A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION FOR CONTINUOUS OPTIMIZATION USING A STEP-FUNCTION TECHNIQUE

CHUAH HOW SIANG

UNIVERSITI SAINS MALAYSIA

2022

A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION FOR CONTINUOUS OPTIMIZATION USING A STEP-FUNCTION TECHNIQUE

by

CHUAH HOW SIANG

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

May 2022

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my main supervisor, Dr Wong Li Pei for giving me support and guiding me towards the fulfilment of my dream in this study. Even on the darkest day of the research study, Dr Wong has been guiding me passionately. His professionalism and enthusiasm in this journey have inspired me to become a better researcher. I would also like to thank my co-supervisor, Dr Fadratul Hafinaz Hassan, who gives me insight in the study of the research areas.

A very special gratitude goes out to all staff members of the School of Computer Sciences. The school has been kind and helpful in arranging the lab availability and research meetings amidst the pandemic of COVID-19. Besides that, I would like to acknowledge Universiti Sains Malaysia Fellowship Scheme for giving me financial support.

Lastly, I would also like to thank everyone that has given me emotional support and encouragement throughout my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

EA	Evolutionary Algorithm
MOEA	Multi-objective Evolutionary Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm Based on Decomposition
MOEA/D-AWA	Multi-objective Evolutionary Algorithm Based on Decomposition with Adaptive Weight Adjustment
MOEA/D-AWACD	A Constant-Distance Based Neighbors for Multi-objective Evolutionary Algorithm Based on Decomposition with Dynamic Weight Adjustment
MOEA/D-DRA	Multi-objective Evolutionary Algorithm Based on Decomposition with Dynamic Resource Allocation
MOEA/D-GRA	Multi-objective Evolutionary Algorithm Based on Decomposition with Generalized Resource Allocation
USM	Universiti Sains Malaysia

ALGORITMA EVOLUSI PELBAGAI OBJEKTIF BERDASARKAN PENGURAIAN UNTUK PENGOPTIMUMAN SELANJAR MENGGUNAKAN TEKNIK FUNGSI LANGKAH

ABSTRAK

Pengoptimuman pelbagai objektif merupakan satu bidang kajian yang menyelesaikan masalah kompleks dunia sebenar melibatkan dua atau tiga objektif. Algoritma Evolusi Pelbagai Objektif Berdasarkan Penguraian (MOEA/D) adalah salah satu algoritma menggunakan konsep penguraian dan kejiranan untuk menyelesaikan masalah pelbagai objektif. Salah satu algorithm MOEA/D yang terkini, iaitu *Constant-distance based Neighbours for MOEA/D with Dynamic Weight Vector Adjustment* (MOEA/D-AWACD), mengintegrasikan konsep kejiranan berdasarkan jarak malar dan reka bentuk vektor berat dinamik. Gabungan ini membolehkan kejiranan fleksibel yang dapat menyesuaikan diri dengan perubahan vektor berat. Namun, prestasi MOEA/D-AWACD bergantung pada satu parameter jarak tetap, δ, yang melaraskan saiz kejiranan. Untuk mendapatkan nilai parameter δ yang sesuai, algoritma perlu dilaksanakan beberapa kali secara berasingan. Situasi ini mengusulkan δMOEA/D-AWACD, yang menggunakan satu fungsi-langkah yang mengawal parameter δ secara deterministik dalam sesuatu pelaksanaan. δMOEA/D-AWACD mengatasi MOEA/D, MOEA/D-AWACD, *MOEA/D with Adaptive Weight Adjustment* (MOEA/D-AWA), dan *Non-dominated Sorting Genetic Algorithm II* (NSGA-II). Algoritma ini juga setanding dengan NSGA-III and *Strength Pareto Evolutionary Algorithm 2* (SPEA2). Selain itu, MOEA/D sering dilengkapi dengan mekanisme peruntukan sumber pengiraan dan strategi pemilihan. Salah satu jurang penyelidikan adalah kekurangan satu kajian yang menyeluruh berkenaan dengan gabungan mekanisme peruntukan

sumber pengiraan dan strategi pemilihan. Oleh itu, sifat penumpuan (iaitu kelajuan penumpuan dan penumpuan pra-matang) dan prestasi empat strategi pemilihan yang berbeza (iaitu Pemilihan Kejohanan, Pemilihan Berkadar Kecergasan, Pemilihan Pemeringkatan Linear dan Pemilihan Pemeringkatan Eksponen) terhadap lapan mekanisme peruntukan sumber berbeza dalam δMOEA/D-AWACD telah dikaji. Keputusan-keputusan eksperimen menunjukkan δMOEA/D-AWACD dengan peruntukan sumber dinamik dan pemilihan kejohanan mengatasi gabungan lain. Pemilihan kejohanan menunjukkan fenomena penumpuan perlahan apabila ia dilaksanakan dengan menggunakan masalah-masalah ujian ZDT (iaitu ZDT1, ZDT2, ZDT3, ZDT4, dan ZDT6) dalam gabungan dengan peruntukan sumber dinamik. Walau bagaimanapun, dalam kebanyakan kes, gabungan pemilihan johan dengan peruntukan sumber dinamik menunjukkan prestasi yang lebih baik daripada strategi pemilihan lain. Sebaliknya, pemilihan berkadar kecergasan, pemilihan pemeringkatan linear dan pemilihan pemeringkatan eksponen mengalami penumpuan pra-matang dalam masalah ZDT2 apabila mereka digunakan bersama dengan mekanisme peruntukan sumber dinamik. Satu kes kajian dunia sebenar mengenai masalah fotovoltaik solar pelbagai objektif juga dijalankan. δMOEA/DAWACD memperoleh nilai objektif yang jauh lebih baik (p-value = 0.0001) untuk ketiga-tiga objektif merentasi semua jenis modul fotovoltaik solar di Bandar Sungai Long, Bangi dan Bayan Lepas berbanding dengan NSGA-II.

A MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION FOR CONTINUOUS OPTIMIZATION USING A STEP-FUNCTION TECHNIQUE

ABSTRACT

Multi-objective optimization is an area of study which solves complex real-world problem that involves two or three objectives. Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) is one of the algorithms that utilize the concepts of decomposition and neighbourhood to solve multi-objective problems. One of the recent MOEA/D algorithms, i.e., Constant-distance based Neighbours for MOEA/D with Dynamic Weight Vector Adjustment (MOEA/D-AWACD), integrates the concept of a constant-distance neighbourhood and a dynamic weight vector design. This combination creates a flexible neighbourhood that can adapt to the weight vectors changes. However, MOEA/D-AWACD's performance is dependent on a constant-distance parameter, δ, that adjusts the neighbourhood size. To obtain an appropriate value of parameter δ, multiple and separate algorithm executions are required. This leads to a proposal of δMOEA/D-AWACD, which employs a step-function to deterministically control the parameter δ within a single algorithm execution. The proposed δMOEA/D-AWACD was tested on 18 continuous optimization benchmark problems, and it statistically outperforms MOEA/D, MOEA/D-AWACD, MOEA/D with Adaptive Weight Adjustment (MOEA/D-AWA), and Non-dominated Sorting Genetic Algorithm II (NSGA-II). It is also on par with NSGA-III and Strength Pareto Evolutionary Algorithm 2 (SPEA2). On the other hand, MOEA/D is often equipped with a computational resource allocation mechanism and a selection strategy. One of the research gaps is the lack of a detailed study on the combination of a computational

resource allocation mechanism and a selection strategy. Thus, the convergence properties (i.e., convergence speed and premature convergence) and performance of four different selection strategies (i.e., Tournament Selection, Fitness Proportionate Selection, Linear Ranking Selection, and Exponential Ranking Selection) on eight different resource allocation mechanisms in δMOEA/D-AWACD is explored. The experimental results show that δMOEA/D-AWACD with dynamic resource allocation and tournament selection outperforms other combinations. Tournament selection shows a slow convergence phenomenon on the ZDT test problems (i.e., ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6) when it is bundled with dynamic resource allocation. However, this combination can significantly perform better than other selection strategies in most cases. On the other hand, fitness proportionate selection, linear ranking selection, and exponential ranking selection experience premature convergence in the ZDT2 test problem when they are used in conjunction with the dynamic resource allocation mechanism. A real-world study case on a multi-objective solar photovoltaic problem is also conducted. δMOEA/DAWACD obtains significantly better (p-value = 0.0001) objective values for all three objectives across all the types of solar photovoltaic modules in Bandar Sungai Long, Bangi and Bayan Lepas compared to NSGA-II.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Optimization is a method that aims to identify optimal solution(s) from a finite set of solutions (Choong et al., 2019). The optimal solution(s) is identified using an objective function. Optimization can be categorized into two different categories, i.e., continuous optimization and discrete optimization. Continuous optimization mainly deals with continuous variables which they are allowed to take on any value within a range of values (i.e., usually real numbers). In applied mathematics, a set of test functions for continuous optimization are available such as the Rosenbrock function, Ackley function and Sphere function. On the contrary, discrete optimization work with a permutation set of discrete variables (e.g., integers). Optimization can be used to solve problems that involve more than one objective.

Multi-objective optimization problems (MOPs) are problems that involve two or three objectives. When more than three objectives are involved, it is known as many-objectives optimization problems (MaOPs) (Su et al., 2019). When an objective is improved, other objectives will be degraded. For instance, when someone is buying a car, she/he wishes to get a car with a low accident rate and low monetary cost. Assuming the accident rate decreases with the increment of monetary cost, a car with a low accident rate and low monetary cost will not be available. Thus, compromises will need to be made by considering other options. These options are called trade-off solutions. However, the number of trade-off solutions can be huge. As such, multi-objective optimization (MOO) can be utilized to obtain a set of trade-off solutions. In MOO, a single solution that simultaneously optimizes each objective does not exist. Instead, a set of Pareto-optimal solutions are generated to provide a range of choices for decision-makers. Generally, MOO focuses on progressing towards the Pareto-optimal set with a widely spread distribution of solutions. It is one of the challenges in MOO to ensure convergence to the true Pareto-optimal solutions with a wide diversity among the solutions. Pareto-optimal solutions are also known as non-dominated solutions in MOP. The key background of Pareto dominance is described as follows:

- A solution is said to dominate another solution if it is not inferior to another solution in all objectives and the solution is strictly better than another solution in at least one objective.

- If a solution dominates another solution, it is better than another solution.

Figure 1.1 shows an example of optimization of two objectives, f_1 and f_2, i.e., maximization of f_1 and minimization of f_2. Based on the concept of Pareto dominance, 3 and 5 are the non-dominated solutions while 1, 2, and 4 are dominated solutions.

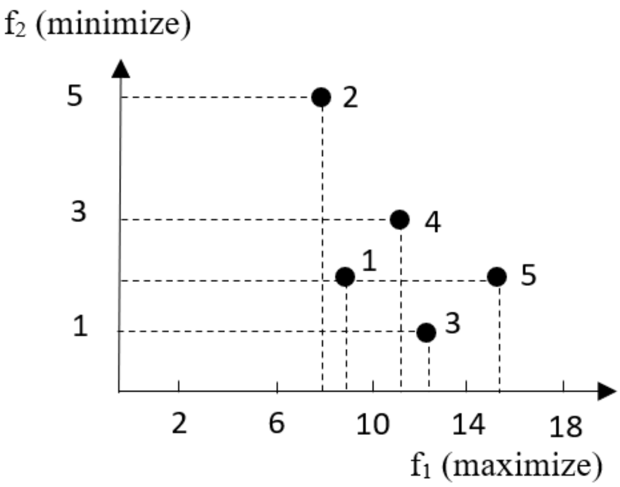

Figure 1.1 Non-Domination Sorting Concept (Deb, 2001)

These non-dominated solutions (i.e., 3 and 5) are also known as Pareto-optimal solutions and form a Pareto front. The concept of Pareto-optimality originates from Vilfredo Pareto (Pareto, 1935). It represents the state of allocation of resources whereby it is not possible to improve one objective without making another objective worse off (Deb, 2001).

In MOO, a decision-maker plays an important role. The MOO methods can be generally classified into three methods (i.e., a priori, a posteriori, and interactive) based on decision making approaches (Purshouse et al., 2014). If a decision-maker knows the weightage of each objective before the optimization process, a priori methods such as the weighted sum method (Marler & Arora, 2010) and goal programming (Zhuang & Hocine, 2018) can be employed to obtain a single optimum solution. If a decision-maker desires to select a solution from a list of solutions, a posteriori methods such as Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) (Zhang et al., 2009), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002), and Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001) can be utilized. In a posteriori MOO, diversity and convergence of the solutions set are important. However, most of the time, it is a challenge to obtain highly diverse and highly converged solutions due to the conflicting relationship between exploration and exploitation in an optimization algorithm. The third method is the interactive method, where a decision-maker can continuously provide feedback on the optimization to obtain desirable solutions. For instance, the NIMBUS method (Miettinen & Mäkelä, 2000) is categorized as an interactive method.

One of the a posteriori methods is the Multi-objective Evolutionary Algorithm (MOEA). Recent advances in MOEA include Non-dominated Sorting Genetic Algorithm III (NSGA III) (Deb & Jain, 2014), Multi-objective Evolutionary Algorithms based on Decomposition (MOEA/D) (Zhang & Li, 2007), and Hypervolume-Based Many-Objective Optimization Algorithm (HypE) (Bader & Zitzler, 2011). Among these variants, MOEA/D is one of the widely studied algorithms that utilize the concept of decomposition and neighbourhood to solve MOPs. MOEA/D is a population-based metaheuristic that utilizes a number of individuals in its optimization mechanism. It divides a MOP into several subproblems using a scalarization technique based on evenly spaced weight vectors. The weight vectors act as the weightage for each objective. As a result of scalarization, each subproblem is a single objective problem. Each subproblem is assigned with a distinct weight vector set. Each subproblem is also associated with each individual in the population of MOEA/D. In other words, subproblem, weight vector and individual refer to the same entity in MOEA/D. Each individual has its own neighbourhood. The neighbourhood is made up of the nearest *k* number of individuals. The distance between the individuals is calculated based on Euclidean distance. Moreover, parents are selected from the neighbourhood during reproduction to create new individuals for each subproblem. After a new individual is produced, it is then broadcast within the neighbourhood and replaces other individuals in the neighbourhood if the new individual is better. In recent years, various strategies (e.g., decomposition approaches, weight vector design, neighbourhood design, mating and replacement, computational resource allocation and reproduction operations) have been proposed to enhance MOEA/D (Trivedi et al., 2016; Ooi, 2019).

1.2 Problem Statement

MOEA/D divides the search space using weight vectors as the guide for each subproblem (Zhang & Li, 2007). These weight vectors are generated using a simplex lattice design and spread out uniformly across the search space. The way of defining the guide (i.e., the weight vectors) is suitable for MOPs with convex shapes MOP. However, in the case of non-convex Pareto shape MOP such as disconnected shape

Pareto front MOP, large gaps can be observed along the Pareto front. Thus, the predefined and uniform weight vectors are no longer suitable to act as the guide for the search process. This problem is addressed by MOEA/D with Adaptive Weight Adjustment (MOEA/D-AWA) (Qi et al., 2014), which adaptively alters the weight vector by considering the sparsity level. Sparsity level measures are based on the number of individuals in close proximity to a chosen individual. The lower the sparsity level, the larger the number of individuals in close proximity with a chosen individual. MOEA/D-AWA removes the individual with a low sparsity level and adds a new individual with a high sparsity level. This allows MOEA/D-AWA to obtain good performance in non-convex Pareto MOP. However, when the weight vector changes, the neighbourhood size in MOEA/D-AWA remains constant. In MOEA/D-AWA, the neighbourhood is created based on the weight vector only once during the initialization. Thus, if the weight vector changes during the optimization process, the existing neighbourhood becomes not suitable for the new weight vectors. This issue is handled by constant-distance based neighbours for multi-objective evolutionary based on decomposition with dynamic weight vector adjustment (MOEA/D-AWACD) (Ooi, 2019). The algorithm integrates the constant-distance concept from Improved MOEA/D with g-Tournament Selection (improved MOEA/D-gTS) (Sato, 2015) into MOEA/D-AWA. Hence, it enables the neighbourhood size to change accordingly when the adaptive weight adjustment from MOEA/D-AWA changes the weight vectors throughout an algorithm execution. However, the performance of MOEA/D-AWACD relies on a parameter, $\delta = \sqrt{2}\frac{h}{H}$, which is used as a threshold to control the neighbourhood size. h is the range of $[0, H]$ and H is a user defined parameter to adjust the number of subproblems. When the value of δ is small, the neighbourhood size is also small. In order to obtain a suitable δ setting, MOEA/D-AWACD requires multiple and separate executions, which is time consuming. For example, when MOEA/D-AWACD is used to solve a MOP that involves two objectives with a user-defined parameter H = 99, there will be 100 sets of experiments with different δ values to be separately executed.

Apart from this, it is noticed that in the area of MOEA/D, a comprehensive study about the bundling effect of a selection strategy and a computational resource allocation mechanism is lacking. The computational resource allocation mechanism mainly works with a utility function to compute a series of values for each individual. These series of values are used as fitness in the mating selection phase of MOEA/D. From here, it can be observed that the computational resource allocation mechanism tightly connects with a selection strategy since the output values of the utility function act as the input of the selection strategy. There are papers that describe the computational resource allocation mechanism in MOEA/D. For instance, MOEA/D with Dynamic Resource Allocation (MOEA/D-DRA) includes dynamic resource allocation and tournament selection (Zhang et al., 2009). Another instance is MOEA/D with Generalized Resource Allocation (MOEA/D-GRA) which includes a set of generalized resource allocation mechanisms and fitness proportionate selection (Zhou & Zhang, 2015). In these papers, the effect of bundling a resource allocation mechanism with a selection strategy is not investigated in detail. It is believed that the use of different selection strategies combined with different computational resource allocation mechanisms will affect the convergence of MOEA/D. With a careless combination of a selection strategy and a resource allocation mechanism, premature convergence and slow convergence may occur. Premature convergence occurs when an algorithm converges fast at the beginning of an algorithm execution and becomes significantly worse than other algorithms at the end of algorithm execution. On the other hand, slow convergence happens when an algorithm converges slower than other algorithms at the beginning of an algorithm execution and does not show any significant difference at the end of algorithm execution.

1.3 Research Questions

The research questions of this research can be summarized as s follow:

1. How to minimize the number of runs for MOEA/D-AWACD to obtain an appropriate setting of δ?

2. What are the influence and convergence properties of various selection strategies on the computational resource allocation mechanism of the proposed algorithm?

3. How does the proposed algorithm perform in a real-world problem?

1.4 Research Objectives

Based on the research questions stated in Section 1.3, the research objectives are formed as follows:

1. To propose a modified MOEA/D-AWACD that can control the setting of the constant-distance parameter, δ.

2. To investigate the performance and convergence properties of various selection strategies combined with different computational resource allocation mechanisms.

3. To evaluate the practicality of the proposed algorithm using a real-world multi-objective optimization problem.

1.5 Research Methodology

The research methodology of this thesis consists of two main phases namely: algorithm design & development and experiment design & evaluation as shown in Figure 1.2.

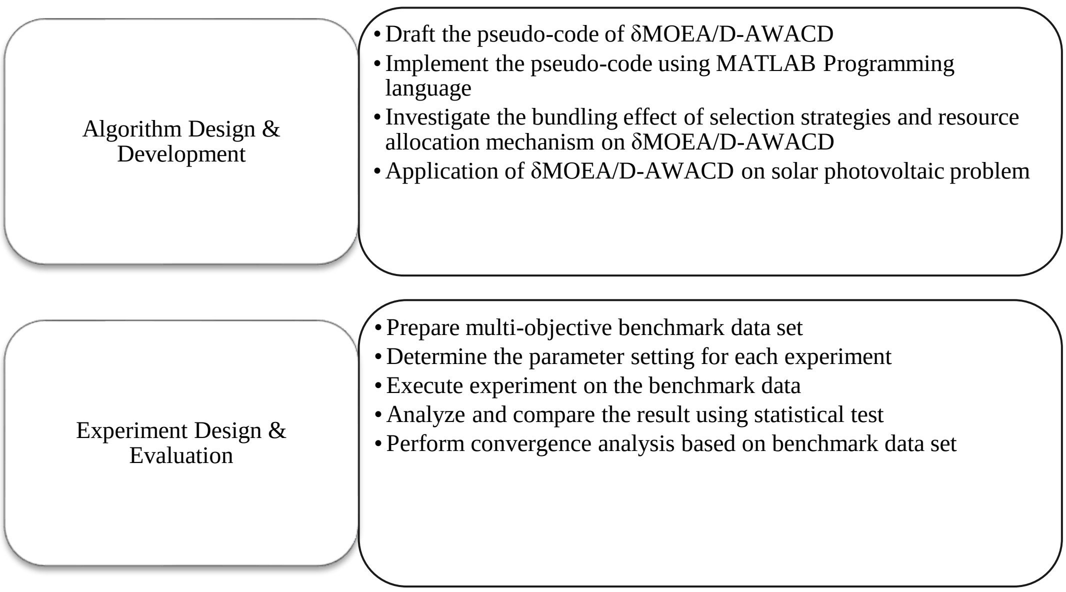

Figure 1.2 Research Methodology

In the algorithm design & development phase, a modified MOEA/D-AWACD is proposed to solve a research problem related to parameter tuning as described in Section 1.2. The proposed algorithm is named Delta Constant-distance based Neighbourhoods for Multi-objective Evolutionary Algorithm based on Decomposition with Dynamic Weight Vector Adjustment (δMOEA/D-AWACD). The proposed δMOEA/D-AWACD utilizes a step function deterministic parameter control to deterministically adjust the δ settings, which is used to define the constant-distance neighbourhood. Besides this, the bundling effect of computational resource allocation and selection strategy is also investigated during this phase. The convergence properties (i.e., premature convergence and convergence speed) are also studied based on experiments that involve various selection strategies on different computational resource allocation mechanisms. After that, the proposed algorithm is applied to solve a real-world case study, i.e., the solar photovoltaic problem.

In the experiment design & evaluation phase, a total of 18 continuous test problems are gathered. The test problems are from the ZDT, UF, and DTLZ test suites, which can be accessed from the Platform for Evolutionary Multi-Objective Optimization, PlatEMO (Tian et al., 2017). A comparison study is then carried out to benchmark the proposed algorithm with existing multi-objective algorithms in continuous optimization. Wilcoxon rank-sum test with a significance level of 5% is utilized to support the quantitative decision in assessing the performance of the proposed algorithm against the benchmark algorithms (Wilcoxon et al., 1970). In the real-world case study, the best set of objectives is selected with respect to the best first objective value (i.e., power conversion efficiency) and then benchmark against MmGA and NSGA-II results (Seera et al., 2021). The results were compared using a sign test (Derrac et al., 2011) with a significance level of 5%.

1.6 Research Scope

This study focuses on modifying the MOEA/D-AWACD algorithm to deterministically control the setting of the δ parameter. In the comparison study, a set of continuous optimization problems with two objectives from ZDT, UF, and DTLZ test suites are employed. The aim is to cover different types of Pareto fronts such as the front with convex, concave, linear, and disconnected patterns. The decision variables of all the selected problems are bounded to a specific range of values (Refer to Appendix Table A.1). This serves as the constraints considered in this research.

Besides, in the study on the influence of selection strategy on computational resource allocation, four selection strategies and eight computational resource allocation mechanisms are studied. The selection strategies are tournament selection, fitness proportionate selection, linear ranking, and exponential ranking, respectively. The computational resource allocation mechanism covers dynamic resource allocation and seven different combinations of utility functions based on generalized resource allocation. A study of convergence performance based on the selection strategy is also discussed. The study of convergence properties of the proposed algorithm comprises two aspects, namely: premature convergence and convergence speed.

Furthermore, this research also tests the practical applicability of the proposed algorithm to a solar photovoltaic problem. The proposed algorithm is benchmarked against the original work (Seera et al., 2021).

1.7 Research Contributions

There are three research contributions in this research. First, this research introduces a step function deterministic parameter control mechanism in MOEA/D-AWACD. The modified MOEA/D-AWACD is denoted by δMOEA/D-AWACD. The aim of the step function deterministic parameter control mechanism is to deterministically control the settings of δ within one algorithm execution. The advantage of this parameter control mechanism is to eliminate the tedious step of tuning δ via multiple and separate algorithm executions. Besides, the proposed δMOEA/D-AWACD inherits a dynamic neighbourhood design that adapts to the changes in weight vectors. This mechanism ensures that neighbourhood size will be dynamically changed throughout an algorithm execution. This allows the search process to be more flexible by reacting to the changes in the search environment.

The second research contribution is to provide a study related to the bundling effect of different selection strategies on different computational resource allocation mechanisms. In the proposed δMOEA/D-AWACD, the computational resource allocation mechanism is tightly bundled with a selection strategy since the output from the computational resource allocation mechanism will act as the input of the selection strategy. In order to study the bundling effect between these two, four selection strategies on eight different computational resource allocation mechanisms are employed. A total of 32 different combinations of selection strategy and computational resource allocation mechanism are presented in Chapter 5. The performance of the combination of four selection strategies on eight different computational resource allocation mechanisms is presented. The convergence properties (i.e., premature convergence and convergence speed) are discussed.

Lastly, the third research contribution is to provide an application of the proposed algorithm to solve real-world problems, i.e., solar photovoltaic problems. The solar photovoltaic problem comprises of maximization of power conversion efficiency, minimization of photovoltaic panel weight per output power, and photovoltaic panel area per output power. The performance of the proposed algorithm as presented in Chapter 6 indicates that it is feasible to solve real-world problems and also outperforms the algorithm in the previous work (Seera et al., 2021).

1.8 Thesis Organization

This thesis comprises seven chapters. The first chapter introduces the research by discussing the problem statement, objectives, scope, and research contribution of this research.

Chapter 2 reviews the related background study of this research. The techniques and advancement of multi-objective optimization are discussed. In this chapter, the research also studies MOEA/D in six different components by reviewing the current state-of-the-art and its limitation. At the end of the chapter, the research gap is discussed.

Chapter 3 discusses the research methodology used in this study. Three main methodological procedures are used to perform the research study namely, problem analysis, algorithm design and development, and experiment design and evaluation.

Chapter 4 presents the proposed algorithm δ MOEA/D-AWACD. The neighbourhood design of the proposed algorithm and the pseudocode is explained. In this chapter, comparison studies with benchmark algorithms are also presented.

Chapter 5 describes the influence of various selection strategies on computational resource allocation of δ MOEA/D-AWACD. The convergence performance of different selection strategies is discussed and the comparison studies among the different combinations of selection strategies and computational resource allocation are analyzed.

Chapter 6 discusses the application of the proposed algorithm to a real-world study case, i.e., a solar photovoltaic problem. The proposed algorithm is also benchmarked against MmGA and NSGA-II of the previous work.

Chapter 7 concludes this research. The findings in this research are summarized and potential future direction is discussed.

CHAPTER 2

LITERATURE REVIEW

This section presents related concepts, existing techniques, and the current state of art in detail. The flow of the literature review is as shown in Figure 2.1. First, multi-objective optimization (MOO) is explored. The technique and categorization of MOO are explained. After that, the parameter control mechanism is reviewed. Consecutively, the real-world application of MOO is discussed. Next, multi-objective algorithms based on decomposition (MOEA/D) are analyzed based on its six components. Finally, a summary of the gap analysis is carried out to discuss the direction of this research.

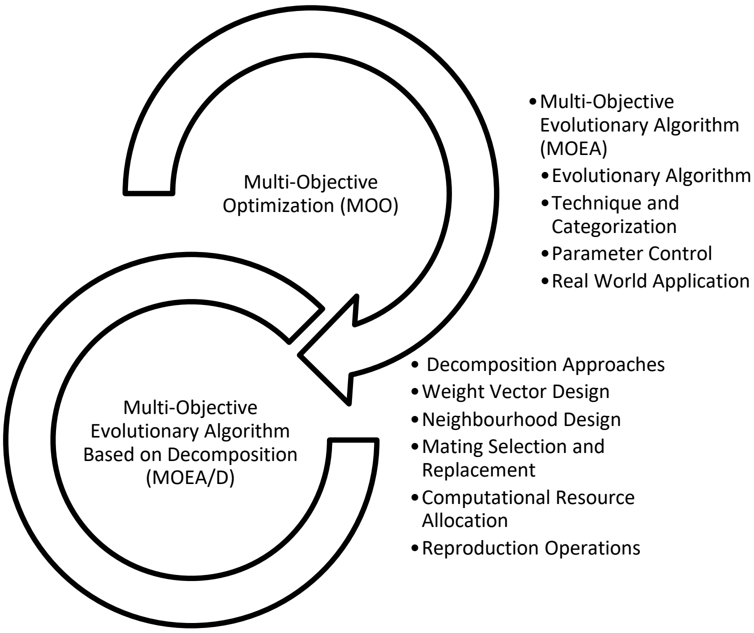

Figure 2.1 Flow of Literature Review

2.1 Multi-Objective Optimization (MOO)

Multi-objective optimization is commonly used to find a set of suitable solutions when there are multiple objectives during the decision-making process (Deb, 2001). Problems that involve two and three objectives are commonly known as multi-objective problems (MOPs). Multi-objective optimization approaches typically revolve around decision making preferences. It can be classified into three categories based on decision-making preferences (Purshouse et al., 2014) as shown in Table 2.1. Although there are three methods of decision-making preferences in multi-objective optimization, each of them comes with its benefits and issues.

A priori methods require decision-makers to define the weightage for each objective before optimization to generate a solution that is desired. Despite this advantage, a priori methods will be difficult if the decision-maker does not have any understanding of his preferences, relationships and dependencies of the objectives or feasible objective values. Examples of the algorithm in this category are the lexicographic method (Stanimirovic, 2012), goal programming (Zhuang & Hocine, 2018) and weighted sum (Marler & Arora, 2010).

A posteriori method allows decision making after the optimization process. This method category is studied extensively to find a set of Pareto-optimal solutions. This method is more complex than the A Priori method as it needs to balance convergence and diversity in its solution to provide a wide range of solutions. Most of the algorithm in this category is based on Evolutionary Algorithm (EA) such as Vector Evaluated Genetic Algorithm (VEGA) (Mao et al., 2012), Strength Pareto Evolutionary Algorithm 2 (SPEA2) (He et al., 2017), Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang et al., 2020), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Cai et al., 2019), and NSGA-III (Cui et al., 2019).

On the other hand, interactive methods allow decision-makers to tweak the objective values in several iterations of optimization to scope down to the desired solutions. The performance of interactive methods is hard to compute as it involves the decision maker's interference from time to time. Examples of the algorithm in this category include the NIMBUS method (Saini et al., 2020) and the GUESS method (Zhou-Kangas et al., 2017).

Table 2.1 Multi-Objective Optimization Approaches Based on Decision Making.

Methods	**Decision Maker Characteristics**	**Examples**
A Priori	The decision-maker defines the preferences parameter before optimization	• Lexicographic Method (Stanimirovic, 2012) • Goal Programming (Zhuang & Hocine, 2018) • Weighted Sum (Marler & Arora, 2010)
A Posteriori	The decision-maker decide on a set of solutions after optimization	• VEGA (Schaffer, 1985) (Mao et al., 2012) • MOEA/D (Zhang & Li, 2007; Li et al., 2019; Zhang et al., 2020) • SPEA2 (Zitzler et al., 2001; He et al., 2017) • NSGA-II (Deb et al., 2002; Cai et al., 2019) • NSGA-III (Deb & Jain, 2014; Cui et al., 2019)
Interactive Method	The decision-maker can define parameters before and after optimization iteratively until the obtained solution is desirable.	• NIMBUS Method (Miettinen & Mäkelä, 1995; Zhou-Kangas et al., 2017) • GUESS Method (Buchanan, 1997; Saini et al., 2020)

Multi-objective Evolutionary Algorithm (MOEA) in the a posteriori category is one of the most widely studied research. This is reviewed in detail in Section 2.1.1.

2.1.1 Multi-Objective Evolutionary Algorithm (MOEAs)

A multi-objective evolutionary algorithm is built upon EA. EA is a population-based stochastic optimization algorithm (Zitzler et al., 2004). It mimics the reproduction mechanism of biological evolution at the chromosome level. The population in EA contains a set of individuals and each individual indicates a potential solution by the end of the optimization process. The population are usually randomly generated at the beginning and each of the individuals in the population is assigned a fitness function to determine its survival probability. The optimization process then utilizes mating selection, and population management mechanisms such as mutation and crossover to select a new set of individuals as the potential solution. The process is repeated until the stopping criterion is met. The pseudocode is as shown in Algorithm 1.

Algorithm 1 Evolutionary Algorithm (Zitzler et al., 2004)

Evolutionary Algorithm (EA)

Input: Population Size
Output: Set of Individuals Initialize population
while stopping condition are not met **do**
    Evaluate individual fitness
    Select mating individuals
    Apply Population Management Mechanism
    - Crossover
    - Mutation

    Replace old population
end while

Examples of evolutionary algorithms are genetic algorithm (GA), evolutionary programming (EP) and evolution strategy (ES). All of them share the same concept of EA with some small differences. For example, GA usually represents the solution in binary and applies recombination and mutation mechanism in the searching process whereas ES usually works with real number vectors and the mutation mechanism is usually emphasized. Similarly, EP follows the concept of EA but does not use any crossover mechanism.

MOEAs inherit the reproduction mechanism from EA and changes are made to the fitness assignment to allow the consideration of multiple objectives. Additionally, some MOEAs also utilize preservation mechanisms or elitism to keep the non-dominated solution from being lost (Deb, 2001). A list of MOEA based algorithms is reviewed and classified in the next section.

2.1.2 Techniques and Categorization of MOEA-based algorithm

In the early stage of MOEA development, the techniques proposed focused more on the way to enable multi-objective solutions. For example, VEGA is the first MOEA that uses subpopulations to evaluate different objectives equally (Schaffer, 1985). The population in VEGA is partitioned into two smaller groups of individuals (i.e., subpopulation). The first subpopulation computes fitness values based on objective one while the second subpopulation computes the fitness values based on objective two.

Later, the diversity problem was discovered as the VEGA solutions often converge around one champion solution. It is vital to provide a diverse range of solutions to the decision-makers in MOO. Thus, an algorithm such as Multiple Objective Genetic Algorithm (MOGA) (Fonseca et al., 1993), Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994) and Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas & Deb, 1994) emerged and diversity become an important aspect in multi-objective optimization. In MOGA, Pareto dominance-based ranking is used in fitness assignment and use niching method for fitness sharing. Similarly, NSGA also proposed the usage of non-dominated sorting to assign fitness and use the same niching mechanism. On the contrary, NPGA implements tournament-based ranking in the selection process and niching fitness sharing. This is deemed more efficient as it only uses Pareto ranking on selected individuals.

At the end of the 1990s, a new concept called elitism was introduced. Elitism allows the best-found individuals (i.e., elites) to be carried over to the next generations. This technique is widely implemented in MOEA to enable faster solution convergences. Some notable examples are Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler & Thiele, 1998), Non-dominated Genetic Algorithm 2 (NSGA-II) (Deb et al., 2002) and Pareto Archived Evolution Strategy (PAES) (Corne et al., 2000). These three algorithms use an external population that stores the best-found individuals. The external population, also known as the archive, is another population created separately from the main population to store elite solutions. There is a limit imposed on the size of the external populations. Initially, if the offspring generated after reproduction is better than its parent, it is added to the external population until the maximum size of the population is reached. If a solution dominates the solutions in the external population, the solutions that are being dominated will be removed from the external population. If the external population is full, then a density comparison will be made to decide which solution will stay in the external population. In SPEA, a clustering strategy is employed to manage the external population. The clustering strategy aims to reduce the size of the external population by grouping the solutions in the external population into several clusters. Then only one solution is chosen from each cluster. Similarly, PAES also utilizes elitism but uses a different density comparison that utilizes cells. The search space is divided into several cells. Some cells will contain more solutions than others. As such, they are known as crowded solutions, and it is less favourable to store a crowded cell of solutions in the external archive. On the other hand, NSGA-II utilizes fast non-dominated sorting and crowding functions to eliminate the dominated and closely aligned individuals and store the non-dominated individuals in the external population.

Later, indicator-based MOEA are proposed. The examples are Indicator-Based Evolutionary Algorithm (IBEA) (Zitzler & Künzli, 2004), S-Metric Selection MOEA (SMS-MOEA) (Beume et al., 2007), An Algorithm for Fast Hypervolume-Based Many-Objective Optimization (HypE) (Bader & Zitzler, 2011), Inverse generational Distance+ Multi-objective Evolutionary Algorithm(IGD+ MOEA) (Lopez & Coello, 2016) and Two-stage R2 Indicator based Evolutionary Algorithm (TS-R2EA) (Li et al., 2018). This type of MOEA includes performance evaluation such as hypervolume in HypE algorithm, IGD+ indicator in IGD+MOEA and R2 indicator in TS-R2EA as a utility function during the evolutionary process to assess the fitness of individuals.

Recently, the focus of MOEA is shifting towards many-objective optimization which considers more than three objectives. Examples of the many-objective algorithm are Non-dominated Sorting Genetic Algorithm 3 (NSGA-III) (Deb & Jain, 2014) and clustering-ranking Evolutionary Algorithm (crEA) (Cai et al., 2015). NSGA-III utilizes a set of reference points to guide the search to produce more diversified solutions for many-objective optimization and crEA employs a clustering technique to promote diversity and a ranking technique to promote convergence in solving many-objective optimization problems.

As the number of objectives increases, the diversity and convergence of the solutions become harder to balance. Thus, decomposition-based MOEA is becoming more popular. The core concept of decomposition is to divide a MOP into several subproblems to solve. Besides this, decomposition-based MOEA allows the search to be more uniform to obtain more diverse solutions when solving a MOP. Examples of decomposition-based MOEA are Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang & Li, 2007) and Improved MOEA/D (I-MOEA/D) (Zheng et al., 2018), An improved MOEA/D algorithm with Adaptive Evolution

Strategy (AES-MOEA/D) (Wang et al., 2020) and MOEA/D with Random Partial Update Strategy (MOEA/D-PS) (Lavinas et al., 2020). This category of algorithm is discussed in detail in Section 2.2 as it is the focus of this research.

Table 2.2 summarizes the existing techniques of MOEAs and categorization. The techniques of MOEAs are categorized into five different categories (i.e., elitism, non-elitism, indicator-based, many-objectives, and decomposition-based). Elitism is a mechanic introduced in the late 1990s to preserve elites in a population by allowing them to be carried over to the next generation. Earlier version of MOEAs is categorized as non-elitism. Besides, as MOEA are evaluated using performance indicators, several MOEAs utilized indicators in the search process. Hence, these types of algorithms are categorized as indicator-based. Later, MOEAs are also employed to solve many-objectives problem, the MOEA that build its mechanic to suit the complex searching environment of many-objectives are categorized as many-objectives based. The last category is decomposition-based where the MOEAs divides the multi-objectives problem into several single-objective problems and solves them simultaneously.

Table 2.2 MOEAs Techniques and Categorization

Year	Author	Algorithm	Category
1985	Schaffer	Vector Evaluated Genetic Algorithm (VEGA)	Non-Elitism
1993	Fonseca et al.	Multiple Objective Genetic Algorithm (MOGA)	Non-Elitism
1994	Srinivas & Deb	Non-dominated Sorting Genetic Algorithm (NSGA)	Non-Elitism
1994	Horn et al.	Niched Pareto Genetic Algorithm (NPGA)	Non-Elitism
1998	Zitzler & Thiele	Strength Pareto Evolutionary Algorithm (SPEA)	Elitism
1999	Knowles and Corne	Pareto Archived Evolution Strategy (PAES)	Elitism
2000	Corne et al.	Pareto Envelope-based Selection Algorithm (PESA)	Elitism
2001	Zitzler et al.	Strength Pareto Evolutionary Algorithm 2 (SPEA2)	Elitism
2002	Deb et al.	Non-dominated Genetic Algorithm II (NSGA-II)	Elitism
2004	Zitzler & Künzli	Indicator-Based Evolutionary Algorithm (IBEA)	Indicator-based
2007	Beume et al.	S-Metric Selection MOEA (SMS- MOEA)	Indicator-based
2007	Zhang & Li	Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D)	Decomposition-based
2011	Bader & Zitzler	An Algorithm for Fast Hypervolume-Based Many- Objective Optimization (HypE)	Indicator-based
2014	Deb & Jain	Non-dominated Sorting Genetic Algorithm III (NSGA-III)	Many-objectives
2015	Cai et al.	clustering-ranking Evolutionary Algorithm (crEA)	Many-objectives
2016	Lopez & Coello	Inverse generational Distance+ Multi-objective Evolutionary Algorithm (IGD+ MOEA)	Indicator-based
2018	Li & Cheng	Two-stage R2 Indicator based Evolutionary Algorithm (TS-R2EA)	Indicator-based
2018	Zheng et al.	Improved MOEA/D (I-MOEA/D)	Decomposition-based
2019	Wang et al.	An Improved MOEA/D algorithm with Adaptive Evolution Strategy (AES-MOEA/D)	Decomposition-based
2020	Lavinas et al.	MOEA/D with Random Partial Update Strategy (MOEA/D-PS)	Decomposition-based

2.1.3 Parameter Control in Multi-objective Optimization

Parameter plays an important in ensuring the good performance of a metaheuristic for MOP. There are two different types of parameter settings. The first type of parameter setting is parameter tuning, which sets the parameter before the optimization. The second type of parameter setting is parameter control. In parameter control, the parameters are set during the optimization. Parameter control can be classified into three different categories as shown in Figure 2.2 (Eiben et al., 2007). A list of multi-objective algorithms that utilize these parameter controls is shown in Table 2.3.

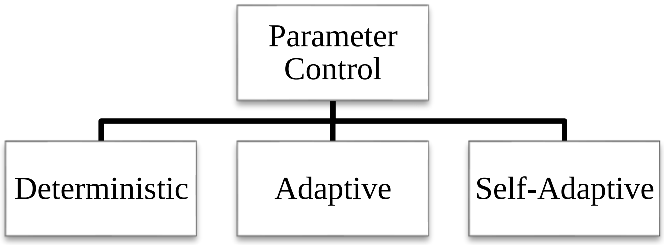

Figure 2.2 Type of Parameter Control

Table 2.3 Parameter Control in MOEAs

Parameter Control	**Author**	**Algorithm**
Deterministic	Tan et al. (2006)	AMO+PAES, PESA, NSGA-II, SPEA2, IMOEA
	Tan et al. (2009)	AVO+NSGA-II, SPEA2
	Fan et al. (2016)	MOEA/D-IEpsilon
	Yang et al. (2017)	MOEA/D-PBI + APS, SPS
Adaptive	Lin et al. (2017)	AMOEA
	Li et al. (2011)	Adap-MODE
Self-Adaptive	Oliver et al. (2017)	MOOEA

Deterministic parameter control is a parameter control technique that is activated at a specified interval. It does not provide any feedback from the search. For instance, a time-varying schedule in the form of a step function can be a form of deterministic parameter control. Tan et al. (2006) designed the adaptive mutation operator (AMO) using deterministic parameter control. In the parameter control, the mutation is initialized with a high value to promote exploration during the early stage of the search process, the mutation is decreased over time to focus on exploitation (Tan et al., 2006). This parameter control is also used in adaptive variation operators (AVO) (Tan et al., 2009). It is also designed to focus on exploration at the beginning and exploitation towards to end of algorithm execution. Besides this, MOEA/D-IEpsilon lowers the parameter level, ε (used in epsilon constraint handling) when the generation counter reaches the control generation (Fan et al., 2016). The parameter, ε is obtained via a four-rule step function to control the search between infeasible and feasible search space. The first rule defines when the value of ε is equal to zero, it will prevent the algorithm from exploring the infeasible search space. The second rule will guide the algorithm to search feasible regions and the third rule explores the infeasible region. The fourth rule exerts the highest guidance towards the feasible region. On the other hand, Yang et al. incorporate the Adaptive Penalty Scheme (APS) to control the setting of a parameter, θ by focusing on convergence at the early search stage and diversity at the late search stage (Yang et al., 2017). The parameter, θ is a crucial factor to balance convergence and diversity whereby a small value of θ will move the search towards convergence while a large value of θ will favour diversity.

The second parameter control technique is adaptive parameter control. It utilizes the feedback from search to serve as the input for a mechanism that determines the change. Li et al. integrate adaptive parameter control which changes the values of Differential Evolution (DE) crossover rate, *CR* and mutation scaling factor, *F* according to the recent success rate (Li et al., 2011). Similarly, Lin et al. proposed an adaptive parameter control on the DE parameters, μ_{Fi} and μ_{CRi} by dynamically updating them