-01	4.7619E+00	4.0113E+00
		Epsilon	3.4816E-01	9.9156E-02	2.5201E-01	5.9869E-01
		R-IGD	1.7278E-01	6.5315E-03	1.6842E-01	1.8988E-01
	NSGSA	R-HV	5.5379E+00	9.6315E-02	5.6158E+00	5.2875E+00
		Epsilon	2.1184E-01	6.9203E-03	2.0322E-01	2.2437E-01
		R-IGD	1.6904E-01	1.1792E-03	1.6789E-01	1.7188E-01
	NSABC	R-HV	5.5980E+00	1.8782E-02	5.6265E+00	5.5663E+00
		Epsilon	**2.0560E-01**	2.1782E-03	2.0254E-01	2.0930E-01
		R-IGD	1.8483E-01	6.8477E-03	1.7431E-01	1.9332E-01
	R-NSGA-II	R-HV	**8.3792E+00**	6.6564E-02	8.4857E+00	8.2837E+00
		Epsilon	3.7102E-01	2.0183E-03	3.6753E-01	3.7232E-01

C.11: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ6 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
DTLZ6	2S-ENDSHHMO	R-IGD	**1.6895E-01**	9.8350E-04	1.6753E-01	1.7077E-01
		R-HV	5.4428E+00	1.8243E-02	5.4773E+00	5.4096E+00
		Epsilon	4.0093E-01	5.2802E-03	3.9267E-01	4.1120E-01
	HHMO	R-IGD	2.2187E-01	1.9510E-03	2.1707E-01	2.2336E-01
		R-HV	4.8669E+00	1.7625E-02	4.9089E+00	4.8524E+00
		Epsilon	4.1018E-01	3.3325E-03	4.0167E-01	4.1275E-01
	MOGWO	R-IGD	1.7768E-01	8.0905E-03	1.6968E-01	1.9037E-01
		R-HV	5.4374E+00	1.0926E-01	5.5922E+00	5.2749E+00
		Epsilon	2.3067E-01	2.4750E-02	2.0533E-01	2.8999E-01
	MOGOA	R-IGD	9.7233E-01	8.1519E-01	1.7361E-01	1.8806E+00
		R-HV	2.7454E+00	2.4244E+00	5.4467E+00	3.1413E-01
		Epsilon	8.4654E-01	5.6825E-01	2.3034E-01	1.4931E+00
	MODA	R-IGD	3.9933E+00	7.7837E-01	2.6549E+00	5.1273E+00
		R-HV	1.7957E-03	4.0342E-03	1.1936E-02	0.0000E+00
		Epsilon	2.7329E+00	4.4999E-01	1.9591E+00	3.3883E+00
	MOSSA	R-IGD	1.3178E+00	7.1206E-01	2.3362E-01	1.8984E+00
		R-HV	1.5933E+00	1.8819E+00	4.6838E+00	2.9705E-01
		Epsilon	1.1711E+00	4.2547E-01	5.1448E-01	1.5195E+00
	NSGSA	R-IGD	1.7035E-01	1.4058E-03	1.6755E-01	1.7224E-01
		R-HV	5.5818E+00	2.1884E-02	5.6181E+00	5.5527E+00
		Epsilon	2.0932E-01	4.2590E-03	2.0329E-01	2.1573E-01
	NSABC	R-IGD	1.6977E-01	1.0703E-03	1.6831E-01	1.7119E-01
		R-HV	5.5846E+00	2.2729E-02	5.6203E+00	5.5486E+00
		Epsilon	**2.0914E-01**	3.7789E-03	2.0356E-01	2.1329E-01
	R-NSGA-II	R-IGD	1.7178E-01	1.8877E-05	1.7173E-01	1.7180E-01
		R-HV	**8.5256E+00**	3.1757E-04	8.5260E+00	8.5251E+00
		Epsilon	3.7082E-01	2.2418E-03	3.6754E-01	3.7232E-01

C.12: Mean, SD, best, worst R-IGD, R-HV and epsilon values in solving the DTLZ7 problem

MOP	Algorithm	Metric	Mean	SD	Best	Worst
DTLZ7	2S-ENDSHHMO	R-IGD	3.6730E+00	3.2276E-03	3.6669E+00	3.6754E+00
		R-HV	**8.3501E+00**	1.5098E-02	8.3776E+00	8.3337E+00
		Epsilon	3.7011E+00	4.1768E-03	3.6942E+00	3.7084E+00
	HHMO	R-IGD	4.2125E+00	4.1372E-01	3.7309E+00	4.9373E+00
		R-HV	4.7775E+00	2.4937E+00	8.1092E+00	1.3925E+00
		Epsilon	4.1236E+00	3.1164E-01	3.7595E+00	4.6684E+00
	MOGWO	R-IGD	3.8823E+00	2.2863E-01	3.6607E+00	4.2969E+00
		R-HV	6.7901E+00	1.7096E+00	8.3633E+00	3.8053E+00
		Epsilon	3.6900E+00	1.9904E-01	3.4992E+00	3.9675E+00
	MOGOA	R-IGD	7.2160E+00	1.7599E+00	4.9161E+00	1.0791E+01
		R-HV	1.8381E-01	4.6458E-01	1.4948E+00	0.0000E+00
		Epsilon	6.1277E+00	1.0933E+00	4.6524E+00	8.3168E+00
	MODA	R-IGD	1.0543E+01	1.3830E+00	7.4492E+00	1.1962E+01
		R-HV	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
		Epsilon	8.1665E+00	8.3053E-01	6.2916E+00	9.0125E+00
	MOSSA	R-IGD	5.1444E+00	6.6167E-01	4.5534E+00	6.2381E+00
		R-HV	1.3980E+00	1.1383E+00	2.6047E+00	1.4433E-02
		Epsilon	4.7826E+00	4.6587E-01	4.2298E+00	5.5290E+00
	NSGSA	R-IGD	3.7647E+00	4.5984E-02	3.6884E+00	3.8205E+00

	R-HV	7.6655E+00	3.6355E-01	8.4147E+00	7.1367E+00
	Epsilon	3.5684E+00	6.3018E-02	3.5249E+00	3.7304E+00
	R-IGD	3.7577E+00	3.4435E-02	3.7044E+00	3.8053E+00
NSABC	R-HV	7.8401E+00	2.5986E-01	8.2173E+00	7.4277E+00
	Epsilon	**3.5461E+00**	4.4666E-02	3.4936E+00	3.6363E+00
	R-IGD	**3.1946E+00**	5.7905E-03	3.1887E+00	3.2023E+00
R-NSGA-II	R-HV	7.0641E+00	2.1187E-02	7.0916E+00	7.0396E+00
	Epsilon	3.7437E+00	5.1019E-04	3.7427E+00	3.7443E+00

Appendix D
Solutions corresponding to the extreme points obtained by each algorithm in solving engineering MOPs

Welded beam MOP

D.1: Decision variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving welded beam MOP

Solution No.	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.2061	3.3382	9.4437	0.2075	**1.7916**	0.0126
2	1.6511	0.3012	10.0000	5.0000	35.3087	**0.0004**

D.2: Decision variables corresponding to extreme points, obtained by using HHMO algorithm in solving welded beam MOP

Solution No.	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.2530	2.9645	8.1542	0.2535	**1.8964**	0.0160
2	0.5717	1.1301	10.0000	4.7485	34.9732	**0.0005**

D.3: Decision variables corresponding to extreme points, obtained by using MOGWO algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.2572	2.7264	9.1473	0.2678	**2.1702**	0.0107
2	1.1416	1.5903	10.0000	5.0000	39.7919	**0.0004**

D.4: Decision variables corresponding to extreme points, obtained by using MOGOA algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.2575	3.4483	6.8986	0.3732	**2.4140**	0.0179
2	3.2083	0.1863	10.0000	5.0000	36.2433	**0.0004**

D.5: Decision variables corresponding to extreme points, obtained by using MODA algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.3132	3.1707	6.7287	0.6044	**3.7029**	0.0119
2	4.7797	0.1015	10.0000	4.9413	36.0834	**0.0004**

D.6: Decision variables corresponding to extreme points, obtained by using MOSSA algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.1250	6.6276	10.0000	0.2323	**2.4198**	0.0094
2	4.7387	0.1382	9.5608	5.0000	35.9447	**0.0005**

D.7: Decision variables corresponding to extreme points, obtained by using NSGSA algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.1250	6.3094	10.0000	0.2145	**2.2046**	0.0102
2	1.4979	0.3582	10.0000	5.0000	35.4265	**0.0004**

D.8: Decision variables corresponding to extreme points, obtained by using NSABC algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.4721	1.4589	8.5756	0.6341	**4.4035**	0.0055
2	4.0123	1.1024	0.5123	0.9073	36.0379	**0.0004**

D.9: Decision variables corresponding to extreme points, obtained by using R-NSGA-II algorithm in solving welded beam MOP

Solution	*h*	*l*	*t*	*b*	Fabrication cost	Deflection of the beam
1	0.1937	3.6330	10.0000	0.2053	**1.8924**	0.0107
2	1.6125	0.3090	10.0000	5.0000	35.3079	**0.0004**

Four-bar Truss

D.10: Decision variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving four-bar truss MOP

Solution	x_1(cm^2)	x_2 (cm^2)	x_3(cm^2)	x_4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.4142	1.0000	**1174.2000**	0.0741
2	1.0000	3.0000	3.0000	3.0000	1836.3081	**0.0322**

D.11: Decision variables corresponding to extreme points, obtained by using HHMO algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.4142	1.0000	**1174.2000**	0.0741
2	1.0361	3.0000	3.0000	3.0000	1850.7469	**0.0322**

D.12: Decision variables corresponding to extreme points, obtained by using MOGWO algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.4142	1.0000	**1174.2000**	0.0741
2	1.0397	3.0000	3.0000	2.9164	1835.4688	**0.0324**

D.13: Decision variables corresponding to extreme points, obtained by using MOGOA algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.5892	1.0000	**1188.4886**	0.0719
2	1.0000	3.0000	2.8880	2.9894	1827.6524	**0.0326**

D.14: Decision variables corresponding to extreme points, obtained by using MODA algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0005	1.4142	1.4540	1.0000	**1177.7307**	0.0736
2	1.0000	3.0000	3.0000	2.9828	1832.8669	**0.0322**

D.15: Decision variables corresponding to extreme points, obtained by using MOSSA algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4909	1.4143	1.0000	**1183.2001**	0.0724
2	1.0000	3.0000	2.9942	2.9156	1819.0943	**0.0324**

D.16: Decision variables corresponding to extreme points, obtained by using NSGSA algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.4142	1.0000	**1174.2000**	0.0741
2	1.0000	3.0000	3.0000	3.0000	1836.3081	**0.0322**

D.17: Decision variables corresponding to extreme points, obtained by using NSABC algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	2.0457	1.4264	1.0104	**1245.4801**	0.0632
2	1.0000	3.0000	2.6807	2.8440	1786.1606	**0.0337**

D.18: Decision variables corresponding to extreme points, obtained by using R-NSGA-II algorithm in solving four-bar truss MOP

Solution	x1(cm^2)	x2 (cm^2)	x3(m^2)	x4(cm^2)	Volume (cm^3)	Deflection (KPa)
1	1.0000	1.4142	1.4142	1.0000	**1174.2000**	0.0741
2	1.0000	3.0000	3.0000	3.0000	1836.3081	**0.0322**

Optimal Power Flow

D.19: Control variables corresponding to extreme points, obtained by using 2S-ENDSHHMO algorithm in solving OPF MOP

Control variables		Solutions	
		A	B
P_G (MW)	P_{G1}	175.56	51.42
	P_{G2}	48.50	80.00
	P_{G5}	21.27	49.96
	P_{G8}	22.84	35.00
	P_{G11}	11.83	30.00
	P_{G13}	12.00	40.00
V (p.u)	V_{G1}	1.10	1.10
	V_{G2}	1.09	1.09
	V_{G5}	1.06	1.08

	V_{G8}	1.07	1.08
	V_{G11}	1.10	1.10
	V_{G13}	1.10	1.10
	T_{11}	0.99	0.98
Tap (p.u)	T_{12}	0.96	0.97
	T_{15}	1.01	1.00
	T_{36}	0.97	0.98
	QC10	1.37	0.33
	QC12	0.44	4.61
	QC15	0.26	1.49
	QC17	4.80	2.36
Capacitor bank (MVar)	QC 20	1.76	0.48
	QC 21	4.31	3.00
	QC 23	4.75	4.42
	QC 24	0.02	0.18
	QC 29	3.91	4.45
Generating cost (Unit/h)		**799.4458**	967.1640
Transmission loss (MW)		8.5986	**2.9759**

D.20: Control variables corresponding to extreme points, obtained by using HHMO algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	B
P_G (MW)	P_{G1}	171.72	61.91
	P_{G2}	49.48	72.43
	P_{G5}	22.19	50.00
	P_{G8}	22.31	35.00
	P_{G11}	13.90	27.47
	P_{G13}	12.10	40.00
V (p.u)	V_{G1}	1.10	1.08
	V_{G2}	1.09	1.08
	V_{G5}	1.06	1.04
	V_{G8}	1.07	1.07
	V_{G11}	1.10	1.09
	V_{G13}	1.09	1.10
Tap (p.u)	T_{11}	1.03	1.03
	T_{12}	0.98	1.09
	T_{15}	1.03	1.08
	T_{36}	1.00	1.01
Capacitor bank (MVar)	QC10	4.73	4.93
	QC12	3.81	2.46
	QC15	3.18	4.11
	QC17	2.63	4.96
	QC 20	2.28	5.00
	QC 21	4.65	2.94
	QC 23	4.97	2.24
	QC 24	4.31	3.62
	QC 29	2.50	1.68
Generating cost (Unit/h)		**799.5767**	948.2577
Transmission loss (MW)		8.3032	**3.4145**

D.21: Control variables corresponding to extreme points, obtained by using MOGWO algorithm in solving OPF MOP

Control variables		Solutions	
		A	B
P_G (MW)	P_{G1}	169.51	51.44
	P_{G2}	47.77	80.00
	P_{G5}	21.78	50.00
	P_{G8}	26.36	35.00
	P_{G11}	12.56	30.00
	P_{G13}	13.75	40.00
V (p.u)	V_{G1}	1.09	1.10
	V_{G2}	1.08	1.10
	V_{G5}	1.04	1.09
	V_{G8}	1.05	1.09
	V_{G11}	1.07	1.10
	V_{G13}	1.08	1.10
Tap (p.u)	T_{11}	0.97	0.97
	T_{12}	1.00	1.06
	T_{15}	1.02	1.10
	T_{36}	0.99	1.02
Capacitor bank (MVar)	QC10	3.45	2.26
	QC12	2.77	4.90
	QC15	3.09	5.00
	QC17	3.55	5.00
	QC 20	2.74	4.94
	QC 21	1.33	3.92
	QC 23	2.48	5.00
	QC 24	4.23	5.00
	QC 29	3.20	4.69
Generating cost (Unit/h)		**800.7885**	967.5126
Transmission loss (MW)		8.3282	**3.0389**

D.22: Control variables corresponding to extreme points, obtained by using MOGOA algorithm in solving OPF MOP

Control variables		Solutions	
		A	B
P_G (MW)	P_{G1}	161.40	90.63
	P_{G2}	45.53	67.89
	P_{G5}	31.42	46.09
	P_{G8}	11.99	18.02
	P_{G11}	23.49	30.00
	P_{G13}	17.94	34.92
V (p.u)	V_{G1}	1.08	1.10
	V_{G2}	1.06	1.09
	V_{G5}	1.03	1.07
	V_{G8}	1.02	1.07
	V_{G11}	0.99	1.09
	V_{G13}	1.00	1.10
Tap (p.u)	T_{11}	1.00	1.02
	T_{12}	1.05	0.95
	T_{15}	1.00	0.99
	T_{36}	0.93	0.97
	QC10	3.72	4.17

Capacitor bank (MVar)	QC12	0.58	0.01
	QC15	0.00	0.01
	QC17	0.67	0.75
	QC 20	3.97	4.91
	QC 21	1.52	2.94
	QC 23	2.69	2.79
	QC 24	4.66	3.88
	QC 29	4.89	4.96
Generating cost (Unit/h)		**815.8582**	899.4137
Transmission loss (MW)		8.3707	**4.1514**

D.23: Control variables corresponding to extreme points, obtained by using MODA algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	**B**
P_G (MW)	P_{G1}	159.19	90.33
	P_{G2}	47.89	59.25
	P_{G5}	23.56	37.21
	P_{G8}	24.82	32.69
	P_{G11}	18.49	28.42
	P_{G13}	17.83	40.00
V (p.u)	V_{G1}	1.04	1.10
	V_{G2}	1.02	1.08
	V_{G5}	0.99	1.05
	V_{G8}	1.01	1.06
	V_{G11}	1.04	1.05
	V_{G13}	1.04	1.01
Tap (p.u)	T_{11}	0.98	1.05
	T_{12}	1.00	0.97
	T_{15}	0.98	0.97
	T_{36}	0.98	0.97
Capacitor bank (MVar)	QC10	5.00	1.59
	QC12	2.94	2.06
	QC15	2.57	2.01
	QC17	2.00	0.06
	QC 20	2.46	2.08
	QC 21	3.00	5.00
	QC 23	4.00	4.87
	QC 24	2.00	0.87
	QC 29	2.95	2.90
Generating cost (Unit/h)		**806.8893**	880.6957
Transmission loss (MW)		8.4853	**4.4897**

D.24: Control variables corresponding to extreme points, obtained by using MOSSA algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	B
P_G (MW)	P_{G1}	160.60	79.70
	P_{G2}	49.42	61.81
	P_{G5}	23.79	48.67
	P_{G8}	25.77	35.00

	P_{G11}	17.73	29.81
	P_{G13}	14.48	32.05
	V_{G1}	1.07	1.10
	V_{G2}	1.06	1.10
V (p.u)	V_{G5}	1.05	1.09
	V_{G8}	1.03	1.09
	V_{G11}	1.05	1.05
	V_{G13}	1.07	1.05
	T_{11}	1.04	1.05
Tap (p.u)	T_{12}	0.90	0.90
	T_{15}	0.96	1.02
	T_{36}	1.03	1.04
	QC10	4.99	0.93
	QC12	3.68	2.70
	QC15	3.38	3.73
	QC17	0.79	2.63
Capacitor bank (MVar)	QC 20	0.34	4.12
	QC 21	1.33	0.09
	QC 23	3.58	3.29
	QC 24	1.25	0.77
	QC 29	2.64	1.62
Generating cost (Unit/h)		**805.3594**	912.3605
Transmission loss (MW)		8.3973	**3.6307**

D.25: Control variables corresponding to extreme points, obtained by using NSGSA algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	B
	P_{G1}	173.46	55.57
	P_{G2}	46.46	75.91
P_G (MW)	P_{G5}	21.57	49.97
	P_{G8}	24.54	34.98
	P_{G11}	13.33	29.99
	P_{G13}	13.32	39.95
	V_{G1}	1.05	1.10
	V_{G2}	1.03	1.10
V (p.u)	V_{G5}	1.00	1.08
	V_{G8}	1.01	1.09
	V_{G11}	1.04	1.10
	V_{G13}	1.04	1.08
	T_{11}	1.04	1.06
Tap (p.u)	T_{12}	0.92	0.96
	T_{15}	1.01	1.02
	T_{36}	0.94	1.01
	QC10	2.93	3.44
	QC12	2.38	2.15
	QC15	3.15	5.00
	QC17	4.25	3.17
Capacitor bank (MVar)	QC 20	2.78	3.13
	QC 21	3.85	3.27
	QC 23	1.48	3.41
	QC 24	2.36	5.00
	QC 29	4.03	3.28

Generating cost (Unit/h)	**803.0399**	958.5062
Transmission loss (MW)	9.2673	**2.9666**

D.26: Control variables corresponding to extreme points, obtained by using NSABC algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	B
P_G (MW)	P_{G1}	176.45	55.69
	P_{G2}	44.91	77.16
	P_{G5}	20.46	50.00
	P_{G8}	21.56	35.00
	P_{G11}	14.76	28.93
	P_{G13}	15.03	40.00
V (p.u)	V_{G1}	1.04	1.04
	V_{G2}	1.02	1.03
	V_{G5}	0.98	1.01
	V_{G8}	1.01	1.01
	V_{G11}	1.03	1.08
	V_{G13}	1.04	1.08
Tap (p.u)	T_{11}	0.90	0.99
	T_{12}	1.05	0.93
	T_{15}	1.02	0.99
	T_{36}	0.99	0.98
Capacitor bank (MVar)	QC10	4.62	3.55
	QC12	3.07	0.00
	QC15	4.87	5.00
	QC17	3.41	4.85
	QC 20	1.05	2.00
	QC 21	1.94	4.10
	QC 23	4.55	4.81
	QC 24	3.78	4.00
	QC 29	3.91	4.19
Generating cost (Unit/h)		**804.5842**	960.1727
Transmission loss (MW)		9.7718	**3.3845**

D.27: Control variables corresponding to extreme points, obtained by using R-NSGA-II algorithm in solving OPF MOP

Control variables		**Solutions**	
		A	B
P_G (MW)	P_{G1}	177.00	60.55
	P_{G2}	48.73	70.87
	P_{G5}	21.62	50.00
	P_{G8}	21.00	34.99
	P_{G11}	11.83	30.00
	P_{G13}	12.02	40.00
V (p.u)	V_{G1}	1.09	1.08
	V_{G2}	1.07	1.08
	V_{G5}	1.04	1.06
	V_{G8}	1.05	1.07
	V_{G11}	1.10	1.10
	V_{G13}	1.10	1.10

Tap (p.u)	T_{11}	1.00	1.01
	T_{12}	0.90	0.91
	T_{15}	0.97	0.97
	T_{36}	0.95	0.96
Capacitor bank (MVar)	QC10	5.00	4.98
	QC12	5.00	4.91
	QC15	5.00	4.99
	QC17	5.00	5.00
	QC 20	4.88	4.77
	QC 21	5.00	5.00
	QC 23	4.27	3.56
	QC 24	4.98	4.91
	QC 29	3.00	2.55
Generating cost (Unit/h)		**799.7188**	949.4215
Transmission loss (MW)		8.7950	**3.0058**

Appendix E
Grey Relational Analysis

GRA of the non-dominated solutions obtained by 2S-ENDSHHMO in solving welded beam multi-objective optimization problem

No.	Objectives		Normalized Objectives		HIGHEST GRG	Rank
	Fabrication cost	Deflection of the beam	Fabrication cost	Deflection of the beam		
1	1.9048	0.0103	0.9966	0.1890	0.687338	62
2	1.8951	0.0103	0.9969	0.1854	0.687101	63
3	1.9116	0.0102	0.9964	0.1912	0.687453	61
4	1.9211	0.0102	0.9961	0.1979	0.688161	60
5	1.8799	0.0106	0.9974	0.1621	0.684235	64
6	1.9364	0.0101	0.9957	0.2034	0.688529	59
7	1.8487	0.0107	0.9983	0.1505	0.683551	65
8	1.9613	0.0100	0.9949	0.2114	0.688994	58
9	1.8331	0.0115	0.9988	0.0870	0.675696	66
10	1.9701	0.0099	0.9947	0.2197	0.690004	57
11	1.9802	0.0098	0.9944	0.2247	0.690462	56
12	1.8100	0.0117	0.9995	0.0744	0.674819	67
13	1.9956	0.0098	0.9939	0.2304	0.690905	55
14	2.0059	0.0097	0.9936	0.2383	0.691835	54
15	1.7916	0.0126	1.0000	0.0000	0.666667	100
16	2.0219	0.0096	0.9931	0.2433	0.692149	53
17	2.0351	0.0095	0.9927	0.2488	0.692651	52
18	2.0444	0.0095	0.9925	0.2514	0.692799	51
19	2.0502	0.0095	0.9923	0.2531	0.692901	50
20	2.0655	0.0094	0.9918	0.2611	0.693754	49
21	2.0864	0.0093	0.9912	0.2685	0.694356	48
22	2.0989	0.0092	0.9908	0.2731	0.69477	47
23	2.1469	0.0089	0.9894	0.3016	0.698231	45
24	2.2055	0.0088	0.9877	0.3091	0.697874	46
25	2.2202	0.0086	0.9872	0.3269	0.700646	43
26	2.2336	0.0085	0.9868	0.3325	0.701279	42
27	2.2917	0.0085	0.9851	0.3371	0.700492	44
28	2.3034	0.0084	0.9847	0.3454	0.701704	41
29	2.3165	0.0083	0.9843	0.3484	0.701912	40
30	2.3310	0.0081	0.9839	0.3642	0.70452	38
31	2.3614	0.0081	0.9830	0.3707	0.704936	37
32	2.3885	0.0081	0.9822	0.3710	0.704241	39
33	2.4550	0.0078	0.9802	0.3957	0.707341	36
34	2.4759	0.0076	0.9796	0.4065	0.709002	35
35	9.0277	0.0018	0.7841	0.8858	0.756248	22
36	9.0345	0.0018	0.7839	0.8859	0.756227	23
37	8.9369	0.0018	0.7868	0.8845	0.756744	18
38	9.1197	0.0018	0.7814	0.8876	0.756097	27
39	8.8474	0.0019	0.7895	0.8831	0.757113	12
40	8.7698	0.0019	0.7918	0.8814	0.75716	9
41	8.7361	0.0019	0.7928	0.8804	0.757017	13
42	9.3807	0.0017	0.7736	0.8920	0.755318	31
43	9.4076	0.0017	0.7728	0.8921	0.754998	34
44	8.5745	0.0019	0.7976	0.8779	0.757776	5
45	8.5657	0.0019	0.7979	0.8778	0.757894	3
46	9.4416	0.0017	0.7718	0.8928	0.755006	33

47	**8.5575**	**0.0019**	**0.7981**	**0.8778**	**0.757986**	**1**
48	8.5376	0.0019	0.7987	0.8773	0.757964	2
49	9.1314	0.0018	0.7810	0.8879	0.756119	26
50	9.3180	0.0018	0.7754	0.8911	0.75561	29
51	8.9690	0.0018	0.7859	0.8855	0.756873	14
52	8.9523	0.0018	0.7864	0.8850	0.756843	17
53	9.0512	0.0018	0.7834	0.8863	0.756206	24
54	9.0646	0.0018	0.7830	0.8866	0.756251	21
55	8.9112	0.0018	0.7876	0.8841	0.756855	15
56	9.0946	0.0018	0.7821	0.8876	0.75644	20
57	8.8275	0.0019	0.7901	0.8829	0.757274	7
58	9.1956	0.0018	0.7791	0.8893	0.756127	25
59	8.8017	0.0019	0.7908	0.8821	0.75712	10
60	8.7866	0.0019	0.7913	0.8818	0.757162	8
61	9.2212	0.0018	0.7783	0.8896	0.756012	28
62	9.2837	0.0018	0.7765	0.8900	0.755356	30
63	8.6969	0.0019	0.7940	0.8800	0.757301	6
64	9.3522	0.0018	0.7744	0.8913	0.75529	32
65	8.6194	0.0019	0.7963	0.8789	0.757787	4
66	8.9788	0.0018	0.7856	0.8856	0.756844	16
67	8.8017	0.0019	0.7908	0.8821	0.75712	11
68	8.9857	0.0018	0.7854	0.8856	0.756743	19
69	35.3087	0.0004	0.0000	1.0000	0.666667	101
70	35.2989	0.0004	0.0003	1.0000	0.666662	102
71	35.2487	0.0004	0.0018	0.9999	0.666778	99
72	35.2160	0.0004	0.0028	0.9999	0.666867	98
73	35.1778	0.0004	0.0039	0.9999	0.666953	97
74	35.1681	0.0004	0.0042	0.9998	0.666972	96
75	35.1435	0.0004	0.0049	0.9998	0.666993	95
76	35.0485	0.0004	0.0078	0.9997	0.667258	94
77	34.8762	0.0004	0.0129	0.9995	0.667634	93
78	34.8199	0.0004	0.0146	0.9995	0.667781	92
79	34.7996	0.0004	0.0152	0.9994	0.667806	91
80	34.7475	0.0004	0.0167	0.9994	0.667915	90
81	34.7377	0.0004	0.0170	0.9993	0.667916	89
82	34.6743	0.0004	0.0189	0.9993	0.668107	88
83	34.5779	0.0004	0.0218	0.9992	0.668332	87
84	34.5661	0.0004	0.0222	0.9992	0.668333	86
85	34.5169	0.0004	0.0236	0.9991	0.668477	85
86	34.4712	0.0005	0.0250	0.9991	0.668584	84
87	34.3604	0.0005	0.0283	0.9990	0.668833	83
88	34.3488	0.0005	0.0286	0.9990	0.668873	82
89	34.3113	0.0005	0.0298	0.9989	0.668941	81
90	34.1357	0.0005	0.0350	0.9987	0.66935	80
91	34.0809	0.0005	0.0366	0.9987	0.6695	79
92	33.9964	0.0005	0.0392	0.9985	0.669654	78
93	33.9720	0.0005	0.0399	0.9985	0.669737	77
94	33.9422	0.0005	0.0408	0.9985	0.669827	76
95	33.9140	0.0005	0.0416	0.9984	0.669857	75
96	33.9090	0.0005	0.0418	0.9984	0.669863	74
97	33.7901	0.0005	0.0453	0.9983	0.670184	73
98	33.7826	0.0005	0.0455	0.9983	0.670198	72
99	33.7614	0.0005	0.0462	0.9983	0.670217	71
100	33.4669	0.0005	0.0550	0.9979	0.670923	70