Effective and Efficient Evolutionary Many-Objective Optimization

Yani Xue

for the degree of Doctor of Philosophy

of the

Department of Computer Science
Brunel University London

November 2020

Declaration

I, Yani Xue, hereby declare that this thesis and the work presented in it is entirely my own. Some of the work has been previously published in journal or conference papers, and this has been mentioned in the thesis. Where I have consulted the work of others, this is always clearly stated.

To my parents.

Abstract

Many-objective optimization is core to both artificial intelligence and data analytics as real-world problems commonly involve multiple objectives which are required to be optimized simultaneously. A large number of evolutionary algorithms have been developed to search for a set of Pareto optimal solutions for many-objective optimization problems. It is very rare that a many-objective evolutionary algorithm performs well in terms of both effectiveness and efficiency, two key evaluation criteria. Some algorithms may struggle to guide the solutions towards the Pareto front, e.g., Pareto-based algorithms, while other algorithms may have difficulty in diversifying the solutions evenly over the front on certain problems, e.g., decomposition-based algorithms. Furthermore, some effective algorithms may become very computationally expensive as the number of objectives increases, e.g., indicator-based algorithms.

The aim of this thesis is to investigate how to make evolutionary algorithms perform well in terms of effectiveness and efficiency in many-objective optimization. After conducting a review of key concepts and the state of the art in the evolutionary many-objective optimization, this thesis shows how to improve the effectiveness of conventional Pareto-based algorithms on a challenging real-world problem in software engineering. This thesis then explores how to further enhance the effectiveness of leading many-objective evolutionary algorithms in general by extending the capability of a very popular and widely cited bi-goal evolution method. Last but not least, this thesis investigates how to strike a balance between effectiveness and efficiency of evolutionary algorithms when solving many-objective optimization problems.

The work reported is based on either real-world or recognized synthetic datasets, and the proposed algorithms are compared and evaluated against leading algorithms in the field. The work does not only demonstrate ways of improving the effectiveness and efficiency of many-objective optimization algorithms but also led to promising areas for future research.

Acknowledgement

First of all, I would like to express my deepest gratitude to my principal supervisor Prof. Xiaohui Liu and second supervisor Prof. Martin Shepperd for the persistent support, professional guidance and invaluable encouragement throughout my PhD research. Prof. Xiaohui Liu always motivated me to do better and gave me the freedom to grow and mature as a researcher. I learned a lot from him, both research and about life in general. His encouragement, inspiration, and enthusiasm are invaluable throughout the period of my study. I would also like to take this opportunity to express my sincere thanks to Prof. Martin Shepperd. He is very responsible and strives for excellence. His support and helpful guidance were tremendously helpful in completing this research.

I would like to express my gratitude to Dr. Miqing Li for his expertise, insight and valuable discussions. His rich knowledge in the field of many-objective optimization and insightful comments have deepened my understanding of this field, and his suggestions have made this work better.

I would like to thank the following people for useful discussions, comments, suggestions and support of my research during the PhD stage: Prof. Zidong Wang, Dr. Stasha Lauria, Dr. Stephen Swift, Dr. Weibo Liu, Dr. Lulu Tian, Dr. Bashir Dodo, Yuchen Guo, Prof. Chunyan Han, Wenbin Yue, Nadia Hussain, Khalipha Nuhu, Bhaveet Nagaria, and Nura Tijjani Abubakar.

Many thanks go to all my colleagues and friends from the Centre for Intelligent Data Analysis for the pleasant and enjoyable working atmosphere: Mashael Al-Luhaybi, Leila Yousefi, Leila Ghanbar, Dr. Diana Suleimenova, Zear Ibrahim, Jieyu Zhang, Athina Ioannou, Nurul Md Saleh.

Finally, I would like to extend my sincere thanks to my family, especially my parents, for their unconditional love, continuous support and encouragement.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ADO	aggregation-based dominance
CD	convergence difference
CTC	cross-tree constraint
DRS	dominance resistant solution
DTLZ	Deb, Thiele, Laumanns and Zitzler test suite
EMO	evolutionary multi-objective optimization
HV	hypervolume
IGD	inverted generational distance
MOP	multi-objective optimisation problem
MaOP	many-objective optimization problem
NCV	number of constraint violations
PBI	penalty-based boundary intersection function
PF	Pareto front
PM	polynomial mutation
SBX	simulated binary crossover
SPL	software product line
SPLOT	software product lines online tools
TCH	Tchebycheff

Algorithms

aBiGE	BiGE with angle-based crowding degree estimation
AR-MOEA	indicator-based MOEA with reference point adaptation
BEE	balancing effectiveness and efficiency
BiGE	bi-goal evolution
EA	evolutionary algorithm
IBEA	indicator based evolutionary algorithm
IBEA-SIP	IBEA with ShrInk Prioritize framework
MaOEA	many-objective evolutionary algorithm
MaOEA-CSS	MaOEA using a coordinated selection strategy
MOEA	multi-objective evolutionary algorithm
MOEA/D	multi-objective evolutionary algorithm using decomposition
MOEA/D-TCH	MOEA/D with Tchebycheff function
MOEA/D-TCH-SIP	MOEA/D-TCH with ShrInk Prioritize framework
MOEA/DD	MaOEA using dominance and decomposition
NSGA-II	non-dominated sorting genetic algorithm II
NSGA-II-ADO	NSGA-II with aggregation-based dominance
NSGA-II-SIP	NSGA-II with ShrInk Prioritize framework
NSGA-III	non-dominated sorting genetic algorithm III
NSGA-II/SDR	NSGA-II with strengthened dominance relation
1by1EA	MaOEA using a one-by-one selection strategy
RPD-NSGA-II	NSGA-II using the reference point-based dominance
RVEA	reference vector-guided evolutionary algorithm
SPEA2	strength Pareto evolutionary algorithm 2
SPEA2+SDE	SPEA2 with shift-based density estimation
SPEA2+SDE-ADO	SPEA2+SDE with aggregation-based dominance
SPEA2+SDE-SIP	SPEA2+SDE with ShrInk Prioritize framework
SRA	stochastic ranking algorithm
VaEA	vector angle-based evolutionary algorithm

Symbols

f^{con}	aggregation value
f_a	angle-based crowding degree
f_i	the ith objective value
$\widetilde{f}_i$	the ith normalized objective value
M	number of objectives
n	number of decision variables
N	population size
P	evolutionary population (solution set)
p_c	crossover probability
p_m	mutation probability
Q	selected solutions for the next generation evolution
$\boldsymbol{w}_j$	a weight vector close to the jth objective axis direction
x_i	the ith decision variable
η_c	distribution index in SBX crossover
η_m	distribution index in polynomial mutation
$\prec$	to Pareto dominate
$\prec_{ADO}$	to ADO-dominate

Chapter 1

Introduction

Many-objective optimization problems (MaOPs) refer to the optimization scenarios having more than three objectives to be considered simultaneously. MaOPs abound in real-world applications, such as industrial scheduling [44], water distribution systems [48], software engineering [64, 148, 116], and automotive engine calibration problems [107]. For example, when assessing the performance of a machine learning algorithm, one may need to take into account not only accuracy but also some other criteria such as efficiency, misclassification cost, interpretability, and security.

In an MaOP, since the performance increase in one objective may lead to a decrease in some other objectives, there is often no single best solution that can optimize all the objectives. Thus the problem becomes one of attempting to find a set of trade-off solutions or non-dominated solutions, which is known as the Pareto set. The corresponding mapping of a Pareto set in objective space is called the Pareto front.

Evolutionary algorithms (EAs) are metaheuristic optimization algorithms inspired by natural selection. They have been successfully applied to tackle bi- and tri-objective problems due to the following characteristics: 1) low requirements on the problem properties, 2) population-based property allows them to simultaneously search for a group of solutions (or individuals) in a single run. These algorithms are referred to as multi-objective evolutionary algorithms (MOEAs). The major purpose of MOEAs is to provide a population (a set of optimal individuals or solutions) that balance convergence or proximity (converging a population to the Pareto front) and diversity (diversifying a population over the whole Pareto front).

During the last decade, there is an increasing interest in the use of evolutionary algorithms to many-objective optimization problems, resulting in a variety of many-objective evolutionary algorithms. These algorithms may be classified into five categories, i.e., the algorithms modifying the Pareto dominance relation [84, 61], the algorithms modifying density estimation in the conventional Pareto-based algorithms [95], the decomposition-based algorithms [34], the indicator-based algorithms [5], and the aggregation-based algorithms [96]. They have been successfully applied to many applications; however, some challenges remain to be addressed.

It has been found that some many-objective evolutionary algorithms may have difficulty in converging their solutions to the Pareto front when solving many-objective optimization problems. For example, the indicator-based [162, 9] and decomposition-based [156, 34] algorithms may fail to provide sufficient selection pressure towards the Pareto front even in the 4-D objective space, according to the study in [93]. Furthermore, some recent aggregation-based approaches, such as bi-goal evolution (BiGE) [96], suffer from insufficient selection pressure on a class of MaOPs where the search process involves dominance resistant solutions [150].

Another important issue is the diversity maintenance of the population in many-objective optimization. Some indicator-based algorithms may bias the search towards a certain region of the Pareto front, such as SMS-EMOA [9] and IBEA [162], which generally prefer the knee points [97] and the boundary points [94] on the Pareto front, respectively. In addition, the improvement in convergence might weaken the population's diversity to some extent. For example, the decomposition-based algorithms exhibit good convergence, but they commonly fail to preserve a set of evenly distributed solutions on many-objective optimization problems with irregular Pareto fronts [71]. For another example, the algorithms modifying Pareto dominance [84, 61] could facilitate the convergence of the population in many-objective optimization, but the obtained set of solutions tend to concentrate on one or several small regions of the Pareto front [28, 106]. Furthermore, some state-of-the-art evolutionary algorithms for many-objective optimization problems may fail to diversify their solutions in certain regions of the Pareto front. For example, SPEA2+SDE [95] has difficulty in covering boundary solutions of the Pareto front when tackling certain problems [86].

Furthermore, some algorithms may not be very efficient when solving many-objective optimization problems with numerous objectives (e.g., ten or more objectives). For example, SPEA2+SDE, which has been demonstrated to be effective for tackling many-objective optimization problems [93, 91], becomes very computationally expensive as the number of objectives increases since its computational complexity is $O(MN^3)$ (N is the size of the population). Moreover, the evolutionary algorithms using the hypervolume indicator [165] (e.g., SMS-EMOA [9] and MO-CMA-ES [67]) suffer from an exponential increase in computational time with the increase of the number of objectives.

1.1 Aim and Objectives

Motivated by the above discussions, this thesis aims to explore how to make evolutionary algorithms *effective* and *efficient* in dealing with many-objective optimization problems. In the context of evolutionary multi-/many-objective optimization, *effectiveness* refers to the performance of an algorithm in obtaining a solution set that are both close to the Pareto front and uniformly distributed over the front. *Efficiency* refers to the amount of computational time required for the execution of an algorithm.

There are three objectives to fulfill this aim:

Objective 1: To explore how to improve the effectiveness of conventional Pareto-based algorithms for many-objective optimization problems.

Objective 2: To investigate how to further enhance the effectiveness of state-of-the-art evolutionary algorithms for many-objective optimization problems.

Objective 3: To study how to design an evolutionary algorithm that can achieve both effectiveness and efficiency for many-objective optimization problems.

First, this thesis addresses the issue of effectiveness in Pareto-based algorithms for MaOPs by focusing on a real-world problem in software engineering, dubbed the optimal feature selection problem for software product lines. Recently, various multiobjective evolutionary algorithms have been employed to find valid software product configurations. However, it has been found that these algorithms have difficulty in balancing the correctness and diversity of obtained solutions within a reasonable time. To enable Pareto-based algorithms to work well on many-objective optimization problems, enhancing the convergence of these algorithms plays a critical role.

Second, this thesis explores how to enhance the effectiveness of advanced evolutionary algorithms for many-objective optimization in order to tackle more generic cases. Recently, the bi-goal evolution (BiGE) approach, which provides a general framework to map a solution set from the original high-dimensional objective space into a bi-goal (objective) space of proximity and crowding degree, has become very popular and received considerable attention in the community. However, BiGE has an important limitation, namely, it may not be able to fully address those many-objective optimization problems with a high chance of producing dominance resistant solutions. This thesis shows how the effectiveness of BiGE may be enhanced by improving its crowding degree estimation.

Third, this thesis investigates how to make evolutionary algorithms for many-objective optimization both effective and efficient. It is very rare that a many-objective evolutionary algorithm performs well in terms of both effectiveness and efficiency on a variety of many-objective optimization problems. For example, it is difficult for the aggregation-based algorithms to strike a balance between convergence and diversity. This also applies to the algorithms modifying Pareto dominance or diversity maintenance. The decomposition-based algorithms often struggle to maintain diversity on problems with irregular Pareto front shapes. Some indicator-based algorithms may become very computationally expensive with the increase of the number of objectives. Furthermore, many algorithms need some extra parameters, particularly for the algorithms modifying the Pareto dominance relation. As such, it is of practical significance to develop a new evolutionary algorithm that can balance effectiveness and efficiency for various many-objective optimization problems.

This thesis explores several innovative approaches to address the above challenges in evolutionary many-objective optimization, including a novel dominance relation for conventional Pareto-based algorithms in searching for a set of high-quality solutions of a real-world many-objective optimization problem (Chapter 3), an angle-based crowding degree estimation method for the bi-goal evolution framework to eliminate the adverse effects caused by dominance resistant solutions in the search process (Chapter 4), and an effective and efficient algorithm that can handle various many-objective optimization problems (Chapter 5).

1.2 Contributions

The main contributions of the thesis are listed as follows.

Firstly, this thesis introduces a novel dominance relation, termed aggregation-based dominance (ADO) in order to make conventional Pareto-based algorithms effective for many-objective optimization problems. ADO focuses on the convergence enhancement of Pareto-based algorithms in order to obtain a population that could strike a balance between diversity and convergence in the high-dimensional objective space. Specifically, since the Pareto dominance criterion is not sufficient to distinguish between individuals, ADO is served as a selection criterion to further distinguish between non-dominated individuals during the evolutionary process. ADO is integrated into two popular Pareto-based algorithms (i.e., NSGA-II and SPEA2+SDE) and has been shown to be effective in accelerating the convergence process, as demonstrated in the configuration of software product lines. Moreover, a comprehensive comparison with four different types of evolutionary algorithms based on the recently proposed ShrInk Prioritize (SIP) method [64] show that ADO is very competitive in finding high-quality solutions for nine tested software product lines with up to seven objectives.

Secondly, this thesis proposes an angle-based crowding degree estimation method for a well-established many-objective evolutionary algorithm BiGE to solve the many-objective optimization problems where the search process involves dominance resistant solutions. In comparison with the crowding degree estimation based on the distance between individuals, the crowding degree estimation based on vector angles between individuals has the advantage of distinguishing dominance resistant solutions from other solutions in a population. In aBiGE, the distance-based crowding degree estimation in BiGE is replaced with the proposed angle-based crowding degree estimation in order to weaken the adverse impact of dominance resistant solutions on the algorithm. Given the preference of solutions with both good proximity and diversity in the bi-goal space, dominance resistant solutions (with both poor proximity and crowding degree) would be removed during the evolutionary process. The performance of the proposed aBiGE algorithm is evaluated by the comparative study of aBiGE and BiGE on a series of benchmark test problems with dominance resistant solutions. The experimental results show the effectiveness of this replacement.

Thirdly, this thesis proposes a many-objective evolutionary algorithm (called BEE) to achieve both high effectiveness and efficiency in many-objective optimization. BEE focuses on the design of environmental selection to preserve promising solutions for the next-generation evolution. It performs the key decision process of environmental selection by selecting boundary solutions as well as non-boundary solutions. The former is to determine the range of the estimated Pareto front and the latter is to select solutions which achieve a good balance between diversity and convergence. In particular, the diversity estimation is based on the solutions that have already been selected for the next-generation evolution, and has a bias towards the solution that is far away from the selected solutions (thus good diversity), whereas the convergence estimation is concerned with the comparison between the unselected solutions. Furthermore, BEE has no additional parameter except those associated with an evolutionary algorithm (e.g., population size and crossover rate). Systematic experiments are carried out to compare BEE with 11 state-of-the-art algorithms on 60 problem instances with up to 15 objectives. The experimental results show that the BEE algorithm significantly outperforms all the compared algorithms on the majority of test problems with various Pareto front shapes.

1.3 Thesis Structure

This thesis is organized as follows.

Chapter 2 provides the necessary background knowledge for the thesis. First, this chapter presents some concepts and terminologies in multiobjective optimization, followed by the description of key evolutionary algorithms and performance indicators. Second, this chapter introduces many-objective optimization and discusses some general issues on many-objective optimization. Third, this chapter reviews a variety of recent advances in the use of evolutionary algorithms in many-objective optimization. The main categories of these algorithms cover modification of Pareto dominance relation, modification of density estimation, decomposition-based algorithms, indicator-based algorithms, grid-based algorithms, and aggregation-based algorithms.

Chapter 3 begins with some background on software product line. Then, this chapter introduce a novel aggregation-based dominance (ADO) for conventional Pareto-based algorithms to make them suitable for many-objective optimization problems, followed by its use in three evolutionary operations, including fitness assignment, mating selection, and environmental selection. Next, the performance of ADO is assessed by investigating 1) whether ADO could improve the performance of two popular Pareto-based algorithms, i.e., NSGA-II and SPEA2+SDE, and 2) how these two Pareto-based algorithms, when integrated with ADO, perform in comparison with other state-of-the-art algorithms in searching valid product configurations for software product lines. Finally, the influence of the parameter setting in ADO is investigated.

In Chapter 4, an angle-based crowding degree estimation for the Bi-Goal Evolution (BiGE) framework is proposed to address the issues caused by dominance resistant solutions in many-objective optimization. First of all, this chapter illustrates the basic idea for this work, and provides a comparison and analysis of density estimation based on Euclidean distance and vector angles in many-objective optimization problems with dominance resistant solutions. Then, this chapter introduces the angle-based crowding degree estimation method and its incorporation into the BiGE framework. Next, the experiments are conducted on two types of many-objective optimization problems with and without dominance resistant solutions during the search process, respectively.

In Chapter 5, an effective and efficient algorithm (called BEE) for various many-objective optimization problems is introduced. Initially, this chapter provides a comprehensive analysis of the issues of the balance between effectiveness and efficiency in existing MaOEAs. In addition, this chapter describes the framework and detailed information of the BEE algorithm. BEE focuses on the design of environmental selection with two operations: selecting boundary solutions and selecting non-boundary solutions, followed by the analysis of time complexity of BEE and the differences between BEE and SPEA2+SDE since both algorithms use the shift-based comparison between solutions in the selection procedure. Furthermore, the performance of BEE is evaluated by the comprehensive comparative study of BEE and 11 state-of-the-art MaOEAs on a benchmark test suite with a variety of Pareto front shapes.

In Chapter 6, the work presented in this thesis is summarized and several directions of future research are presented.

1.4 Publications

The work resulting from this thesis has been reported in the following papers:

- **Y. Xue**, M. Li, M. Shepperd, S. Lauria, and X. Liu. A novel aggregation-based dominance for Pareto-based evolutionary algorithms to configure software product lines. *Neurocomputing*, 364:32–48, 2019. (Resulting from Chapter 3)

- **Y. Xue**, M. Li, and X. Liu. Angle-based crowding degree estimation for many-objective optimization. Advances in Intelligent Data Analysis XVIII, Lecture Notes in Computer Science 12080, M Berthold et al (Eds), pp. 574–586, 2020. (Resulting from Chapter 4)

- **Y. Xue**, M. Li, and X. Liu. Balancing effectiveness and efficiency in evolutionary many-objective optimization (submitted to *IEEE Transactions on Evolutionary Computation*). (Resulting from Chapter 5)

Chapter 2

Background

In this chapter, a review of evolutionary multi- and many-objective optimization is presented. First, this chapter provides basic concepts and terminologies in multiobjective optimization, followed by the description of key evolutionary algorithms. Second, this chapter introduces many-objective optimization and reviews state-of-the-art evolutionary algorithms.

2.1 Multiobjective Optimization

When tackling optimization problems in the real world, two or more performance criteria are usually involved in order to determine how “good” a certain solution is. These criteria are termed as objectives (e.g., cost, safety, efficiency, etc.) that are often in conflict with each other. In most optimization problems there are several restrictions imposed on these objectives according to particular features of available resources, such as time restrictions, physical limitations, and etc. This type of problem is called the multiobjective optimization problem (MOP), which could be observed from a wide range of areas, such as in economics, engineering, and medical field. To model such problems, an MOP can be mathematically defined as follows [31]:

$$\begin{aligned}
\text{Minimize/Maxmize} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \\
\text{subject to} \quad & g_j(\mathbf{x}) \leq 0, \qquad j = 1, 2, ..., J \\
& h_k(\mathbf{x}) = 0, \qquad k = 1, 2, ..., K \\
& L_i \leq x_i \leq U_i, \quad i = 1, 2, ..., n
\end{aligned} \tag{2.1}$$

where $\mathbf{x}$ denotes an n-dimensional decision variable vector from the feasible region in the decision space Ω: $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\mathbf{x} \in \Omega$. The last set of constraints are referred to as variable bounds. These bounds restrict the value of each decision variable x_i within a range of L_i to U_i, and constitute the decision space Ω. $F(\mathbf{x})$ represents an M-dimensional objective vector ($M \geq 2$), $f_i(\mathbf{x})$ is the ith objective function to be minimized or maximized, objective functions $f_1, f_2, \ldots, f_M$ constitute M-dimensional space called the objective space, $g_j(\mathbf{x}) \leq 0$ and $h_k(\mathbf{x}) = 0$ define J inequality and K equality constraints, respectively.

The concept of "optimal" in multiobjective optimization is different from that in single-objective optimization, where the solution is usually a single solution of global maximum or minimum depending on whether the problem is a maximization or minimization problem in specific. In multiobjective optimization, since these objectives often conflict with each other, there is no single optimal solution for an MOP, but rather a set of Pareto-optimal solutions, which are defined on the basis of the Pareto dominance relation. Each solution is a trade-off among the multiple objectives. Namely, the increase in at any one objective leads to a decrease in other objectives simultaneously.

Definition 2.1.1 (Pareto Dominance). *Given two decision vectors* $\mathbf{x}, \mathbf{y} \in \Omega$ *of a minimization MOP,* $\mathbf{x}$ *is said to (Pareto) dominate* $\mathbf{y}$ *(denoted as* $\mathbf{x} \prec \mathbf{y}$ *), or equivalently* $\mathbf{y}$ *is dominated by* $\mathbf{x}$*, if and only if [25]*

$$\forall\, i \in (1, 2, \ldots, M) :\ f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \ \wedge\ \exists\, i \in (1, 2, \ldots, M) :\ f_i(\mathbf{x}) < f_i(\mathbf{y}). \tag{2.2}$$

Namely, given two solutions, one solution is said to dominate the other solution if *it is at least as good as the other solution in any objective and is strictly better in at least one objective.*

Definition 2.1.2 (Pareto Optimality)**.** *For a given MOP, a solution* $\mathbf{x}^* \in \Omega$ *is said to be Pareto optimal if and only if there is no solution* $\mathbf{z} \in \Omega$ *dominates it. All such solutions are called Pareto-optimal (or nondominated) solutions.*

Definition 2.1.3 (Pareto Set)**.** *For a given MOP, all Pareto-optimal (or nondominated) solutions in the decision space constitute the Pareto set (PS).*

Definition 2.1.4 (Pareto Front)**.** *For a given MOP, the Pareto front (PF) is referred to corresponding objective vectors to a Pareto set.*

Definition 2.1.5 (Dominance Resistant Solution)**.** *Given a solution set, dominance resistant solution (DRS) is referred to the solution with an extremely poor value in at least one objective, but with near-optimal value in some other objectives.*

Because of the black-box nature of many objective functions and the complexity of the search space, it is almost impossible to use optimization algorithms or metaheuristic methods to calculate the true Pareto set for an MOP. Instead, algorithms aim to produce an approximation of the Pareto front.

2.2 Evolutionary Multiobjective Optimization

2.2.1 Introduction

Evolutionary algorithms (EAs) are a class of optimization methods that simulate the natural selection principle *survival of the fittest* from the biological world. Over many years, EAs have become very popular for solving MOPs. They are known as *evolutionary multiobjective optimization* (EMO) algorithms or *multi-objective evolutionary algorithms* (MOEAs). Their population-based property allows them to simultaneously search for a number of possible solutions in one run, while traditional optimization methods (e.g. mathematical programming methods and simulated annealing) have to perform a series of separate runs. Furthermore, EAs are less susceptible to the problem characteristics (e.g., the continuity, the Pareto front shape, multimodality, and nonlinear constraint), thus, they are able to handle large and highly complex search space. By contrast, it is very difficult to address these issues using traditional optimization techniques.

A general framework of an MOEA is illustrated in Figure 2.1. First, a population P is initialized by randomly generating N individuals. Second, in the mating selection, individuals that have better quality tend to become parents of the next generation to push for quality improvement. By doing this, the combination of these parents is more likely to generate good offspring. Third, variation operators, i.e., crossover and mutation, are employed to these parents to produce offspring. Fourth, environmental selection determines the survival of solutions (i.e., next-generation population) from the current population and offspring. This evolutionary process continues until a terminating condition (e.g., the number of generations exceeds a predefined upper bound) is reached.

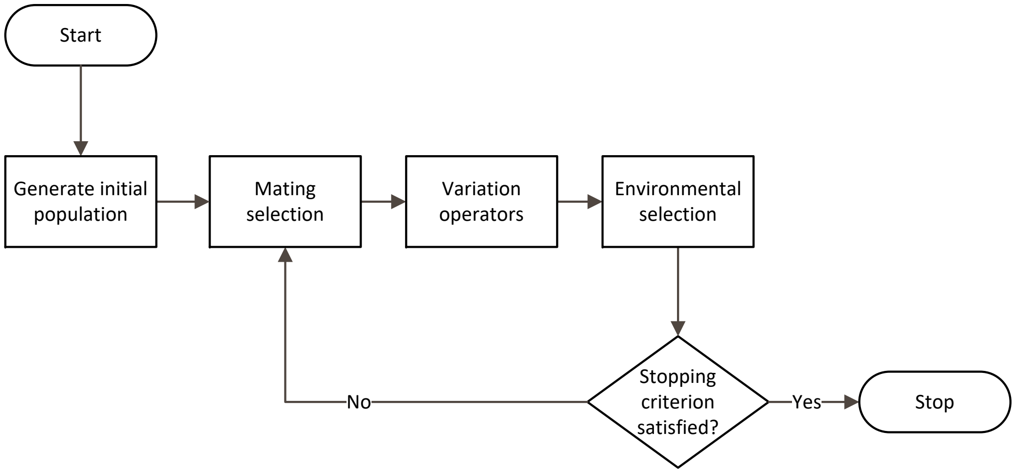


Figure 2.1: A general MOEA framework.

Fitness Assignment

In general, an MOEA requires both the objective function and the fitness function. The objective function defines the optimality condition related to the characteristics of the problem, while the fitness function (in the algorithm domain) measures the degree to which a particular solution fulfills the condition and assigns the corresponding value to the solution. In multiobjective optimization, the concept of *Pareto optimality* is often used to design the fitness function for the optimization of several objectives in parallel.

To approximate the Pareto front, the design of fitness functions in MOEAs usually covers the information regarding both convergence and density (or crowding degree) of each individual in a population. For convergence estimation, since the approximation of the Pareto front usually consists of nondominated solutions, various ranking methods related to the nondominated solutions have been proposed for the optimization of several objectives in parallel [47, 65, 165, 37]. These methods usually sort the individuals in the objective search space before the selection procedure. Each individual is assigned a rank, which is calculated based on the Pareto dominance relation between any two individuals in the population (i.e., an individual dominates, is dominated by or nondominated by other individuals).

Moreover, density information of individuals is also an important component in the fitness function. Considering the Pareto dominance relation lacks the ability to reflect the density information of solutions in a population, various approaches for density estimation have been proposed, such as clustering [165], k-nearest distance [163], crowding distance [37], and niching techniques (e.g., fitness sharing) [47, 126]. Usually, individuals are selected in a convergence-first-and-diversity-second scheme. It has been shown that this scheme works well on MOPs with two or three conflicting objectives [37, 163].

Recently, a large number of studies [156, 92, 75, 162, 42, 9, 67, 53, 16] have developed other criteria, such as indicator-based criterion and aggregation-based criterion, in the fitness assignment of a population. These criteria commonly convert an objective vector into one scalar value, based on which the individuals in a population can be totally ordered. These criteria typically lead to better performance in terms of convergence when compared to the Pareto dominance criterion [9].

Diversity Maintenance

In MOPs, density estimation plays a key role in maintaining the diversity of obtained solution set over the whole Pareto front [11]. The density becomes the main criterion to guide the search when individuals could not be distinguished by using Pareto dominance-based criterion (i.e., these individuals are nondominated with each other) during the evolutionary process. An individual with fewer other individuals located in its neighborhood would be assigned a lower density and has a higher possibility of being selected for the next generation.

Over the last few decades, a variety of methods have been proposed to estimate the density of individuals in a population during the evolutionary process. In a pioneering study, Goldberg [52] suggested the idea of the incorporation of fitness sharing into an evolutionary algorithm to maintain the diversity of population along the Pareto front. Fitness sharing is a widely used niching approach, which promotes the search in sparsely populated regions by degrading the fitness values of those individuals in densely populated regions. Specifically, individuals in densely populated regions are first identified by their niche counts (the number of neighbours of an individual within a predefined niche size σ_{share}). Then, the fitness of these solutions is penalized according to their niche counts. Some representative MOEAs that adopt fitness sharing as the density estimation method include the Multiobjective Genetic Algorithm (MOGA) [47], the Nondominated Sorting Genetic Algorithm (NSGA) [126], and the Niched-Pareto Genetic Algorithm (NPGA) [65].

Furthermore, other density estimation methods have been proposed in the literature. For example, the Strength Pareto EA (SPEA) adopts a *clustering* method to estimate individuals' density [165]. The Strength Pareto Evolutionary Algorithm 2 (SPEA2) utilizes the *kth nearest neighbor* method [163]. The Nondominated Sorting Genetic Algorithm II (NSGA-II) employs the *crowding distance* method, which estimates the density of an individual in the population by calculating the average distance of its two closest neighbours located on either side of this individual on each objective [37]. In grid-based MOEAs, the objective space is commonly divided into a number of hyperboxes. Each individual is located in a specific hyperbox with a certain grid coordinate and its density is defined as the number of individuals in the hyperbox where it is located. The individual located in a less populated hyperbox has a higher possibility to survive in the next generation, and thus the diversity of the population could be maintained. Some representative grid-based MOEAs include the Pareto Envelope-based Selection (PESA) [29], the Pareto-Archived Evolution Strategy (PAES) [80], the Pareto Envelope-based Selection using Region-based Selection (PESA-II) [27], and the Dynamic Multiobjective EA (DMOEA) [153].

2.2.2 Multiobjective Evolutionary Algorithms Classification

Traditional multiobjective evolutionary algorithms have two common goals: one is to converge the population to the Pareto front and the other is to maintain the diversity of population over the Pareto front. Over the past decades, there has been increasing interest in the development of various MOEAs. Generally, MOEAs can be categorized into three types on the basis of their selection mechanisms, including Pareto-based algorithms, decomposition-based algorithms, and indicator-based algorithms [140].

Pareto-based Algorithms

The majority of existing MOEAs belong to the category of Pareto-based algorithms, which are based on the Pareto dominance relation among individuals in a population. Pareto-based algorithms distinguish between individuals based on two criteria. The primary selection criterion is based on the Pareto dominance relation and those nondominated solutions are preferred. Furthermore, since only using Pareto dominance could degrade the diversity of the population, several techniques such as crowding degree and fitness sharing have often been used as the secondary criterion in MOEAs. The Nondominated Sorting Genetic Algorithm II (NSGA-II) [37], the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [163], and the Pareto Envelope-based Selection Algorithm II (PESA-II) [27] are such representative algorithms, which have been widely implemented in various application fields [128, 143, 158]. Among them, NSGA-II may be one of the most popular Pareto-based algorithms, where a fast nondominated sorting method was proposed to rank solutions during the selection process. Despite the fact that Pareto-based algorithms are popular in evolutionary multiobjective optimization, these algorithms have important limitations of inferior performance on MOPs with a high dimensional objective space [140, 73, 96] or complex Pareto sets [87].

Decomposition-based Algorithms

Another category of MOEAs is the decomposition-based algorithms, which decompose an MOP into a set of subproblems (single-objective optimization subproblems [156] or simple multiobjective subproblems [103]) and optimize these subproblems simultaneously. The most representative of this category is the Multiobjective Evolutionary Algorithm using Decomposition (MOEA/D) [156].

The decomposition-based algorithms have several advantages over Pareto-based algorithms. First, aggregation approaches in decomposition-based algorithms are capable of improving the computational efficiency for the fitness evaluation. Second, the decomposition-based algorithms have high search ability for a variety of problems, such as combinatorial optimization problems [17], problems with complicated Pareto sets [87], and problems with a large number of objectives [69].

On the other hand, the decomposition-based algorithms have difficulty in maintaining the diversity of solutions for those problems with *irregular* Pareto front shapes, such as inverted, degenerate, and disconnected Pareto fronts [71]. One reason is that the shape of the weight distribution is not consistent with the shape of the Pareto front, and thus the uniformly distributed weight vectors could not guide the search for a set of evenly distributed solutions over the Pareto front.

In the past decade, numerous MOEA/D variants have been proposed [103, 115, 92, 75, 108, 56, 100] to enhance the performance of MOEA/D when tackling multiobjective optimization problems. Furthermore, the decomposition-based algorithms have been successfully employed in a variety of real-world optimization problems [19, 54, 136].

Indicator-based Algorithms

Another category of MOEAs is characterized by using performance indicators (e.g., the *inverted generation distance* (IGD) [26] and *hypervolume* (HV) [165]) to guide the search of a population towards the Pareto front. These algorithms are known as indicator-based algorithms.

The Indicator-Based Evolutionary Algorithm (IBEA), proposed in [162], provides a general framework that could incorporate arbitrary performance indicators into evolutionary algorithms. Recently, the HV indicator has been widely used for selection mechanisms in MOEAs due to its good theoretical and empirical properties [166, 13]. Some representative HV-based algorithms include the S Metric Selection-based Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [9] and the Multiobjective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [67].

The main shortcoming of HV-based algorithms is that the computational complexity of HV increases exponentially with the increase of the number of objectives [145]. To overcome this drawback, many studies have proposed different approaches to reduce the computational cost of HV with respect to both the exact computation [10, 12, 144, 78] and the approximate estimation [74, 5, 14]. Furthermore, the R2 indicator has been employed as an alternative for the HV because of its lower computational cost than that of HV [15].

2.2.3 Performance Indicators

Multi-objective evolutionary algorithms aim to obtain a set of nondominated solutions that well represent the Pareto front of an MOP and their performance is commonly evaluated by performance indicators. The last decades have witnessed the development of a variety of performance indicators regarding different quality aspects of the solutions obtained by an MOEA, including (1) the closeness to the Pareto front (i.e., convergence), (2) the coverage over the Pareto front (i.e., spread or extensity), and (3) the uniformity amongst solutions (i.e., uniformity). In general, the spread and uniformity are collectively called the diversity of the solution set. Table 2.1 lists some representative performance indicators and the quality aspect(s) evaluated by the corresponding indicators.

As can be seen from Table 2.1, some performance indicators only measure one particular quality aspect of the solution set. Some can reflect both spread and uniformity of the solution set, and others cover all quality aspects (i.e., convergence, spread, and uniformity) of the solution set. Specifically, the Inverted Generational Distance (IGD) [26] and the hypervolume (HV) [165] are two widely used performance indicators, which assess the quality of the solution set in terms of both convergence and diversity (i.e., spread and uniformity).

As one of the most commonly used performance indicators in the area, IGD can well reflect both convergence and diversity given the problem's Pareto front is available. The IGD indicator is calculated as below. Let P^* denote a set of uniformly distributed reference points on the Pareto front and P denote the final solution set obtained by an algorithm. The IGD is the average distance from points in set P^* to their nearest solution in set P. Mathematically,

$$\mathrm{IGD} = \frac{1}{|P^*|} \sum_{z \in P^*} d(z, P) \tag{2.3}$$

where $d(z, P)$ denotes the minimum Euclidean distance between a reference point z and its nearest solution in P, and $|P^*|$ represents the size of P^*. A smaller value of IGD indicates better quality of set P for approximating the Pareto front.

Recently, a modified version of IGD, called IGD$^+$ was proposed [70] to address the main weakness of IGD, i.e., non-compliance with the Pareto dominance relation. In IGD, it is possible that a solution set P has a worse IGD value than another solution set Q even when P are superior to Q regarding the Pareto dominance. In contrast, in IGD$^+$, the distance between a reference point and its nearest solution is refined by considering their Pareto dominance relationship, which ensures dominated solutions contribute nothing to indicator values and is more accurate in the assessment.

The HV indicator (also called *size of dominated space* [165], *hyperarea metric* [137], *S metric* [159], or *Lebesgue measure* [46]) is another frequently used performance indicator in the field, which could reflect a combined performance of convergence and diversity. Compared with IGD, the calculation of HV does not require a reference set which represent the Pareto front, and thus it is more suitable for real-world optimization problems, whose Pareto fronts are usually unknown. Given a solution set A, and

Table 2.1: Performance indicators and quality aspect(s) of a solution set evaluated by the corresponding indicators

No.	Performance Indicators	Convergence	Spread	Uniformity
1	C [164]	√		
2	Generational Distance (GD) [138]	√		
3	Coverage [164, 165]	√		
4	Purity [6]	√		
5	GD$_p$ [123]	√		
6	Convergence Measure [35]	√		
7	Generational Distance (GD)[139]	√		
8	Coverage error ϵ [119]		√	
9	Overall Pareto Spread [146]		√	
10	Maximum Spread (MS) [160, 51, 2]		√	
11	Spread assessment [102]		√	
12	PD [142]		√	
13	Cluster [146]			√
14	Spacing (SP) [122]			√
15	Mininmal spacing [6]			√
16	Entropy Measure [45]			√
17	Uniform Distribution [129]			√
18	Δ Metric [37]		√	√
19	M-DI [3]		√	√
20	DCI [94]		√	√
21	Diversity Measure [35]		√	√
22	Sigma Diversity Metric [111]		√	√
23	Coverage over Pareto front (CPF)		√	√
24	HV$_d$ [77]		√	√
25	$R1$, $R2$, $R3$ [59]	√	√	√
26	Hypervolume (HV) [165]	√	√	√
27	Hyperarea Ratio [137]	√	√	√
28	Coverage Difference [159]	√	√	√
29	Degree of approximation (DOA) [40]	√	√	√
30	Hyperarea difference [146]	√	√	√
31	IGD-NS [135]	√	√	√
32	ϵ-indicator [166]	√	√	√
33	ϵ performance [81]	√	√	√
34	IGD [26]	√	√	√
35	IGD$_p$ [123]	√	√	√
36	G-Metric [105]	√	√	√
37	IGD$^+$ [70]	√	√	√
38	I_{SDE} [86, 95]	√	√	√
39	Averaged Hausdorff Distance Δ_p [123]	√	√	√
40	Performance comparison indicator (PCI) [97]	√	√	√

a reference point $R = (r_1, r_2, \ldots, r_M)^T$, the HV indicator measures the volume of the objective space that is dominated by solutions in A and bounded by R [165]:

$$HV(A) = \lambda(\bigcup_{\mathbf{x} \in A} [f_1(\mathbf{x}), r_1] \times [f_2(\mathbf{x}), r_2] \times \cdots \times [f_M(\mathbf{x}), r_M]) \tag{2.4}$$

where λ represents the Lebesgue measure [46], M denotes the number of objectives, $F = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))$ is an objective vector. The larger value of HV, the better convergence and diversity of the algorithm. Notice that reference point R should be dominated by all nondominated solutions.

2.3 Many-Objective Optimization

2.3.1 Introduction

Many-objective optimization problems (MaOPs) refer to the optimization scenarios having more than three objectives to be considered simultaneously. MaOPs abound in real-world applications, such as software engineering [64, 148, 116], industrial scheduling [44], water distribution systems [48], and automotive engine calibration problems [107]. In the last decade, there is an increasing interest in the use of evolutionary algorithms to MaOPs, resulting in a variety of many-objective evolutionary algorithms (MaOEAs).

2.3.2 Key Challenges in Many-Objective Optimization

MaOPs with higher dimensional objective spaces pose many challenges to the conventional MOEAs, which can be summarized as follows.

First, the Pareto dominance relation between individuals is not effective to facilitate the convergence of the population in many-objective optimization. In Pareto-based algorithms, individuals are compared through two criteria: Pareto dominance relation and density. In a high-dimensional objective space, the majority of individuals in a population become nondominated (i.e., equally good solutions) at an early stage of the search [73, 61]. Since the Pareto dominance-based primary selection criterion fails to distinguish between individuals, the second criterion - density - becomes the main criterion to guide the search, leading to a substantial reduction of the selection pressure towards the Pareto front and the slowdown of the evolutionary process. This is termed the *active diversity promotion* (ADP) phenomenon in [114].

Some studies [73, 140] have observed that the ADP phenomenon could lead to the failure of Pareto-based algorithms in finding a good Pareto front approximation, because of the preference of dominance resistant solutions [68]. These are solutions with an extremely poor value in at least one objective, but with near-optimal value in some other objectives. They have worse performance in terms of convergence, but they are treated as nondominated solutions. As a result of the ADP phenomenon in Pareto-based algorithms, the final set of solutions could be widely spread in the objective space but far away from the true Pareto front.

Second, density estimation can be very challenging in many-objective optimization. Althought various density estimation methods have been proposed in classical MOEAs, they may provide inaccurate information regarding the density of population members when solving many-objective optimization problems. For example, a density estimation method, called *crowding distance* [37], may not be able to accurately measure the density of individuals in a population when the number of objectives is larger than two [82]. The main reason is that it does not treat an individual as a whole, but takes into account the two closest neighbours of an individual on each objective. Furthermore, the density estimation methods in the grid-based MOEAs [80, 29, 27, 153], which are based on the number of individuals in each hyperbox, may fail to provide an accurate density estimation of the individuals in a population for MaOPs. It is mainly because there is an exponential increase in the number of hyperboxes as the number of objectives goes up. In this case, individuals have a high possibility to spread over different hyperboxes, leading to the inaccuracy of these density estimation methods.

Third, the effectiveness of the recombination operator in MOEAs is questionable in many-objective optimization. The main reason is that the objective space becomes extremely large with an increase in the number of objectives. Consequently, it is highly possible that many individuals are far away from each other in high-dimensional objective space, considering the population commonly consists of a small number of individuals. Recent studies [34, 72, 118] have suggested that two distant parents may produce offsprings that are also far away from their parents. Therefore, it may be necessary to develop some special recombination operators (such as mating restriction), which allow the offsprings to be reasonably close to their parents, in order to effectively handling MaOPs.

Fourth, in higher-dimensional space, the computational cost of using some well-known indicators to evaluate the performance of an MOEA is very high. For instance, the computation of exact hypervolume increases exponentially with the number of objectives increases [145].

Fifth, the visualization of a solution set in high-dimensional objective space is a challenging task. It plays a vital role in helping decision makers to evaluate and select certain solutions based on their preferences. The scatter plot, which is an intuitive and effective visualization tool in low-dimensional objective spaces, is no longer an option when the number of objectives exceeds three.

2.4 Evolutionary Aglorithms for Many-Objective Optimization

Over the last decade, there has been a growing interest in developing many-objective evolutionary algorithms (MaOEAs) for various MaOPs. Roughly, these MaOEAs can be classified into six categories.

2.4.1 Modification of Pareto Dominance Relation

The first category is concerned with modifying the conventional Pareto dominance relation for many-objective optimization. For example, a modified Pareto dominance concept for many-objective optimization, called ϵ-dominance was developed in [84]. By relaxing the area that an individual dominates, ϵ-dominance could provide sufficient selection pressure towards the Pareto front.

A *preference order ranking* method was proposed in [39] to replace the traditional nondominated ranking method in Pareto-based algorithms to enhance the selection pressure towards the Pareto front for MaOPs.

A *grid dominance* relation was presented in [152], which compares pairs of solutions based on their coordinate location in the high-dimensional objective space. The relaxation degree is adjusted adaptively by the grid size according to the population during the evolutionary process.

A new dominance relation, termed *fuzzy Pareto dominance* was proposed in [61], which uses the concept of *fuzzy logic* to alleviate the lack of effectiveness of Pareto dominance for many-objective optimization. Specifically, a *fuzzy Pareto dominance*-based fitness evaluation method is employed to continuously differentiate between solutions by assigning each individual a rank value during the search process.

Recently, a *strengthened dominance relation* (SDR) was introduced [134] to achieve a balance between the convergence and diversity of the nondominated solutions for MaOPs. In SDR, a niching method is applied, which only allows one solution to survive in each niche in order to maintain the diversity of a solution set. The degree of the dominance area of a solution is determined by the niche size (parameter θ), which is an acute angle value adaptively evaluated according to the population during the evolutionary process.

Overall, compared with the Pareto dominance relation, these dominance relations allow a solution to be easily dominated by other solutions in the high-dimensional space, thus enhancing the selection pressure towards the Pareto front. However, they have important limitations. First, most of them struggle to balance convergence and diversity, and as a result, the obtained solution set may converge into one or several subareas of the Pareto front [28, 106]. Second, these modified Pareto dominance relations often involve additional parameters and a good performance on a particular problem needs a proper setting of these parameters, especially when the number of objectives is large [85, 98].

2.4.2 Modification of Density Estimation

The second category is concerned with modifying density estimation of the conventional Pareto-based algorithms since maintaining diverse nondominated solutions may harm the convergence of the population evolving towards the Pareto front in the high-dimensional space [114, 73].

For example, a modified NSGA-II, proposed in [140], assigns the crowding distance value of boundary solution a zero value instead of an infinity value in NSGA-II, and therefore, boundary solutions could be rejected during the evolutionary process. Experimental results indicated that the proposed algorithm had better performance than the conventional NSGA-II in terms of the average distance (within $100,000$ function evaluations).

For another example, diversity management mechanism DM1 [2] controls the activation or deactivation of diversity requirements according to the distribution of the population, hoping to achieve a proper balance between convergence and diversity. The experimental results indicated that NSGA-II using DM1 performed consistently better than the original NSGA-II on a set of many-objective optimization problems with 6–20 objectives.

In addition, SPEA2+SDE [95] proposed a Shift-based Density Estimation (SDE) strategy to enable poorly converged solutions to be penalized by high density values. These poorly converged solutions are highly possible to be eliminated during the evolutionary process. The experimental results showed that the SDE strategy could largely improve the performance of three Pareto-based algorithms (i.e., NSGA-II, SPEA2, and PESA-II) on many-objective optimization problems. Especially, the version of SPEA2 with SDE (i.e., SPEA2+SDE) outperformed several leading MaOEAs in terms of balancing the convergence and the diversity for many-objective optimization problems.

However, SPEA2+SDE may fail to well maintain boundary solutions in some problems such as DTLZ1 [38] having a triangular Pareto front [86]. The true Pareto front of the five-objective DTLZ1 and the final solution set of SPEA2+SDE in one typical run on the five-objective DTLZ1 are shown by parallel coordinates in Figure 2.2. From the figure, it can be seen that the solution set obtained by SPEA2+SDE fails to cover the boundaries of the Pareto front, with the range of the solution set being between 0 and around 0.4, whereas the Pareto front of five-objective DTLZ1 in the range between 0 and 0.5 on each objective. One possible reason is that SDE overemphasizes convergence during the evolutionary process.

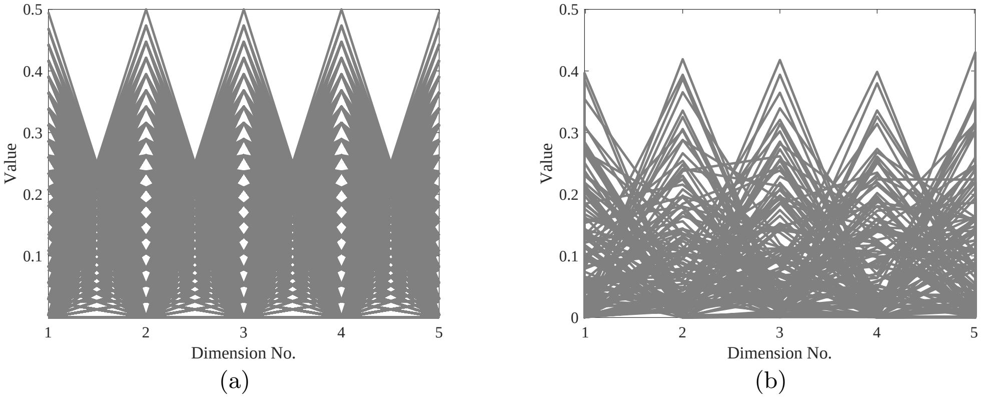


Figure 2.2: The true Pareto front of the five-objective DTLZ1 and the final solution set obtained by SPEA2+SDE on the five-objective DTLZ1 in one typical run, shown by parallel coordinates. (a) the true Pareto front. (b) The final solution set of SPEA2+SDE.

2.4.3 Decomposition-based Algorithms

The third category refers to decomposition-based algorithms. Decomposition is a traditional method for handling multiobjective optimization problems. Since the appearance of the algorithm MOEA/D [156], decomposition-based algorithms have enjoyed their popularity in the area.

MOEA/D, which was originally proposed for solving bi- and tri-objective optimization problems, have been demonstrated to be highly competitive in tackling many-objective optimization problems [34, 69]. In MOEA/D, an original many-objective optimization problem is decomposed into a set of scalarizing optimization subproblems by using a group of evenly distributed weight vectors and an aggregation approach. Some representative aggregation approaches include the penalty-based boundary intersection (PBI) approach, the Tchebycheff (TCH) approach, and the weighted sum (WS) approach, etc. To obtain a population that could well approximate the entire Pareto front, these predefined weight vectors are utilized to maintain the diversity of the population, and an aggregation approach is employed to optimize each subproblem, thus facilitating convergence of the population.

In the past few years, there has been an increasing amount of literature on developing effective decomposition-based algorithms for many-objective optimization problems. For example, NSGA-III [34] integrates the idea of decomposition into the environmental selection procedure of the framework of NSGA-II. In NSGA-III, a set of structure reference points are generated by using Das and Dennis's normal-boundary intersection approach [30]. Notice that each solution is linked to a reference point according to its perpendicular distance towards the reference line. As in the original NSGA-II, the nondominated sorting procedure is employed to divide the combination of the current population and offspring into a set of nondominated fronts F_0, F_1, $\ldots$, F_K. The new population P_{t+1} is constituted by selecting each front in ascending order, starting from F_0, until the size of selected solutions reaches the population size or the critical front (denoted F_L) is found. To determine which solutions in F_L should be placed into P_{t+1}, a niche-preservation operation is utilized to select solutions one by one according to the niche counts (i.e., the number of solutions associated with each reference point).

The Many-Objective Evolutionary Algorithm using Dominance and Decomposition (MOEA/DD) [89] combines the advantages of the two well-known algorithms, i.e., MOEA/D and NSGA-II, for tackling many-objective optimization problems, and it has been shown to be a promising steady-state MaOEA. In MOEA/DD, a set of weight vectors are generated by employing Das and Dennis's [30] normal-boundary intersection approach, and a two-layer weight vector generation approach is utilized to reduce the total number of generated weight vectors in the high-dimension objective space. Each weight vector is associated with a solution, and also corresponds to a subregion that is used for local density estimation purposes. In the mating selection procedure of MOEA/DD, parent solutions are chosen from the neighbourhood (i.e., the T closest subregions) of the current weight vector for generating offspring solutions. In the case that no solutions are associated with the neighbouring subregions, parent solutions are randomly selected from the entire population with a small probability $(1-\delta)$ to enhance the exploration ability. The key feature of MOEA/DD is its update procedure. Each time, only one offspring solution is added to the population, and the nondomination level structure of the population is updated according to the method proposed in [88]. Then, the population is updated in a steady-state hierarchical manner according to the Pareto dominance, local density estimation, and an aggregation function (i.e., the PBI function [156]), sequentially.

The θ-Dominance based Evolutionary Algorithm (θ-DEA) [155] was proposed to facilitate the convergence performance of NSGA-III for many-objective optimization problems. The θ-DEA [155] inherits the strength of NSGA-III in diversity maintenance based on a set of uniformly distributed reference points. In θ-DEA, each solution in the population is allocated to a cluster (represented by a reference point) according to a clustering operator. To balance convergence and diversity, the key concept θ-dominance is employed, which uses the PBI aggregation function [156] in the normalized objective space and is only applied to those solutions within the same cluster. In the environmental selection, a nondominated sorting scheme based on θ-dominance is used to partition the combination of current population and offspring into different θ-nondomination fronts, and a new population is constructed as in NSGA-III, but the solutions in the last accepted front are randomly selected.

The Reference Vector-guided Evolutionary Algorithm (RVEA) [20] adopts a scalarization approach, called Angle-Penalty Distance (APD), as the selection criterion for the elitism selection strategy. RVEA divides the objective space into a set of small subspaces by associating each solution with its closest reference vector (i.e., the reference vector that has the minimal angle to the objective vector of the solution). In each subspace, an elitism selection strategy based on APD is applied. In APD, the convergence of the solutions is estimated by calculating the distance between the solutions and the ideal point, while the diversity of the solutions is measured by the minimal angle between the solutions and the reference vectors. The elitist solution in each subpopulation is defined as the one having the lowest APD value. Additionally, a reference vector adaptive strategy is applied to maintain the population diversity in the objective space to handle the objective functions that are not well normalized.

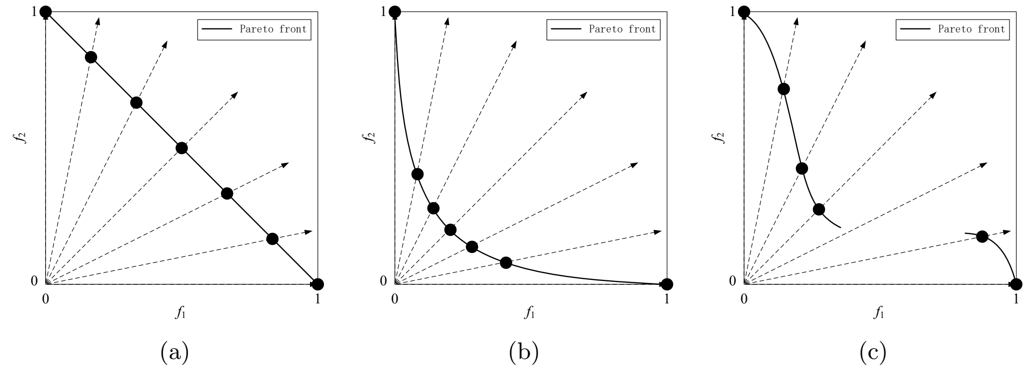
The Vector Angle-based Evolutionary Algorithm (VaEA) was proposed in [147], where the maximum-vector-angle-first and the worse-elimination principles are adopted for the environmental selection. In the selection process, M (number of objectives) extreme solutions and M best-converged solutions (calculated by summing up each normalized objective value) are first selected for the new population P. Then, the maximum-vector-angle-first principle is employed to maintain the wideness and uniformity of the population by adding one solution with the maximum vector angle to the solutions in P each time. Moreover, to facilitate the convergence of the population to the Pareto front, the worse-elimination principle is adopted to replace solutions with poor convergence by other solutions within its niche.

A Strength Pareto Evolutionary Algorithm using Reference Direction (SPEA/R) [76] revives a traditional Pareto-based algorithm, called SPEA, to tackle many-objective optimization problems. In SPEA/R, a set of predefined reference directions are used to divide the objective space into a number of subregions aiming to preserve population diversity. Each solution is associated with a subregion defined by reference directions, thereby solutions are distributed to different subregions. The new fitness assignment scheme is employed on solutions located in each subregion. In the new fitness assignment scheme, both local and global convergence of a solution are estimated according to the raw fitness calculation method in SPEA2. In particular, the local density of a solution is estimated by the proposed reference direction-based density estimator rather than the time-consuming one in SPEA2. In the environmental selection, SPEA/R adopts a diversity-first-and-convergence-second selection strategy, which guides the solutions towards predefined search directions (a set of well-distributed reference directions), thus promoting the selection pressure for many-objective optimization.

More recently, the NSGA-II using the Reference Point-based Dominance (RPD-NSGA-II) was proposed in [41] to deal with many-objective optimization problems. In RPD-NSGA-II, a decomposition-based dominance, called Reference Point-based Dominance (RP-dominance), is adopted to distinguish nondominated solutions for many-objective optimization problems by using a set of uniformly distributed reference points. Moreover, a new diversity criterion based on the PBI approach [156] is employed to evaluate the performance of the solutions in terms of convergence and diversity based on a set of well-distributed reference points. The main difference between NSGA-II and

RPD-NSGA-II is that the latter employs RP-dominance instead of Pareto dominance to perform a non-RPD-dominated sorting. Furthermore, in both algorithms, the new population is formed by the best fronts, however, the truncation operators for the last accepted front are different. In RPD-NSGA-II, the last considered front is truncated by preferring the solutions having the minimum perpendicular distance to the associated reference vectors in the objective space.

Overall, decomposition-based algorithms perform very well in terms of convergence and are able to well maintain a set of diverse solutions for many-objective optimization problems with *regular* or simplex-like Pareto fronts (e.g., a hyperplane or concave hypersphere), when compared with traditional Pareto-based algorithms. A simple example is illustrated in Figure 2.3 (a), where the intersection point between each weight vector and the Pareto front would be an optimal solution to a subproblem.



(a) (b) (c)

Figure 2.3: The distribution of Pareto optimal solutions (solid dots) to be found by using systematically generated weight vectors (dashed arrows) depends on the Pareto front shapes (solid lines). (a) Simplex-like Pareto front (b) Strongly convex Pareto front (c) Disconnected Pareto front.

On the other hand, decomposition-based algorithms typically face challenges of diversifying their solutions over the Pareto front for problems with *irregular* Pareto front shapes, which are disconnected, strongly concave or convex, degenerate, inverted simplex-like, and scaled, etc. [71]. The main reason is that the distribution of weight vectors is not consistent with Pareto front shapes [71]. Two simple examples are given in Figure 2.3 (b) and (c), which illustrate that well-distributed weight vectors may fail to generate well-distributed solutions on problems with highly convex Pareto front and disconnected Pareto front, respectively. In Figure 2.3 (b), most optimal solutions are located in the central area of the Pareto front on a highly convex Pareto front. In Figure 2.3 (c), some weight vectors have no intersections with the Pareto front, resulting in poor performance of decomposition-based algorithms in terms of diversity.

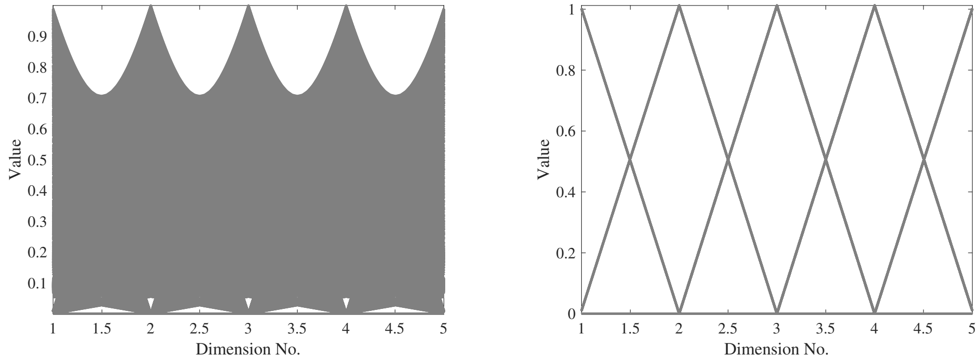
To address the above difficulty, a number of decomposition-based algorithms [4, 18, 149, 56, 100] have developed various adaptive adjustment of weight vectors methods to make them more compliant with different Pareto front shapes.

2.4.4 Indicator-based Algorithms

The fourth category of MaOEAs refers to those based on indicators. Such algorithms adopt performance indicators as selection criteria to guide the search of a population towards the Pareto front. One representative algorithm of this category is the indicator-based evolutionary algorithm (IBEA) [162], which provides a general framework to incorporate arbitrary performance indicators into evolutionary algorithms. A performance indicator could formulate the preference information of decision makers and allow the comparison between pairs of individuals in the population. In [162], dominance preserving indicators, such as the binary additive ϵ-indicator $I_{\epsilon+}$ [166] and I_{HD}-indicator based on the hypervolume concept [165], are suggested. Because of the effectiveness of $I_{\epsilon+}$, IBEA was shown to perform well in terms of convergence for many-objective optimization problems, but it may fail to maintain the solutions' diversity for many-objective optimization problems [141].

Figure 2.4 plots the final solution set of IBEA on the five-objective DTLZ3 problem in one typical run by parallel coordinates. It should be mentioned that the parallel coordinates is a popular approach for visualizing the many-objective solution set using a 2D parallel coordinates plane. Recently, Li et al. [101] provided a systematical explanation of how to read a solution set of a many-objective optimization problem in parallel coordinates. The authors indicated that the parallel coordinates can be used as an assistant tool when evaluating a many-objective solution set, regarding its quality in convergence, coverage, and uniformity.

As shown in Figure 2.4, the solution set obtained by IBEA only converges the



(a)

(b)

Figure 2.4: The true Pareto front and the final solution set obtained by IBEA on the five-objective DTLZ3 in one typical run, shown by parallel coordinates. (a) the true Pareto front. (b) the final solution set of IBEA.

boundary areas of Pareto front, as there are no solutions located inside the range $(0, 1)$ on all objectives.

Because of the good theoretical and empirical properties of HV [165] shown in [166, 13], it has been commonly used in MaOEAs (e.g., SMS-EMOA [9] and MO-CMA-ES [67]) to guide the search for a population that well approximates the Pareto front by maximizing the corresponding HV indicator [42]. However, the main weakness of HV-based search algorithms is that the runtime of the HV increases exponentially with the increase of the number of objectives [145], which leads to a decreasing efficiency for many-objective optimization problems. To deal with the aforementioned issue, an HV estimation algorithm – HypE – was put forward in [5], which adopts the Monte Carlo sampling to approximate the HV value and compares nondominated solutions according to their HV-based fitness. Experimental results showed that Monte Carlo sampling could provide a trade-off between HV computation time and accuracy, which made HV-based search algorithms competitive for many-objective optimization [5, 14].

Recently, a large number of MaOEAs have adopted the R2 indicator as an alternative for the HV indicator because of its weaker monotonicity and lower computational cost than HV [15, 113, 53]. For example, Phan et al. [113] proposed an extension of IBEA with a binary R2 indicator (i.e., R2-IBEA). In R2-IBEA, a hypervolume-based weight vector generation approach is employed to maximize HV, while an adaptive reference point adjustment approach is applied to reduce the bias of the R2 indicator that preferred the central area of the Pareto front [16]. Hernández Gómez and Coello Coello introduced an improved version of MOMBI (i.e., MOMBI-II) [63], which adopts the achievement scalarizing function instead of the weighted Tchebycheff function to enhance the uniform distribution. In MOMBI-II, statistical information regarding the proximity of the population is used to update the reference points. Experimental results showed that MOMBI-II could obtain a solution set with a more uniform distribution than its original version and R2-IBEA for most tested optimization problems with up to ten objectives.

In addition, some algorithms are based on the IGD-NS indicator [135]. For example, the AR-MOEA [132] adopts the IGD-NS indicator with a reference point adaption strategy to address many-objective optimization problems that have different types of Pareto front in both low- and high-dimensional search spaces. The IGD-NS indicator provides a comprehensive measurement of a nondominated solution set and is used as the selection criterion in both mating selection and environmental selection operations. In AR-MOEA, the adaptation of reference points considers both the shape of the approximate Pareto front constituted by those nondominated solutions in the current population and the uniform distribution of reference points. Those reference points are employed to calculate the IGD-NS values of those nondominated solutions in the environmental selection, and allow the IGD-NS indicator to guide the population towards the whole Pareto front for a variety of many-objective optimization problems. AR-MOEA was demonstrated to be promising on various many-objective optimization problems with both regular and irregular Pareto front shapes. However, AR-MOEA is very time-consuming since it adopts a greedy diversity maintenance strategy that iteratively deletes the solution with the worst diversity degree and updates the diversity degrees of all the remaining solutions in the environmental selection.

More recently, some algorithms use the IGD indicator to guide the search in many-objective optimization. For example, Sun et al. [127] proposed a many-objective evolutionary algorithms using IGD, termed MaOEA/IGD, to deal with many-objective optimization problems. In MaOEA/IGD, the decomposition-based nadir point estimation (DNPE) method is utilized to construct the Utopian Pareto front, where a set of uniformly distributed reference points are generated for IGD calculation. Moreover, MaOEA/IGD adopts a computationally efficient dominance comparison strategy, which classifies the population into three fronts by comparing the dominance relation of the solutions to the reference points, followed by three types of proximity distance assignments according to the rank values of individuals, aiming to distinguish individuals with the same front rank values. Then, the solutions are selected based on the assigned rank values and proximity distances of individuals, along with the linear assignment principle in order to improve the convergence and the diversity concurrently. The experiments were conducted on DTLZ and WFG benchmark test suites with 8, 15, and 20 objectives. The experimental results demonstrated that MaOEA/IGD was competitive in addressing many-objective optimization problems compared to five popular many-objective evolutionary algorithms (NSGA-III, MOEA/D, HypE, RVEA, and KnEA). However, one weakness of MaOEA/IGD is that the extreme point estimation method (i.e., DNPE) is very time-consuming.

Furthermore, other algorithms adopt multiple performance indicators as the selection criteria. For example, a hybrid algorithm, called Two_Arch2 [141], takes advantage of the $I_{\epsilon+}$ indicator to enhance convergence on many-objective optimization problems and a L_p-norm-based diversity maintenance to promote diversity. For another example, the multi-indicator-based algorithm for many-objective optimization problems, called SRA [86], adopts a stochastic ranking technique to balance the search biases of the indicators $I_{\epsilon+}$ and I_{SDE} [95]. Both Two_Arch2 and SRA were shown to be competitive on many-objective optimization problems, but they may not be able to preserve boundary solutions of the Pareto front for some many-objective optimization problems [141, 86]. Furthermore, SRA involves extra parameters for balancing the search biases of different indicators and its performance is influenced by the setting of those parameters.

2.4.5 Grid-based Algorithms

The fifth category of MaOEAs is based on dividing the objective space into grids, which are known as grid-based algorithms. The advantage of using the grid is that it can reflect the information of both convergence and diversity, simultaneously. Specifically, in grid-based algorithms, each solution is located in a grid and these solutions in a population are compared by using their grid coordinates. The convergence and diversity of a population can be achieved by grid-based dominance criteria and grid locations, respectively.

A steady-state multiobjective evolutionary algorithm based on the ϵ-dominance concept, termed ϵ-MOEA [36], is a representative of the grid-based algorithms. In ϵ-MOEA, two populations, i.e., an EA population (P) and an archive population (A) are evolved collaboratively. The EA population is randomly initialized while the archive population stores ϵ-nondominated solutions of P. Next, one solution from each population P and A is selected as parents to generate an offspring (i.e., a new individual). The offspring is then used to update the populations P and A, where the comparisons between individuals are based on the Pareto dominance and ϵ-dominance, respectively. It should be noted that the objective space is split into hyperboxes with size ϵ and each hyperbox in the archive is restricted to contain at most one solution, thus allowing the diversity to be maintained in a population. Nevertheless, the computation costs and information storage of each hyperbox increase exponentially with the number of objectives. Moreover, it is highly possible that solutions in a population are far away from each other in the high-dimensional objective space, as a result, the diversity estimation method that only considers the number of solutions in a hyperbox may lose its effectiveness to distinguish between solutions.

Another example is the Grid-based Evolutionary Algorithm (GrEA) [152], which develops effective methods to use a grid in order to make evolutionary algorithms suitable for many-objective optimization problems. In GrEA, the objective space is divided into div^M hyperboxes based on the objective values of individuals in a population, where div denotes the number of grid divisions set by the users beforehand and M is the number of objectives. Compared to ϵ-MOEA where the grid size is a fixed parameter, in GrEA, the grid size is adapted during the evolutionary process, which could better reflect the location of individuals. In GrEA, the performance regarding both convergence and diversity of an individual is evaluated based on grids. Specifically, three grid-based criteria, including grid dominance, grid ranking, and grid coordinate point distance, are employed for convergence estimation in a grid environment. Another criterion, i.e., the grid crowding distance is used for diversity estimation of individuals in the population, which considers the number of its neighbours and the distance difference between itself and these neighbours. In addition, GrEA adopts a fitness adjustment strategy in environmental selection aiming to avoid partial overcrowding and explore different directions in the archive.

Overall, grid-based algorithms have been shown to be effective for many-objective optimization problems under the proper setting of girds, such as hyperboxes size ϵ in ϵ-MOEA and the number of grid divisions *div* in GrEA [140, 98, 58, 152].

2.4.6 Aggregation-based Algorithms

The last category considers aggregation-based algorithms. They aggregate the objectives of solutions into one or multiple criteria to make them comparable more easily. Some algorithms in this category develop novel selection methods to estimate solutions' performance regarding convergence and diversity. For example, the MaOEA using a One-by-One Selection Strategy (called 1by1EA) [104] adopts a convergence indicator and a distribution indicator in the environmental selection to balance convergence and diversity. The environmental selection strategy could break down into two operations. First, the solution with best convergence value is selected to promote the convergence performance in many-objective optimization. Second, solutions in the neighbourhood of the selected solution are de-emphasized by utilizing a niche technique according to the distribution indicator, with the aim of guaranteeing the diversity of the population. Furthermore, a boundary maintenance mechanism is adopted to enhance the spread of a solution set over the Pareto front.

For another example, the MaOEA using a Coordinated Selection Strategy (MaOEA-CSS) [60] focuses on the coordination and complementary design of mating selection and environmental selection. The new mating selection criterion takes into account two issues: 1) the quality of both parents regarding convergence and diversity, 2) the effective combination of selected parents in an extremely large objective space. In the mating selection, the convergence degree of each solution is estimated by a modified version of achievement scalarizing function (ASF), while the diversity of each solution is estimated by an angle value. In the new environmental selection, each time two closest solutions are identified by angle, the one with less contribution to the convergence and diversity performance of the whole population is iteratively eliminated. This design could alleviate the conflicting degree between convergence and diversity in many-objective optimization.

In addition, an innovative computational approach for designing MaOEAs called bi-goal evolution (BiGE) [96] has attracted significant attention in the field. BiGE focuses on addressing two challenges of many-objective optimization problems. First, increasing the number of objectives could aggravate the conflict between proximity and diversity [114, 2]. Second, the Pareto dominance criterion, which works well on bi- and tri-objective spaces, has difficulty in promoting convergence on higher dimensional objective spaces. BiGE overcomes the two aforementioned challenges by transforming a many-objective optimization problem to a bi-objective one (with respect to convergence and diversity) and using the widely employed Pareto dominance criterion in the low-dimensional objective space. In BiGE, the convergence of a solution is estimated by summing up each normalized objective value. The diversity of a solution is evaluated by a sharing function-based crowding degree and a solution that has better performance in terms of convergence than its neighbours in a niche is assigned a better diversity value controlled by a weight parameter (called the sharing discriminator).

Other algorithms integrate the preference information into the fitness assignment. For example, the Knee Point-driven Evolutionary Algorithm (KnEA) [157], which has a bias towards knee points among nondominated solutions during the selection process in order to promote the convergence performance for many-objective optimization problems. In KnEA, knee points are adopted as another selection criterion when the Pareto dominance criterion fails to distinguish solutions during the evolutionary process. In addition, a niche method based on hyperboxes is adopted to identify the knee points in a small neighborhood, thus the diversity of the population is maintained. It is worth mentioning that KnEA is computationally efficient since there is no explicit diversity measure, such as the crowding distance [37].

Overall, aggregation-based algorithms can effectively balance the convergence and diversity for many-objective optimization problems, but some of them require extra efforts for the proper setting of specific parameters, such as the parameter T for controlling the ratio of knee points to the nondominated solutions in KnEA, the parameter k for balancing the accuracy and the computational cost in the density estimation of 1by1EA, and the threshold value t of determining the difference of two closest solutions' Euclidean distance in the environmental selection of MaOEA-CSS.

2.5 Summary

Chapter 2 has presented the necessary background knowledge for the thesis. First of all, some concepts and terminologies in multiobjective optimization have been provided. Then, many-objective optimization has been introduced, and some general issues of applying evolutionary algorithms to many-objective optimization have been discussed. Finally, a review of state-of-the-art evolutionary algorithms for many-objective optimization has been provided. In the next chapter, the effectiveness of popular Pareto-based algorithms in handling a practical many-objective optimization problem is addressed.

Chapter 3

Many-Objective Optimization for Software Product Line Configuration

This chapter addresses the issue of effectiveness in Pareto-based algorithms for many-objective optimization problems. In particular, a challenging real-world problem in software engineering is used as an example of how these algorithms may be made more effective. In software engineering, optimal feature selection for software product lines (SPLs) is an important and complicated task, involving simultaneous optimization of multiple competing objectives in a large but highly constrained search space. A feature model is the standard representation of features of all possible products as well as the relationships among them for an SPL. Recently, various multi-objective evolutionary algorithms have been used to search for valid products (combinations of features that satisfies all of the constraints in the feature model). However, the issue of the balance between correctness and diversity of solutions obtained in reasonable time has been found very challenging to address in these algorithms.

To tackle this problem, a novel dominance relation for Pareto-based algorithms is proposed to enhance the search for high-quality solutions on the optimal feature selection problem. This chapter is organized as follows. Section 3.1 is devoted to the introduction of this work. Section 3.2 details the ADO approach, and then Section 3.3 shows how ADO is integrated in two popular Pareto-based algorithms NSGA-II and SPEA2+SDE. The experimental results are detailed in Section 3.4. Section 3.5 summarizes this chapter.

3.1 Introduction

With the development of mobile and service-based applications, companies need to reconfigure their applications in order to retain and extend their market share. To meet the demand of different customers, companies often develop and maintain many variations of software products [7]. Recently, there is an increasing trend to adopt software product lines (SPLs) to reduce development costs, shorten development cycles, and improve software reusability and flexibility [55]. An SPL is a class of similar software products, all of which share some core functionalities. Each product configuration is different with different features selected that aim to satisfy the specific requirements of a particular market segment [24].

A feature model [7] is a tree structure that provides representations of an SPL for configuring all possible software products. The key task of an SPL is to select a set of desired features from its feature model in order to fulfill multiple functional requirements (e.g., minimize the product cost, maximize users' preferences) and satisfy the constraints related to various features. In practice, real-world SPLs often contain hundreds or even thousands of features and complex constraints. For example, the Linus X86 kernel feature model from LVAT (Linux Variability Analysis Tools) [1] repository contains $6,888$ features and $343,944$ constraints. It is extremely difficult to manually select optimal features for valid products in such a large and constrained search space. This is called the *optimal feature selection* problem [57].

In the past decade, there have been many studies that adopt different *multi-objective evolutionary algorithms* (MOEAs) as automatic configuration approaches to solve the optimal feature selection problem, which can be seen as a *many-objective optimization problem* (MaOP) [120, 62, 130, 64, 148, 121, 112, 151].

A representative MOEA is the indicator-based evolutionary algorithm (IBEA) [162], which adopts the hypervolume indicator to guide the search towards optimal solutions. Some recent studies [121, 120, 130] demonstrated that IBEA outperforms some popular MOEAs, such as the Nondominated Sorting Genetic Algorithm II (NSGA-II) [37] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [163] in searching for valid solutions on both small and large feature models from the online feature model repositories SPLOT [110] and LVAT, respectively.

Nevertheless, pure IBEA has been found to be insufficient to achieve 100% valid solutions on large feature models. To address this issue, some studies focused on the enhancements of this algorithm. Sayyad et al. [120] presented a "seeding" technique for IBEA to find more valid solutions on some large feature models. Tan et al. [130] proposed a feedback-directed IBEA, which adopted the number of constraint violations as feedback for two evolutionary operators to improve the ability of the algorithm to search for more valid solutions. Henard et al. [62] introduced SATIBEA, which integrates two smart evolutionary operators into IBEA. In addition, SATIBEA could identify and fix an invalid solution (i.e., a software product configuration with constraint violations) by using the SAT solver.

Furthermore, other studies investigate the use of other MOEAs for the optimal feature selection problems. For example, Hierons et al. [64] presented a *ShrInk Prioritize* (SIP) method for different types of MOEA. In particular, the novel encoding method is used to identify prunable features, and the $(1+n)$ approach is used to first optimize the number of constraint violations and then other objectives at the same time to guide the search to feasible space. For another example, Xiang et al. [148] proposed SATVaEA, which combines the recently proposed VaEA [147] algorithm and SAT solvers to repair an invalid product configuration and promote diversity of the obtained solutions.

Despite these advances, it is not an easy task for these algorithms to balance correctness and diversity of solutions in a reasonable time. In addition, it is noted that there are few studies focusing on developing new many-objective evolutionary algorithms to deal with the optimal feature selection problem in configuring SPLs. Given the above, the work focuses on two research questions:

1. Are traditional MOEAs really worse than state-of-the-art algorithms in solving optimal feature selection problem for SPLs?
2. Are there methods that could improve the quality of solutions obtained by using traditional MOEAs, such as NSGA-II without using SAT solvers?

In this chapter, investigations are carried out along these lines and an aggregation-based dominance (ADO) for Pareto-based algorithms is proposed to direct the search for high-quality solutions. ADO is in part inspired by the effectiveness of aggregation functions to drive the population to different parts of the Pareto front through weighted vectors [156, 20]. ADO can be incorporated into three evolutionary operations: fitness assignment, mating selection, and environmental selection in different Pareto-based algorithms to search for valid optimal solutions for an SPL.

3.1.1 Optimal Feature Selection Problem in SPLs

In software engineering, a feature model is a standard representation of features for all possible products of an SPL and the relationships between them [79]. A feature model describes a valid product as a combination of features that satisfies all the constraints [131]. A feature model is represented as a tree-like structure composed of a set of nodes representing features and connections between them. For example, Figure 3.1 illustrates a simplified feature model for mobile phone SPL.

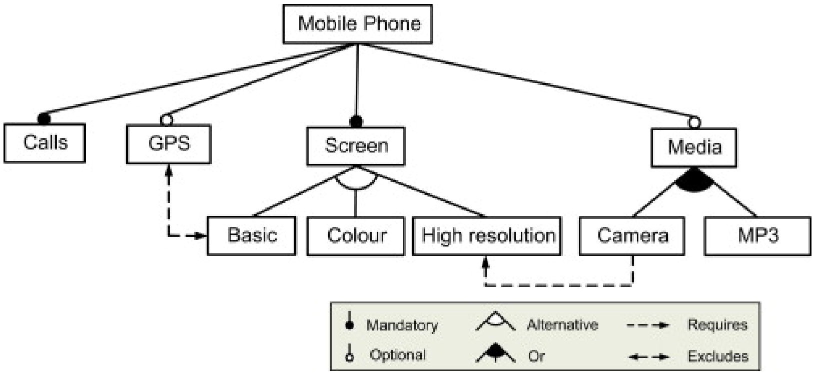


Figure 3.1: A simple feature model for mobile phone SPL (adapted from [8])

Relationships between a parent feature and its child features (or subfeatures) include:

- **Mandatory**. A mandatory feature must be included if its parent feature is included in a product, such as the “call” feature in the example.
- **Optional**. An optional feature can be optionally included in a product, in Figure 3.1, the “Media” feature can be optionally included in products that contain its parent feature (“Mobile Phone” feature).
- **Alternative**. If the parent feature is included in a product, exactly one feature should be selected among a group of sub-features. For instance, a mobile phone must provide support for either a “Basic”, or a “Color”, or a “High resolution” screen in the same product.
- **Or**. If the parent feature is included in a product, one or more of the child features should be selected. In Figure 3.1, a mobile phone can provide support for a “Camera”, an “MP3”, or both of them when their parent feature “Media” is included in the product.

Apart from the above parental relationships between features, feature models also adopt cross-tree constraints (CTCs) to represent the mutual relationship for features. Typically, there are two types of CTCs:

- **Requires**. This relationship allows some features to co-occur, namely, If feature F_a requires feature F_b, the inclusion of F_a implies the inclusion of F_b in this product. In Figure 3.1, the mobile phone with “Camera” feature requires the “High resolution” feature.
- **Excludes**. This relationship indicates that some features cannot exist simultaneously in the same product, namely if a feature F_a excludes a feature F_b, the inclusion of F_a implies the exclusion of feature F_b in this product, and vice versa. In Figure 3.1, the mobile phone with “GPS” feature excludes the “Basic” feature.

3.2 The Proposed Approach

3.2.1 Normalization

A normalization method in [76] is adopted for a many-objective optimization problem where objective values may be disparately scaled in different SPLs. This is to enhance the robustness of the algorithm when the scales of the objective values are different [156, 34].

Algorithm 3.1 shows the pseudocode for the objective normalization procedure. In normalization, every solution in a population is normalized according to the ideal point and nadir point of each objective. Formally, in a minimization MaOP and a population P with N individuals and M objectives, the ideal point $z^{min} = (z_1^{min}, z_2^{min}, \dots, z_M^{min})$ is determined by searching the minimum value of each objective for all individuals in P, which is denoted as $z_j^{min} = \min_{i=1}^{N} f_j(x_i)$, where $x_i \in P$ $(i = 1, 2, \dots, N)$. Similarly, the nadir point $z^{max} = (z_1^{max}, z_2^{max}, \dots, z_M^{max})$ is calculated by $z_j^{max} = \max_{i=1}^{N} f_j(x_i)$, where z_j^{max} is the maximum of $f_j(x_i)$, $x_i \in P$ $(i = 1, 2, \dots, N)$. For each solution $\mathbf{x} \in P$, its objective vector $f(\mathbf{x})$ is normalized to $\widetilde{f}(\mathbf{x}) = (\widetilde{f}_1(\mathbf{x}), \widetilde{f}_2(\mathbf{x}), \dots, \widetilde{f}_M(\mathbf{x}),$ which is calculated by the following equation:

$$\widetilde{f}_j(\mathbf{x}) = \frac{f_j(\mathbf{x}) - z_j^{min}}{z_j^{max} - z_j^{min}} \tag{3.1}$$

Algorithm 3.1 $Normalization(P)$

```
Require: $P$ (current population), $N$ (population size), $M$ (number of objective functions)
for $j \leftarrow 1$ to $M$ do
    $z_j^{min} \leftarrow \min_{\mathbf{x} \in P} f_j(x)$   /* Find the ideal point */
    $z_j^{max} \leftarrow \max_{\mathbf{x} \in P} f_j(x)$   /* Find the nadir point */
end for
for $i \leftarrow 1$ to $N$ do
    $f^{con}(x_i) \leftarrow 0$
    for $j \leftarrow 1$ to $M$ do
        $\widetilde{f}_j(x_i) \leftarrow (f_j(x_i) - z_j^{min})/(z_j^{max} - z_j^{min})$
        $f^{con}(x_i) \leftarrow f^{con}(x_i) + \widetilde{f}_j(x_i)$
    end for
end for
```

During the normalization procedure, the aggregation value of each individual $\mathbf{x}$, denoted as $f^{con}(\mathbf{x})$ is also calculated (line 9 in Algorithm 3.1), which is described in the next section.

3.2.2 Aggregation Function

Aggregation functions are basic techniques in decomposition-based algorithms and are often employed to decompose a many-objective optimization problem into a number of sub-problems and optimize these problems in a collaborative way. However, aggregation functions are utilized in a different manner in this chapter in order to enhance the selection pressure in traditional Pareto-based evolutionary algorithms. Here, an aggregation function is adopted to estimate the convergence performance of an individual, which is based on aggregating the individual's information (by taking all normalized objective values into consideration).

In this chapter, the aggregation function in Bi-Goal Evolution (BiGE) [96] (a meta-objective optimization approach for many-objective optimization problems) is used to estimate the convergence performance for each individual $\mathbf{x}$ in a population P, denoted as $f^{con}(\mathbf{x})$. This aggregation value is calculated by adding up all normalized objective values of the individual in the range [0, 1]:

$$f^{con}(\mathbf{x}) = \sum_{j=1}^{M} \widetilde{f}_j(\mathbf{x}) \tag{3.2}$$

where $\widetilde{f}_j(\mathbf{x})$ denotes the normalized objective value of individual $\mathbf{x}$ in the jth objective, and M is the number of objectives.

This aggregation function is determined by two factors: first the number of objectives, and second the performance in each objective. Given a minimization MaOP, an individual with good convergence (that has slightly worse value in at least one objective but has significantly better value in most of the other objectives) is more likely to obtain a lower (better) aggregation value. A smaller aggregation value of an individual could indicate a good convergence performance. If an individual/solution is a dominance resistant solution, it is more likely to obtain an extremely large aggregation value compared to other individuals in a population. Keep this in mind, the aggregation function will be involved later in the definition of aggregation-based dominance relation, where the aggregated information of an individual is compared with others.

3.2.3 Aggregation-based Dominance

In a minimization MaOP with M objectives, ADO is defined on population P, where each individual $\mathbf{x}$ is assigned an aggregation value representing its estimated convergence performance, denoted as $f^{con}(\mathbf{x})$.

Definition 3.2.1 (Convergence Difference)**.** *Let* $\mathbf{x}$*,* $\mathbf{y} \in P$ *where* $\mathbf{x}$ *and* $\mathbf{y}$ *are nondominated solutions with respect to Pareto dominance, the convergence difference between them is denoted as:*

$$CD(\mathbf{x}, \mathbf{y}) = f^{con}(\mathbf{x}) - k \cdot f^{con}(\mathbf{y}) \tag{3.3}$$

k is a predefined parameter $[0 < k < 1]$*. With the definition of* $CD(\cdot)$*, aggregation-based dominance is defined as follows:*

Definition 3.2.2 (Aggregation-based Dominance)**.** *Let* $\mathbf{x}$*,* $\mathbf{y} \in P$ *where* $\mathbf{x}$ *and* $\mathbf{y}$ *are nondominated solutions with respect to Pareto dominance.* $\mathbf{x}$ *is said to ADO-dominate* $\mathbf{y}$*, or equivalently* $\mathbf{y}$ *is ADO-dominated by* $\mathbf{x}$*, denoted by* $\mathbf{x} \prec_{ADO} \mathbf{y}$*, if*

$$CD(\mathbf{x}, \mathbf{y}) < 0 \quad and \quad CD(\mathbf{y}, \mathbf{x}) > 0 \tag{3.4}$$

Similarly, $\mathbf{y}$ *ADO-dominates* $\mathbf{x}$ *when* $CD(\mathbf{y},\, \mathbf{x}) < 0$ *and* $CD(\mathbf{x},\, \mathbf{y}) > 0$*. Otherwise, both individuals* $\mathbf{x}$ *and* $\mathbf{y}$ *are ADO-nondominated solutions. It is clear that there is no possibility of both* $CD(\mathbf{x},\, \mathbf{y}) < 0$ *and* $CD(\mathbf{y},\, \mathbf{x}) < 0$ *at the same time.*

ADO is partly inspired by the idea of using additional convergence-related criterion in addition to traditional Pareto dominance based criterion, such as knee points in KnEA [157], a grid-dominance-based criterion defined in [152] and so on. The proposed ADO is used as a secondary criterion and is activated when the Pareto dominance-based selection criterion fails to distinguish between individuals in the evolutionary process. For ADO integrated with traditional Pareto-based algorithms, such as NSGA-II, an

ADO-nondominated sorting is employed in the selection procedure to classify P into different ADO-nondomination levels, which will be introduced in the next section.

Similar to ϵ-dominance [84] in the ϵ-dominance-based algorithm [38], the ADO also modifies the traditional Pareto dominance to enhance the selection pressure towards the Pareto front. Both of them can be categorized into a relaxed form of Pareto dominance, which are used to determine the survival of individuals in the evolutionary process. However, the difference is that ϵ-dominance enlarges the dominating space of each individual in the population, while ADO is cooperated with Pareto dominance and designed for those nondominated individuals.

3.3 Integrating ADO into NSGA-II and SPEA2+SDE

In this section, ADO is applied to two representative Pareto-based algorithms, NSGA-II and SPEA2+SDE, denoted as NSGA-II-ADO and SPEA2+SDE-ADO, respectively. Specifically, based on ADO, three modified evolutionary operators: fitness assignment, mating selection, and environmental selection are proposed to facilitate the search for high-quality solutions. In the next three sections, the three evolutionary operators are introduced respectively.

3.3.1 Fitness Assignment

In Pareto-based algorithms, the fitness of individuals should cover the performance of each individual in terms of *both* convergence and diversity in order to converge the population towards the Pareto front as well as diversify the individuals over the front. To deal with ADP phenomenon of Pareto-based algorithms as described in 2.3.2, the fitness assignment strategy in this work focuses on the estimation of convergence by incorporating ADO into Pareto domination-based fitness assignment strategies. In the proposed fitness assignment strategy, Pareto dominance and ADO are used together to evaluate the performance of individuals in terms of convergence, which works by applying ADO when Pareto dominance fails to distinguish individuals in a population. There are different fitness assignment strategies based on Pareto dominance to calculate the performance of individuals. In the experimental study, ADO is tested on the fitness assignment strategies of two popular Pareto-based evolutionary algorithms: NSGA-II and SPEA2+SDE.

The first tested strategy in the experiments is the non-dominated sorting-based fitness assignment of NSGA-II [37]. It suggested using a Pareto dominance relation to distinguish individuals in a mixed set of population. In this strategy, those solutions that are non-dominated in terms of Pareto optimal become the first front. After removing these solutions temporarily, the remaining non-dominated solutions constitute the second front, and so on. Finally, each individual in the same front is assigned a non-domination rank. Lower rank values represent better degrees of convergence for individuals. The final fitness of an individual depends on its nondomination rank (represents convergence information) and the crowding distance in (represents diversity information). Compared to the fitness assignment of the proposed NSGA-II-ADO, the difference is on the non-domination rank calculation of individuals in NSGA-II. Those individuals in each front depend on both Pareto dominance and aggregation-based dominance.

The second tested strategy in the experiments is a fine-grained fitness assignment, which was proposed in SPEA2 [163]. Compared with non-dominated sorting, this strategy only uses total fitness to include both convergence and density information of an individual in a population. In the fitness assignment process, an individual $\mathbf{x}$ is assigned a "strength" value which reflects its domination degree. The strength is calculated based on Pareto dominance counts: the number of solutions it dominates. Then, individual $\mathbf{x}$ is assigned a raw fitness $R(\mathbf{x})$ which is determined by summing strength of the individuals that dominate it. A low raw fitness value means that an individual dominates many individuals which in turn is dominated by many individuals. In particular, non-dominated individuals are assigned value zero as raw fitness. This is followed by the density estimation $D(\mathbf{x})$ of individual $\mathbf{x}$ that takes k-th nearest neighbor of individual $\mathbf{x}$ into consideration. Finally, the total fitness $F(\mathbf{x})$ of individual $\mathbf{x}$ is calculated by the equation: $F(\mathbf{x}) = R(\mathbf{x}) + D(\mathbf{x})$. For SPEA2+SDE, the only difference is the density estimation of an individual $\mathbf{x}$, where $D(\mathbf{x})$ is calculated based on the shifted position of other individuals in a population according to the convergence comparison between individual **x** and other individuals on each objective. In contrast to the fitness assignment of SPEA2+SDE-ADO, the strength value of an individual in SPEA2+SDE is calculated according to both Pareto dominance and aggregation-based dominance among individuals in a population.

In the fitness assignment process, the diversity estimation also plays an important role. It aims to obtain a set of well-distributed solutions along the entire Pareto front. In the proposed fitness assignment, density estimation techniques remain the same as those in the two Pareto-based evolutionary algorithms (i.e., crowding distance in NSGA-II [37] and Shift-Based Density Estimation (SDE) in SPEA2+SDE [95]).

In summary, the main difference in the fitness assignment process by incorporating the ADO is the evaluation of convergence for an individual. It leads to some changes of calculating non-domination rank of individuals in NSGA-II and strength value of individuals in SPEA2+SDE. The rest is applied in the usual way as proposed in the original NSGA-II and SPEA2+SDE. For the NSGA-II-ADO algorithm, the fitness of individuals includes both convergence and diversity information. In general, a smaller value of convergence and a larger value of diversity indicate that the individual performs better than others. In SPEA2+SDE+ADO, a smaller fitness value means the individual has better convergence and diversity than others. In the following, the impact of ADO in the fitness assignment process will be evaluated by considering a sorting strategy in NSGA-II-ADO as an example.

Consider a population of four nondominated individuals in a particular generation, including **A** (10, 17), **B** (1, 18), **C** (11, 6) and **D** (18, 2) in a bi-objective minimization scenario, as shown in Figure 3.2.

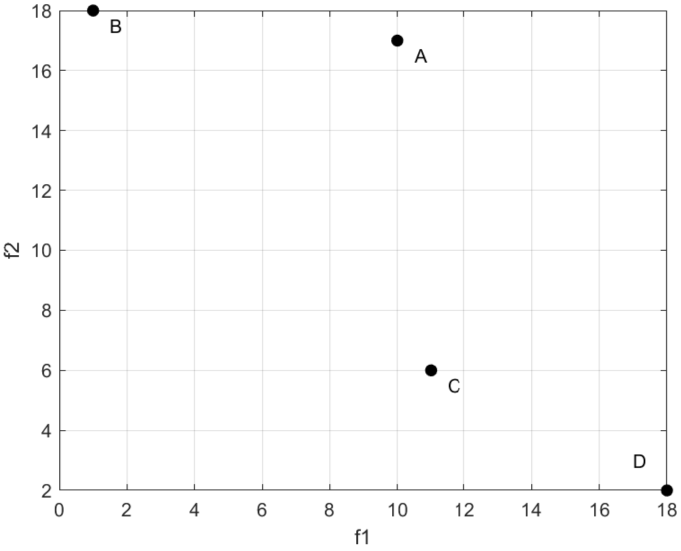


Figure 3.2: Four nondominated individuals in a bi-objective minimization scenario

Table 3.1 shows the ranking assignment results of an individual **A** in a typical bi-objective minimization scenario. In particular, the columns f1 and f2 represent two objective values of each individual, while columns $\widetilde{f_1}$ and $\widetilde{f_2}$ are the corresponding normalized objective values of each individual respectively. The column $f^{con}(\cdot)$ is the aggregation-based fitness value of an individual, which is calculated by simply summing up all normalized objective values of that individual. The column Comparison lists the ADO-based comparison results for two individuals. If there is an individual $\mathbf{x}$ ADO-dominates individual $\mathbf{y}$, namely $\mathbf{x} \prec_{ADO} \mathbf{y}$, $\mathbf{y}$ will be assigned with a rank equal to the rank of $\mathbf{x}$ adding 1, which will be recorded in column Rank. By doing so, an individual with better convergence will be assigned a lower value, and it will be more likely to survive into the next generation.

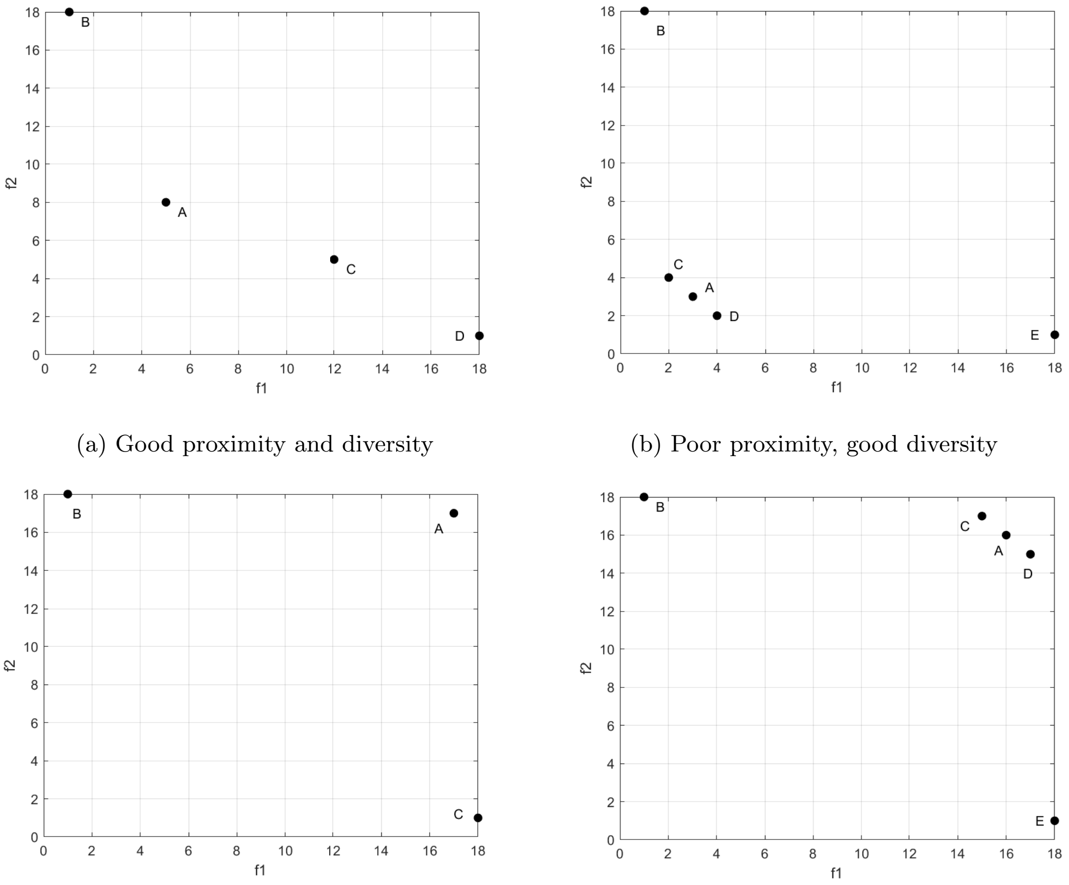
In the fitness assignment process of NSGA-II-ADO, a lower rank of an individual implies better fitness degree. Considering two nondominated individuals **A** and **C** in Table 3.1, **A**, which is ADO-dominated by **C** ($CD(\mathbf{C},\ \mathbf{A}) = 0.84 - k \times 1.47 < 0$, $CD(\mathbf{A},\ \mathbf{C}) = 1.47 - k \times 0.84 > 0$ assuming $k = 0.6$), will have a larger rank value than **C** (the rank value of **A** (2) equals to the rank value of **C** (1) adding 1 [see Table

Table 3.1: The ranking assignment results of an individual in a typical bi-objective minimization scenario by using the ADO.

Individuals	(f1, f2)	$\widetilde{f}_1$	$\widetilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(10, 17)	0.53	0.94	1.47	–	2
B	(1, 18)	0	1	1	–	1
C	(11, 6)	0.59	0.25	0.84	C $\prec_{ADO}$ A	1
D	(18, 2)	1	0	1	–	1

3.1]). Namely, **A** has worse fitness degree compared with C and it is more likely to be removed from the population in the following evolutionary process. In this scenario, individual **A** is a dominance resistant solution since there are two individuals **B** and **C** performing significantly better than individual **A** in one objective but slightly worse than **A** in the other objective. Therefore, this implies ADO could distinguish between dominance resistant solutions.

In order to explore the performance of ADO, we then consider four typical situations of individual **A** in a population for a bi-objective minimization problem, namely, (a) good convergence and diversity, (b) good convergence and poor diversity, (c) poor convergence and good diversity, and (d) poor convergence and diversity, as summarized in Figure 3.3. It should be noted that these individuals in the population are nondominated with each other.



(a) Good proximity and diversity

(b) Poor proximity, good diversity

(c) Good proximity, poor diversity

(d) Poor proximity and diversity

Figure 3.3: Four situations of individual **A** in a population for a bi-objective minimization problem.

Tables 3.2–3.5 show the ranking assignment results of individual **A** of four typical bi-objective minimization scenarios, which correspond to four situations described in Figure 3.3, namely: (a) good convergence and diversity, (b) good convergence and poor diversity, (c) poor convergence and good diversity, and (d) poor convergence and diversity.

Table 3.2: The ranking assignment results of an individual **A** using the ADO ($k = 0.5$) in a typical bi-objective minimization scenarios that has good convergence and diversity.

Individuals	(f1, 2f2)	$\widetilde{f_1}$	$\widetilde{f_2}$	$f^{con}(\cdot)$	Comparison	Rank
A	(5, 8)	0.24	0.41	0.65	–	1
B	(1, 18)	0	1	1	–	1
C	(12, 5)	0.65	0.24	0.89	–	1
D	(18, 1)	1	0	1	–	1

Table 3.3: The ranking assignment results of an individual **A** using the ADO ($k = 0.5$) in a typical bi-objective minimization scenarios that has good convergence and poor diversity.

Individuals	(f1, f2)	$\widetilde{f_1}$	$\widetilde{f_2}$	$f^{con}(\cdot)$	Comparison	Rank
A	(3, 3)	0.12	0.12	0.24	–	1
B	(1, 18)	0	1	1	B $\prec_{ADO}$ A	2
C	(2, 4)	0.06	0.18	0.24	–	1
D	(4, 2)	0.18	0.06	0.24	–	1
E	(18, 1)	1	0	1	E $\prec_{ADO}$ A	2

Table 3.4: The ranking assignment results of an individual **A** using the ADO ($k = 0.6$) in a typical bi-objective minimization scenarios that has poor convergence and good diversity.

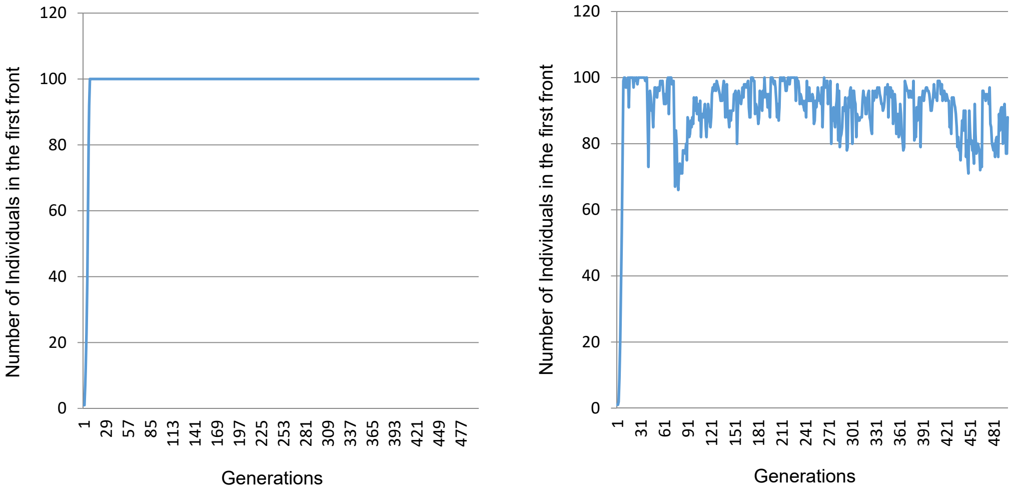
Individuals	(f1, f2)	$\widetilde{f_1}$	$\widetilde{f_2}$	$f^{con}(\cdot)$	Comparison	Rank
A	(17, 17)	0.94	0.94	1.88	–	2
B	(1, 18)	0	1	1	B $\prec_{ADO}$ A	1
C	(18, 1)	1	0	1	C $\prec_{ADO}$ A	1

Table 3.5: The ranking assignment results of an individual **A** using the ADO ($k = 0.6$) in a typical bi-objective minimization scenarios that has poor convergence and diversity.

Individuals	(f1, f2)	$\widetilde{f}_1$	$\widetilde{f}_2$	$f^{con}(\cdot)$	Comparison	Rank
A	(16, 16)	0.88	0.88	1.76	–	2
B	(1, 18)	0	1	1	B $\prec_{ADO}$ A	1
C	(15, 17)	0.82	0.94	1.76	B $\prec_{ADO}$ C	2
D	(17, 15)	0.94	0.82	1.76	B $\prec_{ADO}$ D	2
E	(18, 1)	1	0	1	E $\prec_{ADO}$ A	1

As shown in Tables 3.2 and 3.3, an individual with either both good convergence and diversity, or good convergence but poor diversity will be assigned a low-rank value. Although the two types of individuals have a high possibility of being preserved in the same front, this issue could be solved by the diversity maintenance techniques of different Pareto-based algorithms. In addition, the individuals with poor convergence will be assigned a high-rank (poor) value during the search process no matter how well the performance of diversity is [see Tables 3.4 and 3.5], thus being more likely to be removed during the evolutionary process. Overall, by incorporating ADO, more selection pressure could be provided among nondominated individuals in Pareto-based algorithms, and individuals with both good convergence and diversity [see Figure 3.3 (a)] have the highest chance to survive in the next generation.

Finally, this section examines the ability of ADO for Pareto-based algorithms in dealing with the ADP phenomenon caused by dominance resistant solutions. Figure 3.4 gives the comparative results of the NSGA-II-SIP (NSGA-II based on the SIP method [64]) and NSGA-II-ADO (NSGA-II embedded with the ADO method) by plotting the changes of the number of individuals in the first front in each generation in one run. Clearly, in NSGA-II-ADO, the individuals that belong to the first front continuously change during the evolutionary process, whereas in NSGA-II-SIP all individuals of the population belong to the first front after a few generations. This could be fully attributed to the incorporation of the proposed fitness assignment strategy, which can continuously facilitate the convergence of the population during the evolutionary process. Therefore, it indicates that the proposed ADO could overcome the ADP phenomenon of Pareto-based algorithms to some extent.



(a) NSGA-II-SIP

(b) NSGA-II-ADO

Figure 3.4: The number of individuals in the first front in each generation in one typical run of (a) NSGA-II-SIP (b) NSGA-II-ADO on the feature model - Drupal, respectively. Population size is 100.

3.3.2 Mating Selection with Constraint Handling

In mating selection, a binary tournament selection strategy based on the dominance relation and density information is often used to select promising individuals from the current population for reproduction. In NSGA-II-ADO, mating selection is enhanced by considering four criteria, namely the number of constraint violations (NCV), Pareto dominance, ADO, and crowding distance. Because the search space for SPLs includes a large number of constraints, a constraint handling strategy is introduced by taking into account NCV as the first criterion to check. In addition, the convergence (evaluated by Pareto dominance and ADO) and diversity information (evaluated by crowding distance) of individuals are considered. Algorithm 3.2 gives a detailed procedure of this binary tournament selection in NSGA-II-ADO.

Algorithm 3.2 $Mating\_selection(P)$

```
Require: individuals $a$, $b$ randomly chosen from the population
 1: $Q \leftarrow \varnothing$            /* New population set for crossover and mutation*/
 2: if $NCV(a) < NCV(b)$ then
 3:    $Q \leftarrow Q \cup \{a\}$
 4: else if $NCV(b) < NCV(a)$ then
 5:    $Q \leftarrow Q \cup \{b\}$
 6: else
 7:    if $a \prec b$ then
 8:       $Q \leftarrow Q \cup \{a\}$
 9:    else if $b \prec a$ then
10:       $Q \leftarrow Q \cup \{b\}$
11:    else
12:       if $a \prec_{ADO} b$ then
13:          $Q \leftarrow Q \cup \{a\}$
14:       else if $b \prec_{ADO} a$ then
15:          $Q \leftarrow Q \cup \{b\}$
16:       else
17:          if $crowd\_distance(a) > crowd\_distance(b)$ then
18:             $Q \leftarrow Q \cup \{a\}$
19:          else if $crowd\_distance(b) > crowd\_distance(a)$ then
20:             $Q \leftarrow Q \cup \{b\}$
21:          else
22:             if $random(0,1) < 0.5$ then
23:                $Q \leftarrow Q \cup \{a\}$
24:             else
25:                $Q \leftarrow Q \cup \{b\}$
26:             end if
27:          end if
28:       end if
29:    end if
30: end if
31: return $Q$
```

In the binary tournament mating pool, two individuals are randomly selected from the current population. If an individual has fewer constraint violations than the other, then the former is chosen (lines 2–5). For individuals that have the same number of constraints, then the one that dominates the other is preferred (lines 7–10). If the two individuals are nondominated with respect to each other, the one that ADO-dominates the other wins the tournament (lines 12–15). If these both individuals are ADO-nondominated to each other, then crowding distance is used for comparison (lines 17–20), and the individual with the larger crowding distance value (i.e., the individual is located in a sparse area) is chosen. When crowding distance still cannot distinguish between the two individuals, one of them will be randomly chosen for reproduction (lines 22–25).

3.3.3 Environmental Selection with Constraint Handling

Environmental selection preserves a group of best solutions in the population found so far as the parent population for the next generation. The proposed environmental selection in NSGA-II-ADO adopts a three-level-sorting method to assign rank values to individuals in the population, with aim of promoting convergence of the population and handling the constraints included in SPLs. The three-level-sorting method is based on three criteria: the number of constraint violations as the primary criterion, followed by Pareto-dominance and ADO.

Algorithm 3.3 illustrates the framework of the environmental selection procedure in NSGA-II-ADO for many-objective optimization. First, the combined set (denoted as S) of the current population and their offspring population is normalized, and the aggregation value of each individual in S is calculated (line 2). Then, the three-level-sorting is performed to divide the solution set S into l fronts (line 3).

Algorithm 3.3 $Environmental\_selection(S)$

```
Require: $S$ (combined set of the current population and their offspring population), $N$ (population size)
 1: $Q \leftarrow \varnothing$          /* Parent population for the next generation */
 2: $Normalization(S)$
 3: $(F_1, F_2, \ldots, F_l) \leftarrow Three\_level\_sorting(S)$
 4: while $|Q| + |F_i| < N$ do
 5:     $Q \leftarrow Q \cup F_i$          /* Add $i$th front in the parent population $Q$ */
 6:     $i \leftarrow i + 1$          /* Check the next front for inclusion */
 7: end while
 8: if $|Q| = N$ then
 9:     return $Q$
10: else
11:     $sort(F_i, \prec_n)$          /* Sort front $F_i$ in descending order using $\prec_n$ */
12:     $Q \leftarrow Q \cup F_i[\,1 : (N - |Q|)\,]$          /* Add first $(N - |Q|)$ individuals from $F_i$ */
13: end if
14: return $Q$
```

The three-level-sorting method can be seen as a complex nondominated sorting method proposed in NSGA-II. It works as follows: (1) the population S is sorted according to the number of constraint violations, and S is divided into different fronts, where each front contains the same number of constraint violations. Thus, the one with fewer constraint violations belongs to the first layer. Those with the second fewest number of violated constraints belong to the second layer, and so on. The purpose of this is to quickly drive the population to feasible regions, namely finding valid products (or valid solutions) and keeping those valid solutions in the next generation. (2) For those individuals with the same number of violated constraints in front F_a, Pareto-dominance criterion is activated to further classify front F_a into n nondominated fronts (F'_a, F'_{a+1}, ..., F'_{a+n}). (3) For those nondominated individuals in new front F'_a, ADO is applied to further distinguish them and divide front F'_a into m AD-nondominated fronts (F''_a, F''_{a+1}, ..., F''_{a+m}).

After performing the three-level-sorting, l fronts (F_1, F_2, ..., F_l) are selected one by one to form the next-generation population Q. Individuals which belong to the first front (F_1) are the most promising individuals in S. If the size of F_1, denoted as $|F_1|$ is equal to population size N, all individuals in F_1 are chosen. In case that $|F_1|$ is smaller than population size N, the remaining individuals in S are chosen from the subsequent fronts according to the front rank. That is, NSGA-II-ADO turns to the second front (F_2) for choosing the remaining individuals, followed by adding the individuals from the third front (F_3), and so on (lines 4–7). By doing this, individuals with a lower number of constraint violations would always be included first to form the new population. Thus, NSGA-II-ADO could quickly search for valid individuals (product configurations in an SPL).

If the number of individuals in the last included front (denoted as F_i) is larger than $N - |Q|$, then a crowded-comparison operator denoted as $\prec_n$ (the same in NSGA-II) is applied to sort individuals in $|F_i|$ in descending order (line 11). After that, the first $(N - |Q|)$ individuals in sorted F_i are chosen to fill up the population for the next generation (line 12).

3.4 Experimental Results

In this section, the improvement for Pareto-based evolutionary algorithms by using three enhanced evolution operators is investigated. First, this section introduces the experimental design, which includes the description of subject models and optimization objectives, competitor algorithms, the basic parameter settings and system description, and performance metrics. Next, this section compares NSGA-II-ADO and SPEA2+SDE-ADO with four different types of MOEAs based on SIP method. After that, this section discusses the findings on time comparison, hypervolume comparison, and parameter sensitivity.

3.4.1 Experimental Design

SPL Feature Model

The characteristics of nine SPL feature models used in this empirical study are summarized in Table 3.6. For each feature model, it presents the number of features, the number of fixed features (core and parents features with the mandatory or alternative relationship), the number of Cross-Tree Constraints (CTC), and the number of optimization objectives (Objectives).

Table 3.6: SPL feature models summary.

Feature models	# Total features	#Fixed features	#CTC	#Objectives
BerkeleyDB [125]	13	3	0	4
ERS [43]	36	11	0	4
Web Portal [109]	43	15	6	4
E-Shop [83]	290	88	21	4
Drupal [117]	48	9	21	4
Amazon EC2 [49]	79	20	0	4
Random-10000 [110]	10,000	4,078	0	4
DrupalReal [117]	48	9	21	7
AmazonEC2Real [49]	79	20	0	7

These SPL feature models were obtained from the recently published SPL literature. The *BerkeleyDB* model describes the variability of a database system. *ERS* is a feature model for an Emergency Response System. Two feature models *Web Portal* and *E-Shop* are from the widely used SPLOT feature model repository [110]. The *Web portal* is the feature model for a web portal product line. The *E-Shop* is the largest feature model in SPLOT with 290 features. The *Drupal* and *Amazon EC2* are two recently published feature models, which represent the variability of open-source web content management framework and Amazon Elastic Computing Service, respectively. The *Random-10000* model is a randomly generated feature model with $10,000$ features, which is the largest model in the experiments to evaluate the scalability of proposed ADO in Pareto-based algorithms. Two feature models *DrupalReal* and *AmazonEC2Real* are derived versions of *Drupal* and *Amazon EC2* with realistic attribute values, respectively.

In Table 3.6, for those feature models with four optimization objectives, the following two optimization objectives are considered to be maximized [121, 62, 64]:

- *Richness of features.* How many features that are selected in a configuration.
- *Features that were used before.* How many features that were used before in a configuration, i.e., the number of "true" for this attribute.

In addition, two objectives should be minimized [121, 62, 64]:

- *Known defects.* How many known defects in a configuration.
- *Cost.* The total cost of a configuration.

For *DrupalReal* with seven objectives, the following three optimization objectives should be maximized [64]:

- *The richness of features.* How many features that are selected in a configuration.
- *Test assertions.* How many test assertions of each feature.
- *The number of reported installations.* How many times a feature has been installed as reported by Drupal users.

In addition, four objectives should be minimized [64]:

- *The number of lines of code.*
- *Cyclomatic complexity.* How many independent logic paths used in a program.
- *The number of developers.* How many developers is involved in the development of each DrupalReal feature.

- *The number of changes.* How many changes are made in each feature.

For *AmazonEC2Real* with seven objectives, five optimization objectives should be maximized [64]:

- *The richness of features.* How many features that are selected in a configuration.
- *Instance.cores.* How many cores of the instance. Randomly generated value between 1 and 32.
- *Instance.ecu.* The Amazon EC2 Compute Unites. Randomly generated value between 0 and 108.
- *Instance.ram.* The memory of the instance. Randomly generated value between 0 and 250.
- *Instance.ssBacked.* If the instance storage is SSD backed. Boolean.

In addition, two objectives should be minimized [64]:

- *EC2.costMonth.* Randomly generated value between 0 and $20,000$.
- *Instance.costHour.* Randomly generated value between 0 and 18.

Finally, many recent studies also consider correctness (the number of violated constraints in a configuration) as an optimization objective that should be minimized. However, in the experiments, correctness is used to handle constrained search space of feature models rather than an objective (see more explanations in Sections 3.3.2 and 3.3.3).

Representation

A representation (called novel encoding) introduced in [64] is to represent a product as a binary string. Representation of products is not the main contribution of this chapter, but it does impact the performance of search techniques. Compared with some other representations [121, 120, 62], the novel encoding is able to avoid producing invalid solutions and reduce search space because it simplifies the representation of a product.

Competitor Algorithms and Parameter Settings

Since there was no clear winner of MOEAs using the SIP method [64], The modified algorithms (i.e., NSGA-II-ADO and SPEA2+SDE-ADO) were compared with four SIP-based MOEAs (i.e., NSGA-II-SIP, SPEA2+SDE-SIP, IBEA-SIP, and MOEA/D-TCH-SIP) taken from [64]. To facilitate a fair comparison with the advances for solving optimal feature selection problem in SPLs, settings used in the experiments were the same as those in [64]. Settings for all the MOEAs are:

- 30 runs for each algorithm per feature model to decrease the impact of their stochastic nature.
- The termination criterion was a predefined maximum of $50,000$ evaluations.
- For crossover and mutation operators, uniform crossover and bit-flip mutation were used, with crossover and mutation probability set to 1.0 and $1/n$, respectively, where n represents the number of decision variables.

There are some special configurations specific to certain MOEAs.

- The size of population for NSGA-II, SPEA2+SDE, and IBEA, was set to 100, while the size of population for MOEA/D-TCH was set to 126 and 120 for four- and seven-objective optimization problems, respectively.
- We set the scaling factor k in IBEA to 0.05 as suggested in the original paper [162], and the neighborhood size to 10 percent of the population size in MOEA/D-TCH as suggested in the original paper [156].
- The parameter used in the proposed ADO was set to 0.5 for both NSGA-II-ADO and SPEA2+SDE-ADO.

All algorithms were implemented in **C** and all the experiments were conducted on an Intel(R) Core(TM)$i5-2500$ CPU @ 3.30GHz with 4 GB RAM, running on Windows 7.

Performance Assessment

Four performance metrics were adopted to evaluate the performance of algorithms.

- VN - the number of runs for each algorithm that returns at least one valid solution (i.e., without violating any constraints) in the final solution set [64]. This is used to evaluate the ability of an algorithm to find valid products from a feature model. The run with invalid solutions are discarded in the experiments, since invalid product configuration is useless in practice.

- VP - the proportion of valid distinct solutions in the final population [64]. A high value of VP is preferred, as more options are provided to software engineers.

- TT100% - the time to reach 100% valid solutions in a population. This quality indicator was first introduced by Sayyad et al. [120]. It measures the speed of convergence to a large number of points within the Pareto front that have no violations. It will be calculated when the VP values for different algorithms are the same.

- Hypervolume (HV), as a popular performance metric in evolutionary many-objective optimization, was used to evaluate how well a set of solutions fulfils the optimization objectives in terms of both convergence and diversity. It is suitable for practical many-objective optimization problems since it does not need a reference set to represent the Pareto front. It has been introduced in Chapter 2.2.3 in detail. Following the suggestions in [99], in this experiments, all the objective values of solutions were normalized to [0, 1] according to different ranges of a particular objective. Furthermore, the reference point was set to (1.0, 1.0, . . . , 1.0), which is constructed with the worst value on each objective, so-called Nadir point. A larger HV value is preferred because it reflects better quality of A in terms of convergence, diversity, and uniformity. Note that only those obtained nondominated solutions with VP of 100% were used to calculate HV.

3.4.2 Performance Comparison

Table 3.7 shows the mean results of six algorithms on all the nine feature models in terms of VN (\30), VP, the mean of TT100%, and the mean and standard deviation of HV metric among all the 30 runs, respectively. For each test feature model, among different algorithms, the algorithm that has the best result based on the specific metric is shown in bold. To investigate the advantages of the proposed algorithms in a statistical sense, Wilcoxon's rank-sum test [161] was performed between the two proposed algorithms (NSGA-II-ADO and SPEA2+SDE-ADO) and each of the competitor algorithms over each feature model at a 0.05 significance level. The symbols a, b, c indicate that the result of the algorithm in the corresponding column is significantly outperformed by NSGA-II-ADO, SPEA2+SDE-ADO, and both NSGA-II-ADO and SPEA2+SDE-ADO, respectively.

Usually, only valid products are useful in practice, and thus, these algorithms are first distinguished by their VN values and VP values (when some algorithms had the best VN). The larger value of VN indicates that an algorithm could provide more runs that return valid products than other algorithms. The larger value of VP means that more valid products are provided for software engineers. The run with no one valid product (i.e., 100 invalid solutions returned by an algorithm) will be discarded in the experiments. As can be seen in Table 3.7, almost all algorithms could obtain valid solutions (VN of 30 and VP of 100%) for all feature models (both with four and seven objectives). One exception is IBEA-SIP, which only returned valid solutions with a VP value of 100% in 12/30 runs for AmazonEC2Real. In addition, none of the algorithms could return valid solutions in all 30 runs for Random-10000. SPEA2+SDE-ADO returned valid solutions in 29 out of 30 runs (VN = 29), followed by NSGA-II-ADO and SPEA2+SDE-SIP (VN = 27), IBEA-SIP (VN = 25), NSGA-II-SIP (VN = 19), and MOEA/D-TCH-SIP (VN = 14).

The success of returning a large number of valid products in a search may be mainly attributed to 1) the usage of the recently developed *novel encoding* in [64], which could reduce the search space for producing invalid products, and 2) constraint handling strategies in both mating selection and environmental selection, which prefer

Table 3.7: Mean performance results of six algorithms on nine test feature models.

Feature model	Metric	NSGA-II-ADO	SPEA2+SDE-ADO	NSGA-II-SIP	SPEA2+SDE-SIP	IBEA-SIP	MOEA/D-TCH-SIP
BerkeleyDB	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	2.2	23.7	1.8	20.9^a	40.4^c	**0.8**
	HV	0.6730	**0.6773**	0.6394^c	0.6445^c	0.6381^c	0.6203^c
	Std. Dev.	0.0297	0.0344	0.0021	0	0.0012	0.0072
ERS	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	48.1	566.8	44.0	501.8^a	1018.4^c	**6.0**
	HV	0.4596	0.4488	**0.4643**	0.4531	0.3964^c	0.4220^c
	Std. Dev.	0.0095	0.0223	0.0056	0.0173	0.0143	0.0211
WebPortal	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	9.5	107.0	9.0	101.9^a	197.9^c	**2.4**
	HV	0.8067	**0.8518**	0.6087^c	0.6894^c	0.7067^c	0.6326^c
	Std. Dev.	0.0369	0.0469	0.0085	0.0144	0.016	0.0089
E-Shop	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	52.4	257.4	45.1	245.5^a	485.4^c	**22.8**
	HV	0.7657	**0.7882**	0.5828^c	0.6770^c	0.7180^c	0.6812^c
	Std. Dev.	0.0695	0.0445	0.0144	0.0169	0.0225	0.0218
Drupal	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	12.6	108.7	10.3	100.8^a	204.3^c	**2.6**
	HV	0.8952	**0.9136**	0.5815^c	0.6554^c	0.6976^c	0.6310^c
	Std. Dev.	0.0332	0.0367	0.0063	0.0175	0.0179	0.0119
AmazonEC2	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	76.2	671.9	72.6	628.3^a	1169.6^c	**24.2**
	HV	0.6451	**0.6862**	0.6518^b	0.6829	0.6082^c	0.4657^c
	Std. Dev.	0.0404	0.0498	0.0252	0.0439	0.0401	0.0548
Random-10000	VN(\30)	27	**29**	19	27	25	14
	TT100% (ms)	**28653.2**	36081.4	29438.3	35982.3^a	39675.2^c	37225.3^c
	HV	0.6068	**0.6546**	0.5211^c	0.6172^b	0.6485^c	0.6247
	Std. Dev.	0.0391	0.0513	0.0122	0.0291	0.0324	0.0418
DrupalReal	VN(\30)	**30**	**30**	**30**	**30**	**30**	**30**
	TT100% (ms)	13.4	122.1	11.8	113.7^a	1080.1^c	**2.9**
	HV	0.1943	**0.4008**	0.1912^b	0.3931	0.3688^b	0.3812^b
	Std. Dev.	0.0159	0.0198	0.016	0.02	0.0283	0.0037
AmazonEC2Real	VN(\30)	**30**	**30**	**30**	**30**	12	**30**
	TT100% (ms)	109.6	1305.6	**91.4**	841.8^a	11998.2^c	213.8^a
	HV	0.4233	**0.4730**	0.3992^c	0.4654	0.2820^c	0.2694^b
	Std. Dev.	0.0353	0.0451	0.0448	0.042	0.0632	0.0512

Note: The value of VP metric is 100% for all algorithms in 30 independent runs.

[a] indicates that the result is significantly outperformed by NSGA-II-ADO at a 0.05 significance level by the Wilcoxon's rank-sum test.

[b] indicates that the result is significantly outperformed by SPEA2+SDE-ADO at a 0.05 significance level by the Wilcoxon's rank-sum test.

[c] indicates that the result is significantly outperformed by both NSGA-II-ADO and SPEA2+SDE-ADO at a 0.05 significance level by the Wilcoxon's rank-sum test.

the solutions with less number of constraint violations. It is interesting to investigate the influence of representation method and the influence of constraint handling strategy in the future.

Due to the similar good performance in terms of VN and VP, the TT100% metric is used to further compare the execution time of these algorithms. Statistically, both NSGA-II-ADO and SPEA2+SDE-ADO perform better than SPEA2+SDE-SIP and IBEA-SIP on all tested feature models. MOEA/D-TCH-SIP is shown to be the fastest algorithm for seven of the nine feature models. NSGA-II-SIP is the second fastest algorithm for eight of the nine feature models and is the fastest algorithm for AmazonEC2Real. It should be noted that the execution of NSGA-II-ADO is slower than NSGA-II-SIP for almost all feature models but is the fastest algorithm for the largest feature model Random-10000, which only takes under an average of 29 seconds. In addition, although SPEA2+SDE-ADO performs worse than those competitor algorithms except IBEA-SIP in terms of TT100%, the longest execution time of SPEA2+SDE-ADO is under an average of 37 seconds for the largest feature model in the experiments. This indicates ADO is an efficient method in searching valid product configurations in SPLs.

Next, the quality of obtained solutions is investigated by comparing the HV metric values of algorithms. As shown in Table 3.7, the proposed SPEA2+SDE-ADO is the most effective tested algorithm with regard to the best results in eight test feature models out of nine (apart from ERS). The proposed NSGA-II-ADO performs slightly worse than SPEA2+SDE-ADO on those feature models (BerkeleyDB, WebPortal, E-Shop, Drupal) with four objectives, but significantly better than the four SIP-based MOEAs. Statistically, the proportion of the tested feature models where NSGA-II-ADO outperforms NSGA-II-SIP, SPEA2+SDE-SIP, IBEA-SIP, and MOEA/D-TCH-SIP is 6/9, 4/9, 8/9, and 6/9, respectively. Furthermore, the proportion of the tested feature models where SPEA2+SDE-ADO outperforms NSGA-II-SIP, SPEA2+SDE-SIP, IBEA-SIP, and MOEA/D-TCH-SIP is 8/9, 5/9, 9/9, and 9/9, respectively.

Table 3.8 summarizes the HV differences between NSGA-II-ADO and NSGA-II-SIP, and the differences between SPEA2+SDE-ADO and SPEA2+SDE-SIP on nine feature models. Positive values of HV indicate improvements, while negative values of HV indicate the opposite.

Table 3.8: HV differences between MOEAs based on the proposed ADO method and that based on the SIP method on nine feature models.

Feature models	NSGA-II (%)	SPEA2+SDE (%)
BerkeleyDB (4)	5.25	5.10
ERS (4)	-0.99	-0.94
WebPortal (4)	32.52	23.55
E-Shop (4)	31.39	16.43
Drupal (4)	53.96	39.38
Amazon EC2 (4)	-1.03	0.48
Random-10000 (4)	16.44	6.06
DrupalReal (7)	1.62	1.96
AmazonEC2Real (7)	6.03	1.63

As seen in Tables 3.7 and 3.8, the proposed NSGA-II-ADO performs better than NSGA-II-SIP in seven out of nine tested feature models (with statistical significance on six feature models). The best performance is on Drupal with an increase of 53.96%, followed by WebPortal (32.52%), E-Shop (31.39%), and Random-10000 (16.44%). For feature models with seven objectives, NSGA-II-ADO could also obtain better results with 1.62% and 6.03% improvement for DrupalReal and AmazonEC2Real, respectively. Although there is a decrease of 0.99% on ERS and 1.03% on Amazon EC2, it could be seen as similar results due to the randomness. In addition, the proposed SPEA2+SDE-ADO performs better than SPEA2+SDE-SIP in eight out of nine tested feature models (with statistical significance on five feature models). The largest percent of HV increase is 39.38% and only a very small deterioration (0.94%) on ERS. In summary, compared with the SIP method, the integration of ADO can help NSGA-II and SPEA2+SDE to generally improve the quality of obtained solutions for seven and eight out of nine feature models in this experiment, respectively.

Figures 3.5–3.13 plot the distribution of the HV values of all 30 runs for all algorithms on each feature model. The experimental results indicate that the ADO has clear advantages over SIP for the nine experimental subjects, ADO can largely improve the performance of Pareto-based algorithms to find high-quality solutions. Compared with NSGA-II and SPEA2+SDE based on the SIP method, using ADO could obtain more high-quality solutions, which is evident from the HV values shown in Table 3.7. In particular, SPEA2+SDE-ADO is the most effective MOEA on three feature models that proved to be most challenging for search, including those with a larger number of objectives (Drupal and Amazon with realistic attribute values) and the larger randomly generated model (Random-10000).

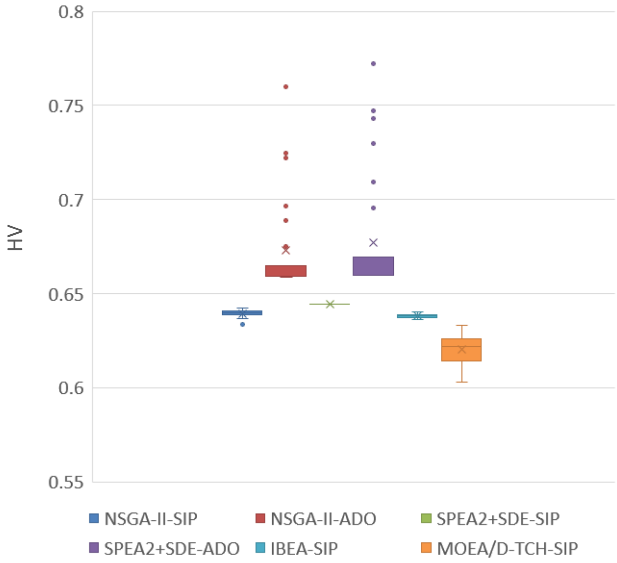


Figure 3.5: HV value comparison of six algorithms on the feature model - BerkeleyDB.

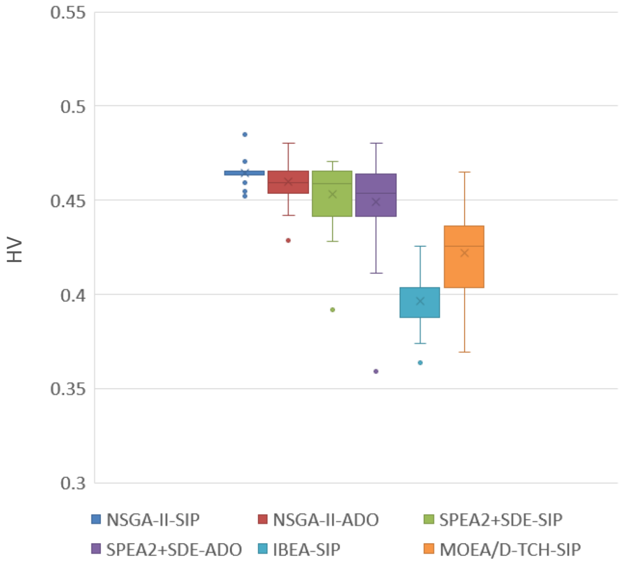


Figure 3.6: HV value comparison of six algorithms on the feature model - ERS.

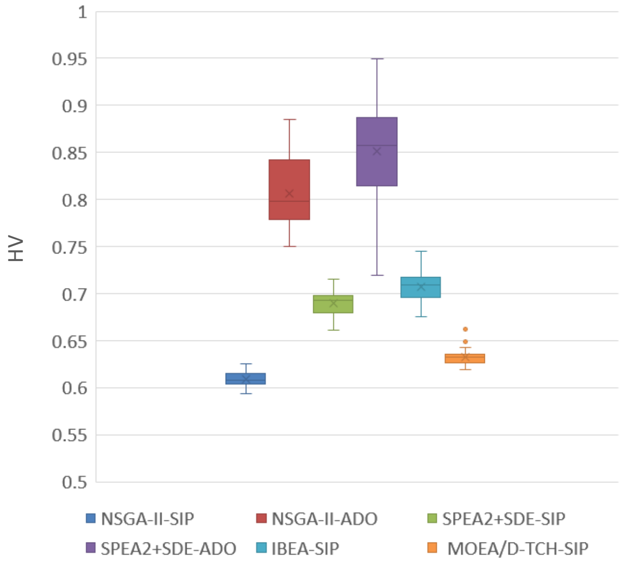


Figure 3.7: HV value comparison of six algorithms on the feature model - WebPortal.

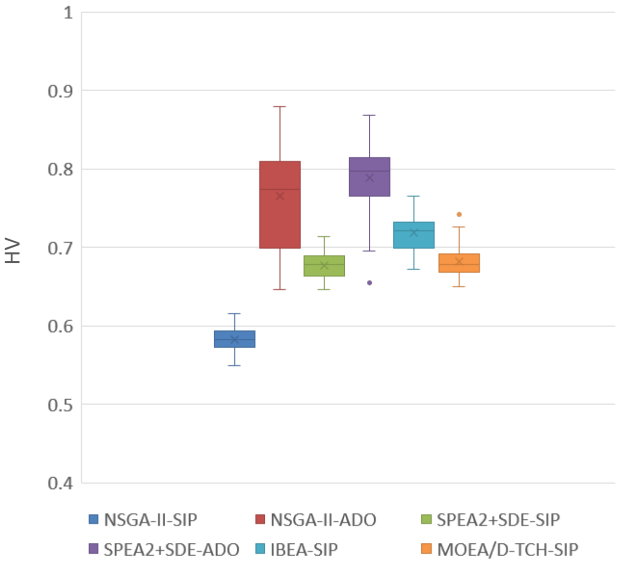


Figure 3.8: HV value comparison of six algorithms on the feature model - E-Shop.

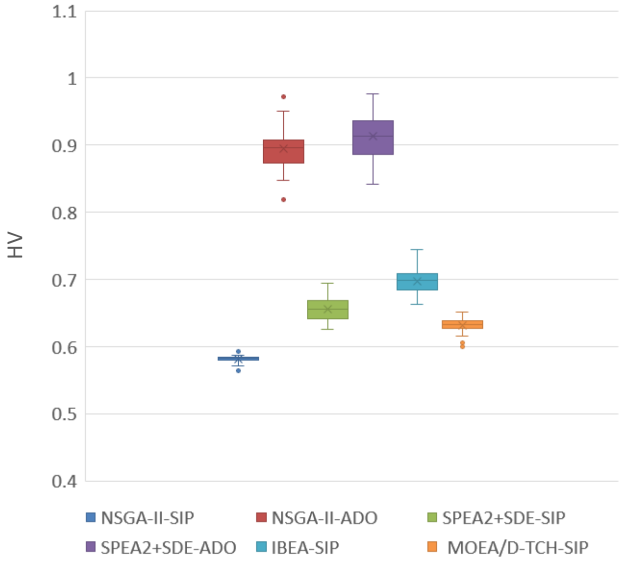


Figure 3.9: HV value comparison of six algorithms on the feature model - Drupal.

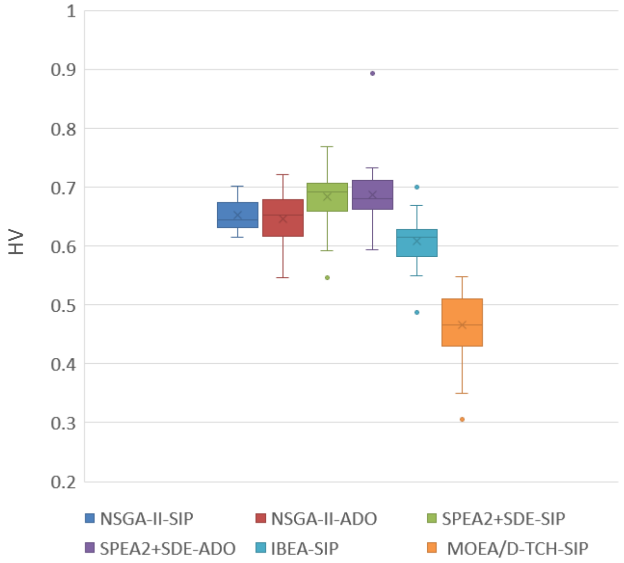


Figure 3.10: HV value comparison of six algorithms on the feature model - Amazon EC2.

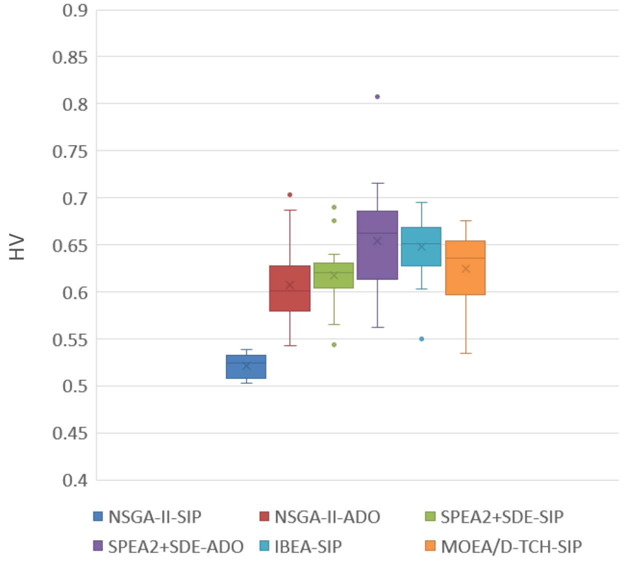


Figure 3.11: HV value comparison of six algorithms on the feature model - Random-10000.

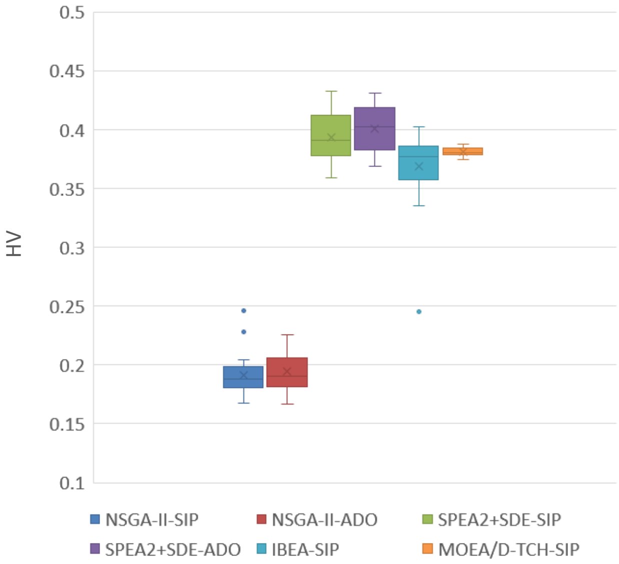


Figure 3.12: HV value comparison of six algorithms on the feature model - DrupalReal.

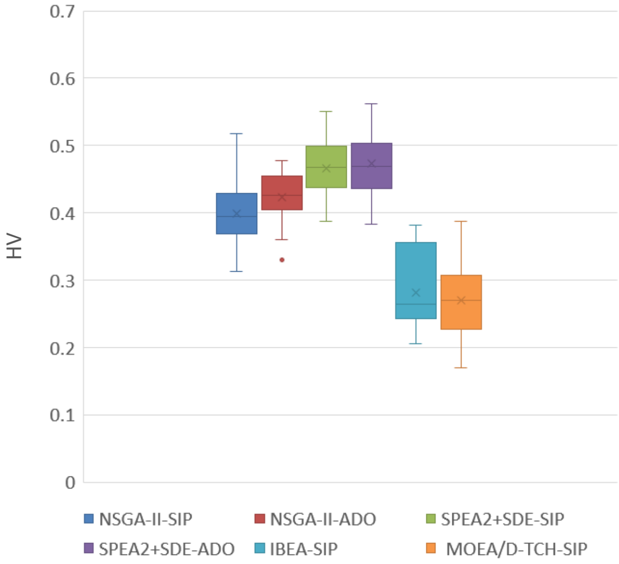


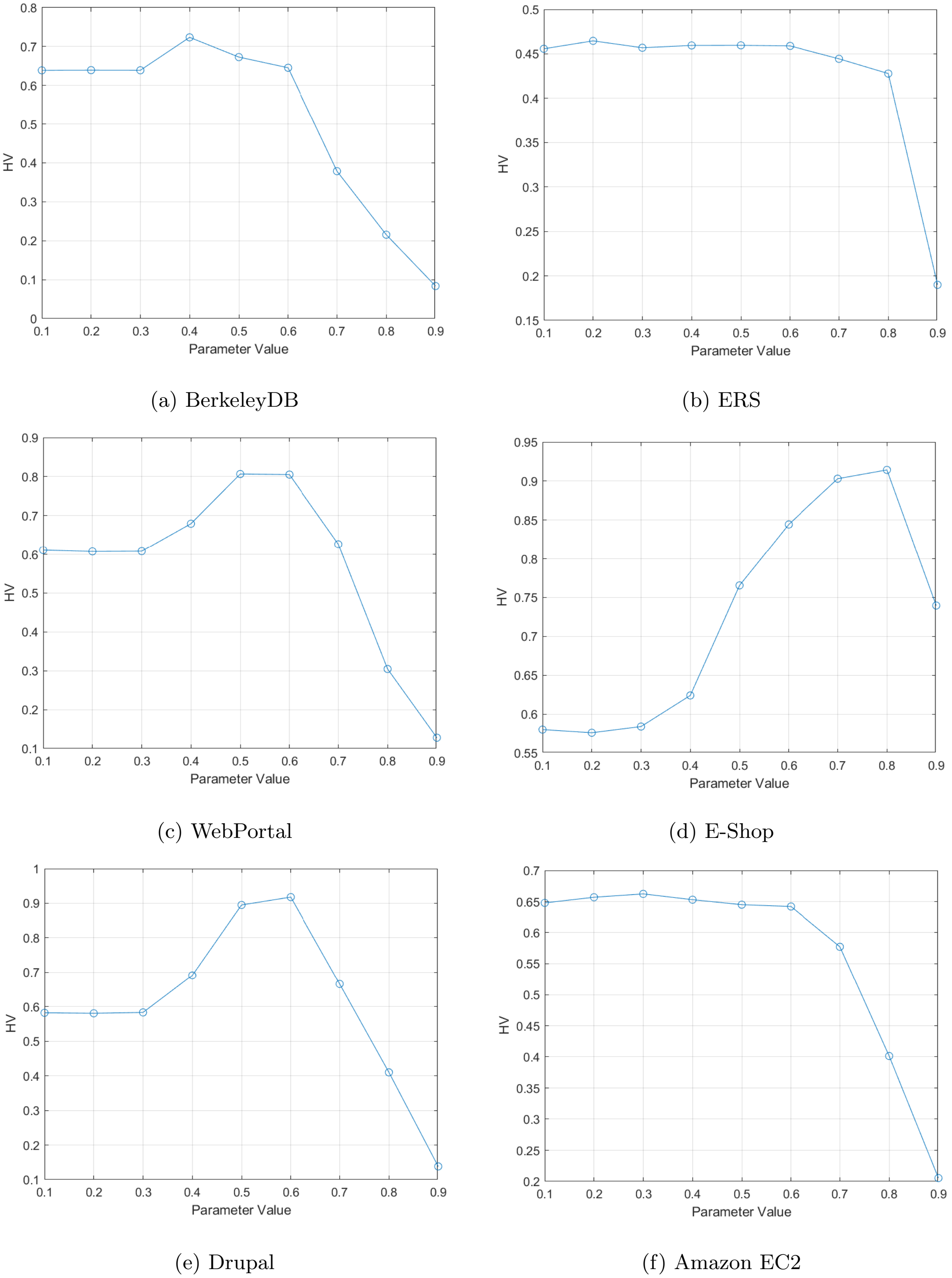
Figure 3.13: HV value comparison of six algorithms on the feature model - AmazonEC2Real.

3.4.3 Parameter Sensitivity Analysis

To evaluate the sensitivity of the parameter k in ADO, this section investigates how much the HV value fluctuates with the increase of the value of parameter k in the predefined intervals. Here, the results for NSGA-II-ADO on six feature models (BerkeleyDB, ERS, WebPortal, E-Shop, Drupal, and Amazon EC2) with four objectives are shown.

In Figure 3.14, relative changes of the HV values corresponding to the increases of k from 0.1 to 0.9 with a step of 0.1 for 30 independent runs and $50,000$ evaluations are plotted. In Figures 3.14 (a)–(f), almost all of the feature models reach roughly highest HV values when the parameter k value is between 0.5–0.6, except for BerkeleyDB in Figure 3.14 (a) with the value of k is 0.4, and E-Shop in Figure 3.14 (d) with the value of k is 0.8. In addition, for most of the feature models, the average of HV values is sharply reduced when the value of k exceeds 0.6. However, there is one exception E-Shop, as shown in Figure 3.14 (d), with its average HV value rising significantly with the k value grows from 0.3 to 0.8, followed by a significant decrease. WebPortal, as shown in Figure 3.14 (c) and Drupal, as shown in Figure 3.14 (e) have similar performance in that their average HV values change little with the increase of k from 0.1 to 0.3, and then a significant improvement of HV values with the increase of k from 0.3 to 0.5, which followed by a little fluctuate between range k from 0.5 to 0.6, and finally a significant decrease between range k from 0.6 to 0.9.

In general, the experiments in this section suggest that k should be set within a range of 0.5 to 0.6 in order to achieve a good balance between convergence and diversity. This is because small values have little influence on the final solutions set, while large value may lead to a subpopulation set that is limited in local optimal regions of the Pareto front. Although the HV value fluctuates a little between 0.5 and 0.6, the changes are small and the sensitivities are at acceptable levels. NSGA-II-ADO seems robust on the parameter k in the suggested ranges. High-quality solutions with regard to HV metric obtained by NSGA-II-ADO do not fluctuate much with the value of parameter k changes in the suggested range.



(a) BerkeleyDB

(b) ERS

(c) WebPortal

(d) E-Shop

(e) Drupal

(f) Amazon EC2

Figure 3.14: The curves of HV with regard to k values varying from 0.1 to 0.9 with a step of 0.1. The value of HV for each feature model is an average value of 30 independent runs generated by proposed NSGA-II-ADO.

3.5 Summary

The optimal feature selection problem, which can be seen as a many-objective optimization problem with a constrained search space, poses great challenges for multi-objective evolutionary algorithms. In this chapter, a new dominance relation, called *aggregation-based dominance* (ADO) for Pareto-based algorithms was proposed to enhance the search for high-quality solutions with respect to correctness and diversity in a reasonable amount of execution time. The proposed ADO was integrated into two widely Pareto-based algorithms (i.e., NSGA-II and SPEA2+SDE), resulting in two new algorithms NSGA-II-ADO and SPEA2+SDE-ADO.

This chapter carried out experiments to compare these two algorithms with four multi-objective evolutionary algorithms based on the state-of-the-art SIP method on nine different SPLs with up to $10,000$ features and two real-world SPLs with up to seven objectives. The experiments have shown the effectiveness and efficiency of both ADO-based NSGA-II and SPEA2+SDE. First, both algorithms could generate 100% valid solutions for all feature models. Second, the performance of both algorithms was improved as measured by the hypervolume metric in 7/9 and 8/9 feature models. Third, even for the largest tested feature model with $10,000$ features, it required under 40 s on a standard desktop to find 100% valid solutions in a single run of both algorithms.

In summary, this work has inherited strengths from existing Pareto-based algorithms while managing to address the weakness in dealing with complex SPL selection optimization problems. In particular, ADO is an effective method that is capable of accelerating the convergence information in order to enhance the selection pressure towards the Pareto front by reducing the detrimental impact caused by the active diversity promotion phenomenon. By embedding ADO into Pareto-based algorithms, these algorithms can achieve higher effectiveness with regard to convergence and diversity when solving many-objective optimization problems. Despite the above, this work does not limit itself to tackle a specific many-objective optimization problem. The next chapter attempts to enhance the effectiveness of leading evolutionary algorithms in order to tackle more generic cases.

Chapter 4

Angle-based Crowding Degree Estimation for Many-Objective Optimization

This chapter explores how the effectiveness of evolutionary algorithms may be enhanced for many-objective optimization. In particular, this chapter focuses on a very popular and widely cited method called BiGE [96], which adopts a meta-objective optimization approach. BiGE (Bi-Goal Evolution) maps solutions from the original high-dimensional objective space into a bi-goal space of proximity and crowding degree has received increasing attention in the area. It has many distinctive advantages, however, it has been found that BiGE tends to struggle on a class of many-objective problems where the search process involves *dominance resistant solutions*, namely, those solutions with an extremely poor value in at least one of the objectives but with (near) optimal values in some of the others.

In this chapter, an angle-based crowding degree estimation method for BiGE (denoted as aBiGE) is proposed to replace distance-based crowding degree estimation in BiGE. The rest of the chapter is organized as follows. In Section 4.1, the introduction of this work is given. Section 4.2 presents angle-based crowding degree estimation method and its incorporation with BiGE. The experimental results are detailed in Section 4.3.

Section 4.4 summarizes this chapter.

4.1 Introduction

Recently, a meta-objective optimization algorithm, called Bi-Goal Evolution (BiGE) [96] for many-objective optimization problems was proposed and has become the most cited paper in the Artificial Intelligence journal since its publication in November 2015 according to the Web of Science. BiGE is inspired by two observations in many-objective optimization: 1) there is an aggravating conflict between proximity and diversity as the dimensions of the objective space increase, and 2) the Pareto dominance relation is not effective in solving many-objective optimization problems. In BiGE, two indicators are used to estimate the proximity and crowding degree of solutions in the population, respectively. By doing so, BiGE maps solutions from the original objective space to a bi-goal space and deals with the two goals by the nondominated sorting. It is able to provide sufficient selection pressure towards the Pareto front, regardless of the number of objectives that the optimization problem has. Despite its attractive features, BiGE may not be able to handle dominance resistant solutions (DRSs) during the search process. DRSs are far away from the main population and always ranked as good solutions by BiGE, thus hindering the evolutionary progress of the population. To address this issue, this chapter proposes an angle-based crowding degree estimation method for BiGE (denoted as aBiGE).

4.2 The Proposed Algorithm: aBiGE

4.2.1 A Brief Review of BiGE

Algorithm 4.1 shows the basic framework of BiGE. First, N individuals are randomly generated to initialize a population. Then, the proximity and crowding degree estimation are performed for all individuals in the population. Next, the mating selection is implemented to select the promising individuals from the current population based on the proximity and crowding degree values. Afterward, variation operators (e.g., crossover and mutation) are employed to those selected individuals (i.e., the parents) to produce new individuals (i.e., the offspring). Finally, the environmental selection is performed to reduce the expanded population of parents and offspring to N individuals as the new population for next generation.

Algorithm 4.1 Basic Framework of BiGE [96]

```
Require: $P$ (current population), $N$ (population size)
 1: $P \leftarrow Initialization(P)$
 2: while termination criterion not fulfilled do
 3:     $Proximity\_Estimation(P)$
 4:     $Crowding\_Degree\_Estimation(P)$
 5:     $P' \leftarrow Mating\_Selection(P)$
 6:     $P'' \leftarrow Variation(P')$
 7:     $Q \leftarrow P' \bigcup P''$
 8:     $P \leftarrow Environmental\_Selection(Q)$
 9: end while
10: return $P$
```

In particular, a simple aggregation function is adopted to estimate the proximity of an individual. For an individual x in a population, its aggregation value (denoted as $f_p(\mathbf{x})$) is calculated by the sum of each normalized objective value in the range [0, 1] (line 3 in Algorithm 4.1), formulated as [96]:

$$f_p(\mathbf{x}) = \sum_{j=1}^{M} \widetilde{f}_j(\mathbf{x}). \tag{4.1}$$

where $\widetilde{f}_j(\mathbf{x})$ denotes the normalized objective value of individual x in the jth objective, and M represents the number of objectives. A smaller f_p value of an individual usually indicates better performance on proximity. In particular, for a DRS, it is more likely to obtain a significantly large f_p value in comparison with other individuals in a population.

In addition, the crowding degree of an individual $\mathbf{x}$ in a population P (line 4 in

Algorithm 4.1) is defined as follows [96]:

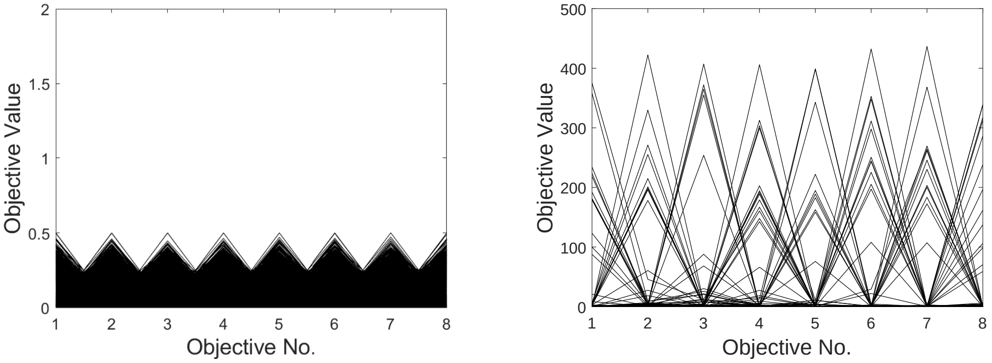
$$f_c(\mathbf{x}) = (\sum_{\mathbf{y} \in P, \mathbf{x} \neq \mathbf{y}} sh(\mathbf{x}, \mathbf{y}))^{1/2}. \tag{4.2}$$

where $sh(\mathbf{x}, \mathbf{y})$ stands for a sharing function between a pair of individuals $\mathbf{x}$ and $\mathbf{y}$ in the objective space:

$$sh(\mathbf{x}, \mathbf{y}) = \begin{cases} (0.5(1 - \frac{d(\mathbf{x},\mathbf{y})}{r}))^2, & \text{if } d(\mathbf{x}, \mathbf{y}) < r, f_p(\mathbf{x}) < f_p(\mathbf{y}) \\ (1.5(1 - \frac{d(\mathbf{x},\mathbf{y})}{r}))^2, & \text{if } d(\mathbf{x}, \mathbf{y}) < r, f_p(\mathbf{x}) > f_p(\mathbf{y}) \\ rand(), & \text{if } d(\mathbf{x}, \mathbf{y}) < r, f_p(\mathbf{x}) = f_p(\mathbf{y}) \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

where r represents the radius of a niche, adaptively calculated by $r = 1/\sqrt[M]{N}$ (N is the population size and M is the number of objectives), and $rand()$ denotes a function that randomly assigns *either* $sh(\mathbf{x}, \mathbf{y}) = (0.5(1 - d(\mathbf{x}, \mathbf{y})/r))^2$ and $sh(\mathbf{y}, \mathbf{x}) = (1.5(1 - d(\mathbf{x}, \mathbf{y})/r))^2$ *or* $sh(\mathbf{x}, \mathbf{y}) = (1.5(1 - d(\mathbf{x}, \mathbf{y})/r))^2$ and $sh(\mathbf{y}, \mathbf{x}) = (0.5(1 - d(\mathbf{x}, \mathbf{y})/r))^2$. Note that the lower the f_c value (i.e., crowding degree) of the individual, the better the diversity performance.

It is observed that BiGE tends to struggle on a class of many-objective optimization problems where the search process involves DRSs, such as the DTLZ1 and DTLZ3 test functions (in a well-known benchmark test suite DTLZ [38]). Figure 4.1 shows the true Pareto front of the eight-objective DTLZ1 and the final solution set of BiGE in one typical run on the eight-objective DTLZ1 by parallel coordinates. The parallel coordinates plot is a commonly used visualization tool in many-objective optimization, which maps a set of solutions in the high-dimensional objective space onto a 2D parallel coordinates plane. Particularly, Li et al. in [101] systematically explained how to read many-objective solution sets in the parallel coordinates, and indicated that the parallel coordinates can *partly* reflect the quality of a many-objective solution set in terms of convergence, coverage, and uniformity.



(a) The true Pareto front

(b) The final solution set of BiGE

Figure 4.1: The true Pareto front of the eight-objective DTLZ1 and the final solution set of BiGE on the eight-objective DTLZ1, shown by parallel coordinates.

Clearly, there are some solutions that are far away from the Pareto front in BiGE, with the solution set of eight-objective DTLZ1 ranging from 0 to around 450 compared to the Pareto front ranging from 0 to 0.5 on each objective. Such solutions always have a poor proximity degree and a good crowding degree (estimated by Euclidean distance) in bi-objective space (in terms of convergence and diversity), and will be preferred since there is no solution in the population that dominates them in BiGE. These solutions are detrimental for BiGE to converge the population to the Pareto front considering their poor performance regarding convergence. A straightforward method to remove DRSs is to change the crowding degree estimation method.

4.2.2 Basic Idea

The basic idea of the proposed method is based on some observations of DRSs. Figure 4.2 shows one typical situation of a non-dominated set with five individuals including two DRSs (i.e, **A** and **E**) in a bi-objective minimization scenario.

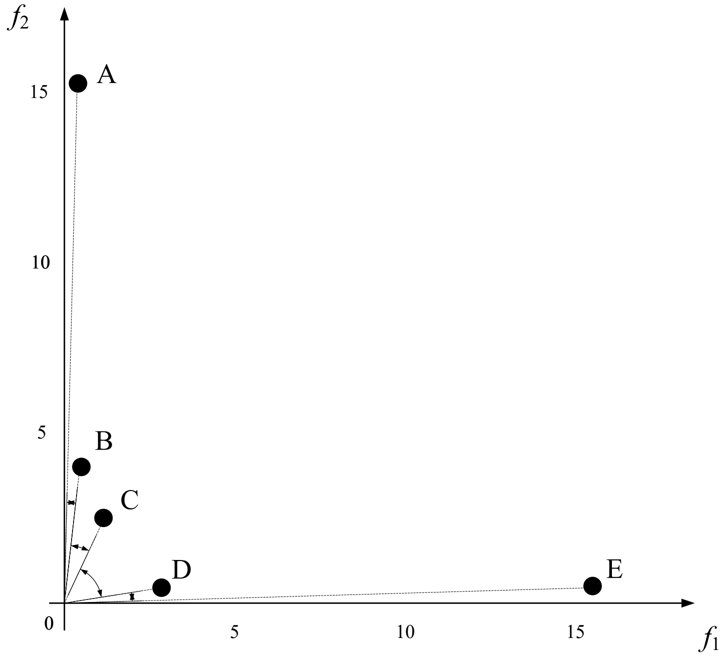


Figure 4.2: An example that DRSs (**A** and **E**) obtain good crowding degrees if estimated by the Euclidean distance but poor crowding degrees if calculated by the vector angle between two neighbors.

As seen from Figure 4.2, it is difficult to find a solution that dominates DRSs when estimating the crowding degree of the solutions based on Euclidean distance. Take individual **A** as an example, it performs well on objective f_1 (slightly better than **B** with a near-optimal value) but inferior to all the other solutions on objective f_2. It is difficult to find a solution with a better value than **A** on objective f_1, same as individual **E** on objective f_2. **A** and **E** (with poor proximity degree and good crowding degree) are considered as good solutions and have a high possibility to survive in the next generation in BiGE. However, the results would be different if the crowding degree estimation based on distance is replaced by that based on vector angles. Specifically, it can be observed that 1) an individual in a crowded area (e.g., **B**) would have a smaller vector angle to its neighbor compared to the individual in a sparse area (e.g., **C**), 2) a DRS would have an extremely small value of vector angle to its neighbor, e.g., the angle between **A** and **B** or the angle between **E** and **D**. Namely, these DRSs would have both poor proximity and crowding degree, and have a high possibility to be eliminated during the evolutionary process. Overall, vector angles have the advantage of distinguishing DRSs from other solutions in a population and could be adopted in the crowding degree estimation.

4.2.3 Angle-based Crowding Degree Estimation

Inspired by the work in [147], a novel angle-based crowding degree estimation method is proposed and integrated into the BiGE framework (line 4 in Algorithm 4.1), called aBiGE. Before estimating the crowding degree of an individual, some basic definitions are first introduced.

Definition 4.2.1 (Norm)**.** *For an individual* $\mathbf{x}$ *in a population (i.e.,* $\mathbf{x} \in P$*), its norm, termed* $norm(\mathbf{x})$ *in the normalized objective space, is calculated by the following equation [147]:*

$$norm(\mathbf{x}) = \sqrt{\sum_{j=1}^{M} \widetilde{f}_j(\mathbf{x})^2}. \tag{4.4}$$

Definition 4.2.2 (Vector Angles)**.** *The vector angle between any two individuals* $\mathbf{x}$ *and* $\mathbf{y}$ *in a population* P *is computed as [147]:*

$$angle_{\mathbf{x}\rightarrow\mathbf{y}} = arccos\left|\frac{F'(\mathbf{x}) \bullet F'(\mathbf{y})}{norm(\mathbf{x}) \cdot norm(\mathbf{y})}\right|. \tag{4.5}$$

where $F'(\mathbf{x}) \bullet F'(\mathbf{y})$ *is the inner product between* $F'(\mathbf{x})$ *and* $F'(\mathbf{y})$ *defined as:*

$$F'(\mathbf{x}) \bullet F'(\mathbf{y}) = \sum_{j=1}^{M} \widetilde{f}_j(\mathbf{x}) \cdot \widetilde{f}_j(\mathbf{y}). \tag{4.6}$$

It is worth mentioning that $angle_{\mathbf{x}\rightarrow\mathbf{y}} \in [0, \pi/2]$*.*

Then, each individual $\mathbf{x} \in P$ is assigned a vector angle value (θ), which is the minimum vector angle value between $\mathbf{x}$ and another individual in population P, namely,

$$\theta(\mathbf{x}) = \Big\{ \min_{\mathbf{y} \in P \setminus \{\mathbf{x}\}} angle_{\mathbf{x}\rightarrow\mathbf{y}} \mid \mathbf{x} \in P \Big\}. \tag{4.7}$$

In the proposed angle-based crowding degree estimation, once an individual $\mathbf{x}$ is selected, its $\theta(\mathbf{x})$ value will be punished in order to strike a balance between diversity and proximity. Specifically, some important factors are considered:

- The individuals that have more adjacent individuals in a niche should be penalized. In aBiGE, a punishment to an individual $\mathbf{x}$ is on the basis of the number of individuals in the niche having a lower (better) proximity degree than the individual $\mathbf{x}$ (denote as c). The punishment is aggravated with an increase of c.
- To avoid a situation where some individuals have the same θ value, individuals should be further punished. In aBiGE, another penalty is performed according to the proportion of the $\theta(\mathbf{x})$ to the sum of the θ values of all individuals in the niche, denoted as p.

Keep the above factors in mind, in aBiGE, the crowding degree (denoted as f_a) of an individual $\mathbf{x}$ in a population is defined as follows:

$$f_a(\mathbf{x}) = \frac{c+1}{\theta(\mathbf{x}) \cdot (p+1) + \dfrac{\pi}{90}}. \tag{4.8}$$

The angle-based crowding degree estimation in aBiGE aims to enhance the selection pressure on those non-dominated solutions in the population of each generation and avoid the negative influence of DRSs during the optimization process. It is worth mentioning that a smaller value of f_a is preferred.

4.3 Experimental Results

4.3.1 Experimental Design

To test the performance of the proposed aBiGE on those many-objective optimization problems where the search process involves DRSs, the experiments were conducted on nine DTLZ test instances. For each test problem (i.e., DTLZ1, DTLZ3, and DTLZ7), five, eight, and ten objectives were considered, respectively.

To make a fair comparison with the state-of-the-art BiGE for MaOPs, settings for both BiGE and aBiGE are given as follows:

- The size of population for both algorithms were set to 100 for all test problems.
- Each algorithm on each test problem was executed 30 independent runs to reduce the impact of its randomness.
- The stopping condition of a run was a predefined maximum number of $30,000$ evaluations.
- For crossover and mutation operators, the simulated binary crossover [32] and polynomial mutation [33] were adopted, with both distribution indexes set to 20. The crossover and mutation probability were set to 1.0 and $1/n$ (n is the number of decision variables), respectively.

Li et al. [96] investigated the effect of the population size on the performance of six algorithms. In [96], the performance of BiGE remains effective under the scenarios with different sizes of the population. Other parameters were kept unchanged, including the number of generations, and etc. [96]. In this chapter, the settings of the population size and the stopping criterion of both algorithms in the experiments are the same as in [96, 152]. For crossover and mutation operators in the experiments, the crossover probability, the distribution index of crossover, the mutation probability, and the distribution index of mutation were set based on the conventions [96, 152, 124, 154].

Algorithm performance was assessed by the IGD$^+$ [70] indicator. IGD$^+$ is a modified version of the inverted generational distance indicator (IGD) [26] (see more details in Chapter 2.2.3). It has been shown that IGD is not compatible with Pareto dominance [166, 70]; consequently, DRSs in the solution set could largely affect the evaluation results. On the other hand, IGD$^+$ can well reflect the quality of a solution set in terms of both convergence and diversity given the problem's Pareto front is available, and overcome the problem from DRSs when these solutions are included in the reference set [99]. Therefore, IGD$^+$ is well-suited for assessing the performance of the evolutionary approaches in our comparative study.

4.3.2 Performance Comparison

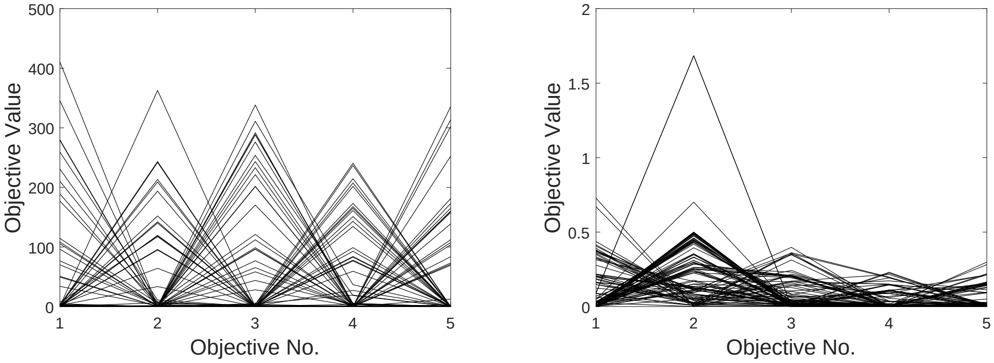
Test Problems with DRSs

Table 4.1 shows the mean and standard deviation of IGD$^+$ metric results on nine DTLZ test instances with DRSs. For each test problem, among different algorithms, the algorithm that has the best result based on the IGD$^+$ metric is shown in bold. As can be seen from the table, for many-objective optimization problems with DRSs, the proposed aBiGE performs significantly better than BiGE on all test instances in terms of convergence and diversity.

Table 4.1: Mean and standard deviation of the IGD$^+$ metric on nine DTLZ test instances. The best result for each test instance is highlighted in boldface.

Problem	Obj.	BiGE		aBiGE	
		Mean	SD	Mean	SD
DTLZ1	5	8.4207E-01	3.59E-01	**1.1768E-01**	3.41E-02
	8	1.9350E+00	1.27E+00	**1.9495E-01**	9.44E-02
	10	1.9653E+00	1.36E+00	**2.2763E-01**	9.57E-02
DTLZ3	5	1.5705E+01	5.87E+00	**6.0008E+00**	3.50E+00
	8	3.3434E+01	1.17E+01	**9.6401E+00**	6.30E+00
	10	3.5720E+01	1.58E+01	**1.2780E+01**	5.40E+00
DTLZ7	5	4.6666E-01	1.52E-01	**3.1701E-01**	6.48E-02
	8	3.0415E+00	6.03E-01	**2.6350E+00**	8.59E-01
	10	5.6152E+00	7.41E-01	**4.0059E+00**	4.53E-01

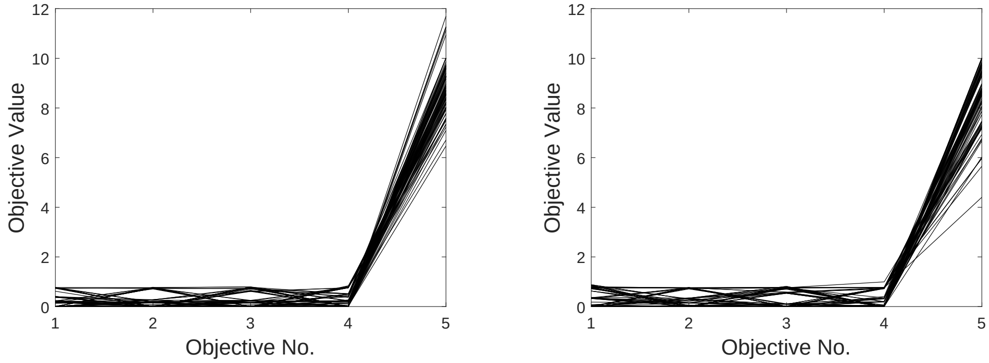
For visual observations, Figures 4.3 and 4.4 show, by parallel coordinates, the solution sets of one run of the two algorithms on the five-objective DTLZ1 and the five-objective DTLZ7. This run corresponds to the solution set with the closest result to the average value of IGD$^+$. As shown in Figure 4.3 (a), the solution set obtained by BiGE has a poor convergence on the five-objective DTLZ1, with the range of its solution set from 0 to 400 in contrast to the Pareto front ranging from 0 to 0.5 on each objective. From Figure 4.3 (b), it can be observed that the proposed aBiGE could converge its solution set to the Pareto front except for a few solutions.



(a) BiGE

(b) aBiGE

Figure 4.3: The final solution sets obtained by BiGE and aBiGE on the five-objective DTLZ1, shown by parallel coordinates.



(a) BiGE

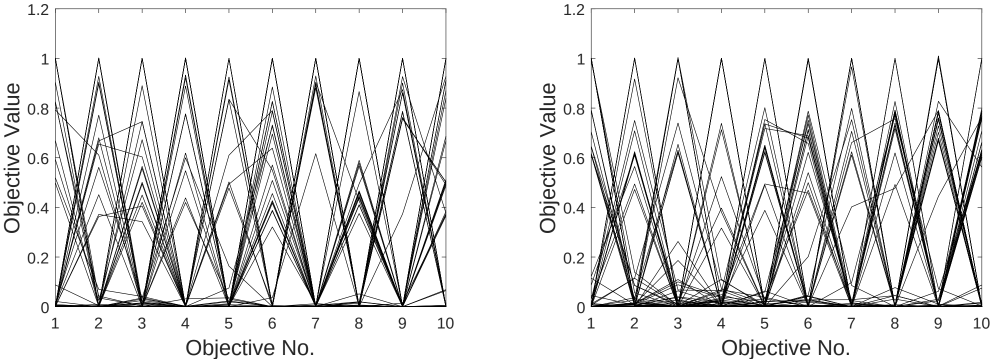
(b) aBiGE

Figure 4.4: The final solution sets obtained by BiGE and aBiGE on the five-objective DTLZ7, shown by parallel coordinates.

For the solutions of the five-objective DTLZ7, the boundary of the first four objectives is in the range [0, 1], and the boundary of the last objective is in the range $[3.49, 10]$ according to the formula of DTLZ7. As can be seen from Figure 4.4, all solutions of the proposed aBiGE appear to converge into the Pareto front. In contrast, some solutions (with their objective values beyond the upper boundary in the 5th objective) of BiGE fail to reach the range of the Pareto front. In addition, the solution set of the proposed aBiGE has better extensity than BiGE on the boundaries. In particular, the solution set of BiGE fail to cover the region from 3.49 to 6 of the last objective, while the solution set of the proposed aBiGE do not cover the range of the Pareto front below 4 on the 5th objective.

Test Problem without DRSs

Figure 4.5 gives the final solution sets (with the closest results to the average values of IGD$^+$ among 30 runs) of both algorithms on the ten-objective DTLZ2 in order to visualize their distribution on the many-objective optimization problems without DRSs. As can be seen, the final solution sets of both algorithms could reach the range of the Pareto front (with lower and upper boundary within [0, 1] on each objective), i.e., good convergence. Moreover, refer to [101], parallel coordinates in Figure 4.5 partly reflect that the diversity of the solutions obtained by aBiGE is sightly worse than that obtained by BiGE. This observation can be assessed by the IGD$^+$ performance indicator where BiGE obtains a slightly lower (better) IGD$^+$ value than the proposed aBiGE.



(a) BiGE

(b) aBiGE

Figure 4.5: The final solution sets of BiGE and aBiGE on the ten-objective DTLZ2, shown by parallel coordinates, and their evaluation results by using the IGD$^+$ indicator (the lower value indicates better performance). (a) BiGE (IGD$^+$ = 2.4319E-01) (b) aBiGE (IGD$^+$ = 2.5021E-01).

Overall, aBiGE algorithm performs generally better than the original BiGE algorithm on test problems with DRSs in terms of convergence and diversity, even for the problem instances with ten objectives. It sometimes performs slightly worse than BiGE on test problems without DRSs in terms of diversity. One possible explanation for this is that the crowding degree estimation in aBiGE can reduce the adverse impact of DRSs during the search process, thereby achieving a good balance between convergence and diversity in many-objective optimization. As such, aBiGE is suitable for those problems where the search process involves DRSs. Despite the advantages, for those problems without DRSs, the crowding degree estimation in aBiGE may be inaccurate since it only considers the distribution of individuals in a niche, while the crowding degree estimation in aBiGE considers the distribution of individuals within or outside a niche.

4.4 Summary

In this chapter, an issue of a well-established many-objective evolutionary algorithm BiGE on the challenges in handling dominance resistant solutions during the search process has been addressed. An angle-based crowding distance estimation method has been proposed to replace the distance-based crowding distance estimation method in BiGE, thus significantly reducing the effect of dominance resistant solutions to the algorithm.

The effectiveness of the proposed method has been well evaluated on three representative problems with dominance resistant solutions. It is nevertheless worth noting that for problems without dominance resistant solutions the proposed method performs slightly worse than the original BiGE. After addressing the issue of effectiveness faced by a well-established many-objective evolutionary algorithm on more generic cases, efficiency is another important issue in evolutionary many-objective optimization to tackle. The next chapter addresses the issue of the balance between effectiveness and efficiency in many-objective optimization by developing a new evolutionary algorithm.

Chapter 5

Balancing Effectiveness and Efficiency in Evolutionary Many-Objective Optimization

In the past two chapters, the effectiveness of many-objective evolutionary algorithms has been addressed by showing how it can be achieved on a challenging real-world problem as well as how this type of algorithms may be further enhanced in general. However, some effective algorithms may become very computationally expensive with the increase of the number of objectives, e.g., certain indicator-based algorithms. In this chapter, both effectiveness and efficiency are addressed by developing a novel many-objective evolutionary algorithm that is able to handle various many-objective problems without any additional parameter. This chapter has conducted extensive experiments and found that the proposed algorithm outperforms 11 state-of-the-art many-objective evolutionary algorithms on the majority of the tested problems.

The remainder of this chapter is organized as follows. In Section 5.1, the introduction of this work is given. Section 5.2 describes the framework and detailed information of the proposed algorithm. The experimental results are presented in Section 5.3. Section 5.4 summarizes this chapter.

5.1 Introduction

In the context of evolutionary multi-/many-objective optimization, effectiveness refers to how a search algorithm performs in terms of converging its solutions into the Pareto front and also diversifying them over the front. Efficiency refers to how quickly a search algorithm is executed. Achieving both high effectiveness and efficiency is not easy, particularly on practical applications where the problem's Pareto front may be highly complex and unpredictable.

Convergence is especially challenging in many-objective optimization. Pareto-based algorithms in evolutionary multiobjective optimization (EMO), e.g., NSGA-II [37] and SPEA2 [163] often fail to scale up well in objective dimensionality. Moreover, recent studies [93] suggest that well-established decomposition-based [156, 34], and indicator-based [162, 9] algorithms may also struggle to converge their solutions even when the objective dimension is as low as four.

Maintaining a well-distributed solution set is another important issue in many-objective optimization. Improving convergence may come at a cost of compromising the diversity. For example, new dominance relations (e.g., *ϵ-dominance* [84] and *fuzzy Pareto dominance* [61]), which were designed for promoting the convergence, may lead the population into one (or several) sub-area of the Pareto front [28, 106]. Decomposition-based algorithms, which perform very well in terms of convergence, typically face challenges of diversifying their solutions over the Pareto front for problems with irregular Pareto front shapes [71]. In addition, indicator-based algorithms tend to favor a certain region of the Pareto front, such as IBEA [162] for the boundary solutions of the Pareto front [94] and SMS-EMOA [9] for the knee point(s) [97]. In contrast, some recent many-objective evolutionary algorithms may miss a certain region of the Pareto front, such as SPEA2+SDE [95] unable to well maintain boundary solutions in some problems [86].

On the other hand, the efficiency of some EMO algorithms decreases dramatically with the number of objectives. For example, the time requirement of algorithms based on the hypervolume indicator [165], such as SMS-EMOA [9] and MO-CMA-ES [67], increases exponentially in objective dimensionality. For another example, SPEA2+SDE, which has been shown effective in recent studies [93, 91], also suffers from poor efficiency as the computational complexity is $O(MN^3)$, where N denotes the size of the population.

Lastly, many multi-/many-objective evolutionary algorithms need some extra parameters which should be set properly and individually for different problems. For example, for algorithms based on the modified Pareto dominance relation (e.g., ϵ-MOEA [36]), it is crucial to specify the relax degree of Pareto dominance (i.e., the area that a solution dominates) for a problem with a large number of objectives [85, 98]. For another example, in some region-based MaOEAs it is important to find the right parameters to determine the size of the region, such as the grid division parameter in GrEA [152] and the neighborhood size in KnEA [157]. However, the *sweet-spot* of such parameters may significantly shrink with the number of objectives [114]. It is difficult or even impossible to find their best settings in the high-dimensional objective space [98].

Given the above, this chapter aims to develop a novel balancing effectiveness and efficiency (called BEE) algorithm to solve many-objective optimization problems without any extra parameter. Specifically, the proposed algorithm has the following desirable features:

- effectiveness in the sense of converging its solutions into the Pareto front and also diversifying them on the front;
- efficiency in the sense of a reasonable amount of execution time;
- suitability for MaOPs with various Pareto front shapes;
- no additional parameter except those associated with an evolutionary algorithm (e.g., population size and crossover rate).

The proposed algorithm BEE is characterized by selecting boundary solutions as well as non-boundary solutions. The former is to determine the range of the estimated Pareto front, from which the latter is to maintain a set of well-distributed and well-converged solutions in the high-dimensional space.

5.2 The Proposed Algorithm

5.2.1 Framework of BEE

The framework of the proposed BEE is described in Algorithm 5.1. The basic procedure follows many MaOEAs. First, a population P containing N solutions is randomly generated. Next, each solution in P is assigned a fitness value according to its nondominated rank. Then, mating selection is applied to select promising solutions regarding their nondominated ranks, followed by variation operations to generate an offspring population. Finally, the environmental selection procedure is performed to preserve the N best solutions for next generation. Like most of existing MaOEAs (e.g., NSGA-III [34]), the proposed algorithm focuses on the design of environmental selection, an operation which plays a key role in algorithm performance in many-objective optimization.

Algorithm 5.1 Framework of BEE

```
Require: $P$ (population), $N$ (population size)
$P \leftarrow Initialize\_population(N)$
$Front \leftarrow Pareto\_nondominated\_sort(P)$;
while the termination criterion is not met do
    $P' \leftarrow Mating\_selection(P, Front)$
    $P'' \leftarrow Variation(P')$
    $P \leftarrow Environmental\_selection(P \cup P'')$
end while
return $P$
```

5.2.2 Environmental Selection

Main Procedure

Environmental selection is an evolutionary operation to determine the survival of solutions (i.e., next-generation population) from the current population and offspring. Algorithm 5.2 shows the main procedure of the environmental selection in BEE. First, the combined set of the current population and their offspring population is sorted into different fronts $(F_1, F_2, \dots)$ using the nondominated sorting method [37] (line 4). Then, fronts are selected one by one to construct a new population P, starting from F_1, until the critical front F_i is found, where $|F_1 \cup F_2 \cup \cdots \cup F_{i-1}| < N$ and $|F_1 \cup F_2 \cup \cdots \cup F_{i-1} \cup F_i| \geq N$ (N denotes the population size) (lines 5–8). In fact, for MaOPs, the solutions are typically Pareto nondominated to each other and consequently the critical front is usually the first front, namely i is equal to 1. For size $|P| + |F_i| > N$, function $Findout\_best$ in line 12 is designed to select promising $N - |P|$ solutions from the critical front F_i. For simplicity, "candidate set" is used to represent all the solutions in the critical front. The function $Findout\_best$ can break down into two operations: selecting boundary solutions and selecting non-boundary solutions as shown in Algorithm 5.3. In the next two sections, the two operations will be introduced respectively.

Algorithm 5.2 $Environmental\_selection(P)$

```
Require: $P$ (combination of the current population and the newly produced solutions)
 1: $Q \leftarrow \emptyset$
 2: $i \leftarrow 1$
 3: $(F_1, F_2, \dots) \leftarrow Pareto\_nondominated\_sort(P)$
 4: while $|Q| + |F_i| < N$ do
 5:     $Q \leftarrow Q \cup F_i$
 6:     $i \leftarrow i + 1$
 7: end while
 8: if $|Q| = N$ then
 9:     return $Q$
10: else
11:     $K \leftarrow N - |Q|$
12:     $P \leftarrow Findout\_best(F_i, K)$    /* Select $K$ solutions from $F_i$ for the next-generation
        evolution */
13:     return $Q$
14: end if
```

Algorithm 5.3 $Findout\_best(P, K)$

```
Require: $P$ (candidate set), $K$ (number of solutions to be selected from the candidate set)
 1: $Q \leftarrow Boundary\_solution\_selection(P)$
    /* Select boundary solutions from $P$ */
 2: $Q \leftarrow Nonboundary\_solution\_selection(P, Q, K)$
    /* Select promising non-boundary solutions from $P$ */
 3: return $Q$
```

Selecting Boundary Solutions

Given a candidate set P, M (M denotes the number of objectives) boundary solutions are first selected from P and placed into a set Q (which stores the solutions for the next-generation evolution). Specifically, boundary solution b_j corresponds to the one in P that minimizes a modified version of Tchebycheff function $agg(\mathbf{x}, \boldsymbol{w}_j)$, as suggested and practiced in the recent studies [34, 92]:

$$b_j = arg \min_{\mathbf{x} \in P} agg(\mathbf{x}, \boldsymbol{w}_j) \tag{5.1}$$

$$agg(\mathbf{x}, \boldsymbol{w}_j) = \max_{i=1}^{M} \left\{ \frac{1}{\boldsymbol{w}_{j,i}} |f_i(\mathbf{x}) - z_i^{min}| \right\} \tag{5.2}$$

where P stands for the candidate set, and $\boldsymbol{w}_j = (w_{j,1}, w_{j,2}, \ldots, w_{j,M})^T$ denotes a weight vector close to the jth objective axis direction:

$$\boldsymbol{w}_{j,i} = \begin{cases} 1, & i = j \\ 10^{-6}, & i \neq j \end{cases} \tag{5.3}$$

In the case that multiple weight vectors share the same boundary solutions, duplicate boundary solutions are deleted. Algorithm 5.4 gives the procedure of selecting boundary solutions.

Algorithm 5.4 *Boundary_solution_selection*(P)

```
Require: $P$ (candidate set), $M$ (number of objectives)
1: $Q \leftarrow \emptyset$
2: for $j \leftarrow 1$ to $M$ do
3:     $b_j \leftarrow arg\ \min_{\mathbf{x} \in P} agg(\mathbf{x}, \boldsymbol{w}_j)$
4: end for
5: $Q \leftarrow \{b_1, b_2, \ldots, b_M\}$
6: $Q \leftarrow unique(Q)$;
   /* Delete duplicate boundary solutions */
7: $Normalization(P, Q)$
   /* Objective normalization */
8: return $Q$
```

In Algorithm 5.4, function *Normalization* in line 7 is used to normalize each objective of solutions in the candidate set for the preprocessing of selecting non-boundary solutions as described in the next section. Here, the normalization operation in NSGA-III [34] is employed. Specifically, for each solution $\mathbf{x}$ in the candidate set, its ith objective value $f_i(\mathbf{x})$ is normalized as follows:

$$\widetilde{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - z_i^{min}}{a_i} \tag{5.4}$$

where z_i^{min} denotes the minimum value of the ith objective for all solutions in the candidate set. The denominator a_i represents the intercept of the $(M-1)$-dimensional linear hyperplane with the objective axis f_i. The $(M-1)$-dimensional linear hyperplane is constructed by M boundary solutions b_1, b_2, ..., b_M of the candidate set. In the case that the number of boundary solutions is less than M, a_i is assigned the maximum value of the ith objective for all solutions in the candidate set. Algorithm 5.5 gives the normalization procedure.

Algorithm 5.5 $Normalization(P, Q)$

```
Require: $P$ (candidate set), $Q$ (selected solutions), $M$ (number of objectives)
for $i \leftarrow 1$ to $M$ do
    $z_i^{min} \leftarrow \min_{\mathbf{x} \in P} f_i(\mathbf{x})$
    $\bar{f}_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) - z_i^{min}$    /* Translate the objectives */
end for
$(a_1, a_2, \ldots, a_M) \leftarrow Compute\_intercept(\bar{f}, Q)$
for each solution $\mathbf{x} \in P$ do
    for $i \leftarrow 1$ to $M$ do
        $\widetilde{f}_i(\mathbf{x}) \leftarrow \bar{f}_i(\mathbf{x})/a_i$
    end for
end for
```

Selecting Non-boundary Solutions

After the boundary solutions are selected, now BEE selects solutions far away from them (thus good diversity), but at the same time takes into account how these solutions perform in terms of their closeness to the Pareto front, relative to other solutions. That is, BEE selects solutions that have a good balance between diversity and convergence, in which the diversity is measured based on the solutions BEE has already selected while the convergence is concerned with the comparison between the unselected solutions. To do so, a step-by-step selection approach is proposed.

First, a metric (sd) is defined to reflect the quality of the unselected solutions in terms of both convergence and diversity, relative to the selected solutions:

$$sd(\mathbf{x}, Q) = \min_{\mathbf{y} \in Q} \; dist(\mathbf{x}, \mathbf{y}') \tag{5.5}$$

where Q denotes the set of solutions which are already selected for the next-generation evolution, $dist(\mathbf{x}, \mathbf{y}')$ represents the normalized Euclidean distance between solution $\mathbf{x}$ and $\mathbf{y}'$, $\mathbf{x} = (\widetilde{f}_1(\mathbf{x}),\ \widetilde{f}_2(\mathbf{x}),\ \ldots,\ \widetilde{f}_M(\mathbf{x}))$ is an unselected solution in the candidate set, and $\mathbf{y}' = (\widetilde{f}_1(\mathbf{y}'),\ \widetilde{f}_2(\mathbf{y}'),\ \ldots,\ \widetilde{f}_M(\mathbf{y}'))$ is the shifted version of selected solution $\mathbf{y} = (\widetilde{f}_1(\mathbf{y}),\ \widetilde{f}_2(\mathbf{y}),\ \ldots,\ \widetilde{f}_M(\mathbf{y}))$, which is defined as follows:

$$\widetilde{f}_i(\mathbf{y}') = \begin{cases} \widetilde{f}_i(\mathbf{x}), & \text{if } \widetilde{f}_i(\mathbf{y}) < \widetilde{f}_i(\mathbf{x}) \\ \widetilde{f}_i(\mathbf{y}), & \text{otherwise} \end{cases} \tag{5.6}$$

where $\widetilde{f}_i(\mathbf{x})$, $\widetilde{f}_i(\mathbf{y})$, and $\widetilde{f}_i(\mathbf{y}')$ stand for the ith normalized objective value of individuals $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{y}'$, respectively. Based on $\mathbf{y}'$, the normalized Euclidean distance between $\mathbf{x}$ and $\mathbf{y}'$, namely $dist$, is calculated:

$$dist(\mathbf{x}, \mathbf{y}') = \sqrt{\sum_{i=1}^{M} (\widetilde{f}_i(\mathbf{x}) - \widetilde{f}_i(\mathbf{y}'))^2} \tag{5.7}$$

where M denotes the number of objectives.

Second, some promising non-boundary solutions are selected. Specifically, the solution with the maximum sd value among unselected solutions (i.e., $\mathbf{s} = arg \max_{\mathbf{x} \in P \setminus Q}$ $sd(\mathbf{x},\ Q)$) is selected and placed into Q. The above steps are repeated until the size of selected solutions reaches the population size.

It should be noted that once a solution $\mathbf{s}$ is selected, the sd value of each remaining solution $\mathbf{x}$ in the candidate set is updated according to $dist(\mathbf{x}, \mathbf{s}')$ (i.e., $sd(\mathbf{x},\ Q) = \min\{dist(\mathbf{x},\ \mathbf{s}'),\ sd(\mathbf{x},\ Q)\}$), where $\mathbf{s}'$ represents the shifted version of the new selected solution $\mathbf{s}$ according to equation (5.6). Algorithm 5.6 gives the procedure of selecting non-boundary solutions.

Algorithm 5.6 $Nonboundary\_solution\_selection(P, Q, K)$

```
Require: $P$ (candidate set), $Q$ (selected solutions), $K$ (number of solutions to be selected from the candidate set)
$P \leftarrow P \setminus Q$
for $\mathbf{x} \in P$ do
    $sd(\mathbf{x}, Q) \leftarrow \min_{\mathbf{y} \in Q} dist(\mathbf{x}, \mathbf{y})$    /* Initialize the $sd$ values of all unselected solutions in the candidate set */
end for
while $|Q| < K$ do
    $\mathbf{s} \leftarrow arg \max_{\mathbf{x} \in P} sd(\mathbf{x}, Q)$    /* Select promising non-boundary solutions from the unselected solutions */
    $Q \leftarrow Q \cup \{\mathbf{s}\}$
    $P \leftarrow P \setminus \{\mathbf{s}\}$
    for $\mathbf{x} \in P$ do
        $sd(\mathbf{x}, Q) \leftarrow \min\{dist(\mathbf{x}, \mathbf{s}'), sd(\mathbf{x}, Q)\}$
    end for
end while
return $Q$
```

An Example of Environmental Selection

Consider a bi-objective minimization problem where a set of seven candidate solutions **A**–**G** to be selected. Figure 5.1 shows the environmental selection procedure. The actual objective values (f_1, f_2) and sd values of the non-boundary solutions in the candidate set are given in Table 5.1.

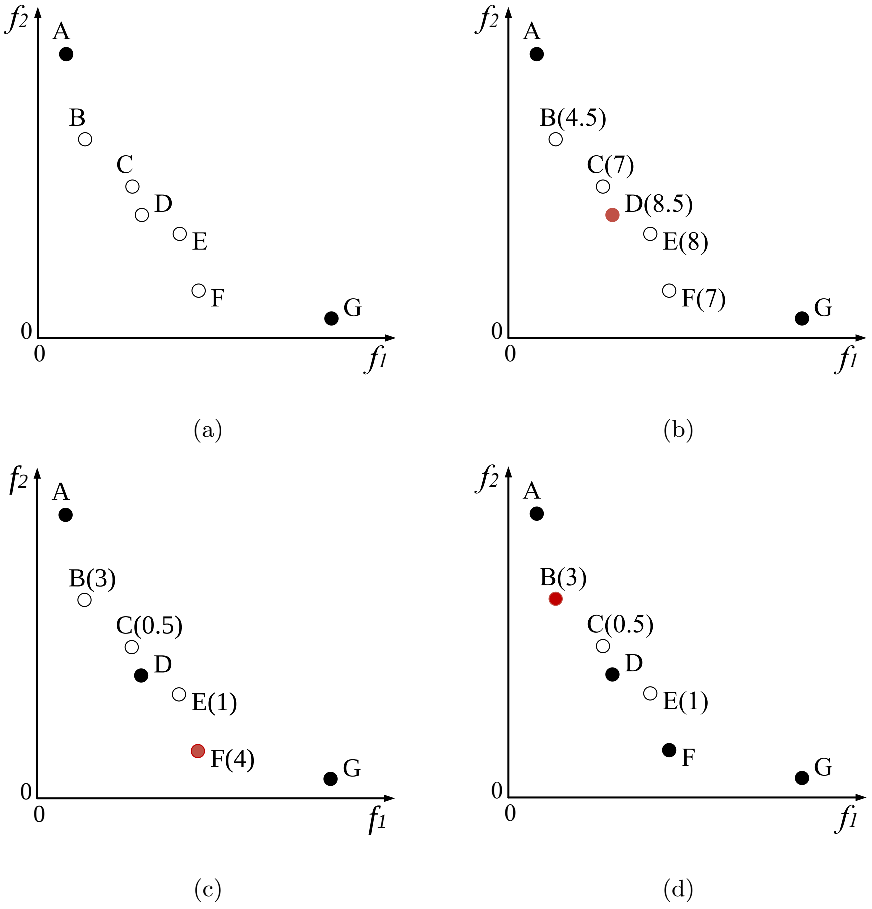


Figure 5.1: A bi-objective example of the environmental selection procedure, where the population size is set to five and the candidate set consists of seven solutions **A**–**G**. The number in parentheses associated with a non-boundary solution represents the *sd* value of that solution. (a) Selecting **A** and **G** (since they are two boundary solutions). (b) Selecting **D** (since it has the maximum *sd* value 8.5 among non-boundary solutions **B**–**F**). (c) Selecting **F** (since it has the maximum *sd* value 4 among non-boundary solutions **B**, **C**, **E**, **F**). (c) Selecting **B** (since it has the maximum *sd* value 3 among non-boundary solutions **B**, **C**, **E**).

Table 5.1: An illustration of how the *sd* value of each solution changes during selecting non-boundary solutions. The framed solutions mean that they have been selected. The population size is set to five.

(a)

	(f_1, f_2)	sd
A	(1.5, 15)	-
G	(15.5, 1)	-
B	(2.5, 10.5)	(4.5)
C	(5, 8)	(7)
D	(5.5, 6.5)	(8.5)
E	(7.5, 5.5)	(8)
F	(8.5, 2.5)	(7)

(b)

	(f_1, f_2)	sd
A	(1.5, 15)	-
G	(15.5, 1)	-
D	(5.5, 6.5)	-
B	(2.5, 10.5)	(3)
C	(5, 8)	(0.5)
E	(7.5, 5.5)	(1)
F	(8.5, 2.5)	(4)

(c)

	(f_1, f_2)	sd
A	(1.5, 15)	-
G	(15.5, 1)	-
D	(5.5, 6.5)	-
F	(8.5, 2.5)	-
B	(2.5, 10.5)	(3)
C	(5, 8)	(0.5)
E	(7.5, 5.5)	(1)

(d)

	(f_1, f_2)	sd
A	(1.5, 15)	-
G	(15.5, 1)	-
D	(5.5, 6.5)	-
F	(8.5, 2.5)	-
B	(2.5, 10.5)	-
C	(5, 8)	(0.5)
E	(7.5, 5.5)	(1)

First, **A** and **G** are selected (i.e., $Q = \{\mathbf{A}, \mathbf{G}\}$) since both **A** and **G** are boundary solutions identified by equations (5.1) and (5.2), as shown in Figure 5.1 (a). For the non-boundary solutions **B**–**F**, their *sd* values are calculated, as shown in Table 5.1 (a).

Second, the non-boundary solution **D** is selected (i.e., $Q = \{\mathbf{A},\ \mathbf{G},\ \mathbf{D}\}$) since it has the maximum *sd* value among non-boundary solutions **B**–**F** (Figure 5.1 (b)). Accordingly, the *sd* values of the unselected solutions **B**, **C**, **E**, **F** are updated (Table 5.1 (b)) according to lines 9–11 in Algorithm 5.6.

Third, the non-boundary solution **F** is chosen (i.e., $Q = \{\mathbf{A},\ \mathbf{G},\ \mathbf{D},\ \mathbf{F}\}$) since it has the maximum *sd* value among non-boundary solutions **B**, **C**, **E**, **F** (Figure 5.1 (c)). The *sd* values of unselected solutions **B**, **C**, **E** remain unchanged (Table 5.1 (c)) since the Euclidean distance between these solutions and the shifted version of **F**, namely *dist* values, are no less than their original *sd* values.

Finally, the non-boundary solution **B** is selected since it has the maximum *sd* value among non-boundary solutions **B**, **C**, **E** (Figure 5.1 (d)). The *sd* values of **C** and **E** remain unchanged (Table 5.1 (d)) since the Euclidean distance between these solutions and the shifted version of **B**, namely *dist* values, are no less than their original *sd* values. The final selected solutions are **A**, **G**, **D**, **F**, **B**.

In the environmental selection, its two operations, i.e., selecting boundary solutions and selecting non-boundary solutions work collaboratively. The former is to preserve solutions which perform best at least one objective. On the basis of these selected boundary solutions, the latter is to preserve well-balanced solutions by using the *sd* metric. A higher *sd* value indicates a better quality in terms of convergence and diversity. As can be seen in Figure 5.1, the best balanced solution, relative to the current selected solutions, is selected in turn; i.e., select **D** (relative to **A** and **G**), select **F** (relative to **A**, **G**, and **D**), and select **B** (relative to **A**, **G**, **D** and **F**).

5.2.3 Time Complexity

Considering a situation having M objectives and the evolutionary population consisting of N individuals. Here, $N \geq M$ is assumed and the worst case is considered where all solutions are nondominated with each other in the population. In one generation of BEE, nondominated sorting (line 3 in Algorithm 5.2) of the population requires $O(MN^2)$ comparisons. In the environmental selection, selecting boundary solutions (lines 2–4 in Algorithm 5.4) requires $O(M^2N)$ comparisons. Finding the minimum value of each objective (line 2 in Algorithm 5.5) requires $O(MN)$ comparisons. The translation of the objectives (line 3 in Algorithm 5.5) requires $O(MN)$ comparisons. Computing the intercepts (line 5 in Algorithm 5.5) requires $O(M^3)$ comparisons. The normalization of candidate solutions (lines 6–10 in Algorithm 5.5) requires $O(N)$ comparisons. Initializing the *sd* values of all unselected solutions in the candidate set (lines 2–4 in Algorithm 5.6) requires $O(M^2N)$ computations. Selecting promising non-boundary solutions from the unselected solutions using the quick sort (line 6 in Algorithm 5.6) requires $O(M^2N)$ computations. Updating the sd values for the unselected solutions (lines 7–12 in Algorithm 5.6) requires $O(MN^2)$ computations. In summary, by taking into account all the above operations, the overall time complexity of one generation of BEE in the worst case is $O(MN^2)$.

5.2.4 Comparison with SPEA2+SDE

Since both the proposed BEE algorithm and SPEA2+SDE [95] use the shift-based comparison between solutions in the selection procedure, this section will introduce their differences. In SPEA2+SDE, the solutions are shifted for the density estimation, thus the performance depends heavily on the accuracy of the density estimation operator of the original SPEA2. In contrast, BEE is a standalone algorithm which focuses on how to select a set of well-converged and diverse solutions. Furthermore, BEE preserves the boundary solutions explicitly, but SPEA2+SDE does not and may lose the boundary solutions of the Pareto front on some problems, as shown in [86]. In addition, the time complexity of BEE is $O(MN^2)$, lower than $O(MN^3)$ of SPEA2+SDE.

The above differences result in BEE to outperform SPEA2+SDE in terms of both performance and computational time, as will be seen in the next section.

5.3 Experimental Results

5.3.1 Experimental Design

To assess BEE, this chapter considers the following test suite, performance metric, state-of-the-art peer algorithms, and parameter settings, which are summarized in Table 5.2.

Table 5.2: Experimental setup

Factor	**Details**
Test problems	MaF1–15 [21]
Performance metric	IGD [26]
Number of objectives (M)	{3, 5, 10, 15}
Population size (N) for $M = \{3, 5, 10, 15\}$	{105, 126, 230, 240}
SBX [32] probability (p_c)	1.0
PM [33] probability (p_m)	$1/n$
Distribution index for SBX (η_c)	20
Distribution index for PM (η_m)	20
Number of runs	30
Maximum of generations ($MaxGen$)	300
Parameter values of RVEA	$\alpha = 2$, $f_r = 0.1$
Parameter values of MOEA/DD	$T = 0.1N$, $\delta = 0.9$, $\theta = 5$
Parameter values of 1by1EA	$k = 0.1N$, $R = 1$
Parameter values of MaOEA-CSS	$t = 0$
Parameter values of SRA	$p_c = 0.6$

Test Problems

The MaF test suite [21] is a continuous benchmark test suite proposed for the CEC'2017 competition on evolutionary many-objective optimization. In MaF, 15 test functions with diverse properties are selected or modified from existing test problems to well represent various real-world scenarios. In comparison with other test suites such as DTLZ [38] and WFG [66], MaF includes more problems with irregular Pareto front shapes, which poses bigger challenges for MaOEAs to approximate the whole Pareto front. In Table 5.3, the characteristics of all the MaF problems are summarized. The number of objectives was set to $M = 3, 5, 10, 15$. Their parameters were set according to [21].

Table 5.3: Characteristics of test problems in MaF.

Problem	Characteristics
MaF1	Linear, no single optimal solution in any subset of objectives
MaF2	Concave, no single optimal solution in any subset of objectives
MaF3	Convex, multimodal
MaF4	Concave, multimodal, badly scaled and no single optimal solution in any subset of objectives
MaF5	Convex, biased, Badly scaled
MaF6	Concave, degenerate
MaF7	Mixed, disconnected, multimodal
MaF8	Linear, degenerate
MaF9	Linear, degenerate, Pareto optimal solutions are similar to their image in the objective space
MaF10	Mixed, biased
MaF11	Convex, disconnected, nonseparable
MaF12	Concave, nonseparable, biased deceptive
MaF13	Concave, unimodal, nonseparable, degenerate, complex Pareto set
MaF14	Linear, partially separable, large scale, non-uniform correlations between decision variables and objective functions
MaF15	Convex, partially separable, large scale, non-uniform correlations between decision variables and objective functions

Performance Metric

The Inverted Generational Distance (IGD) [26] was adopted as the performance metric to assess the performance of algorithms. It has been introduced in Chapter 2.2.3 in detail. As one of the most commonly used performance metrics in the area, IGD can well evaluate the effectiveness (i.e., convergence and diversity) of a set of solutions given the problem's Pareto front is available, which is the case in the experiments conducted in this chapter. Following the suggestions in [99], the Pareto dominance relation among the solution sets obtained by different algorithms is tested. The results indicated that the compared solution sets in the experiments were nondominated to each other, and therefore, IGD is well-suited to the experiments. Notice that dominance resistant solutions are not the main challenge to be handled during the search process in this chapter, and thus the original version of IGD was adopted rather than IGD+ that was used in Chapter 4.

Compared Algorithms

To evaluate the performance of the proposed algorithm, eleven state-of-the-art many-objective evolutionary algorithms were chosen for comparison. They are NSGA-III [34], RVEA [20], MOEA/DD [89], RPD-NSGA-II [41], 1by1EA [104], MaOEA-CSS [60], VaEA [147], NSGA-II/SDR [134], SPEA2+SDE [95], SRA [86], AR-MOEA [132]. These algorithms span over the majority of categories (cf. Section 2.4). Specifically, NSGA-III, RVEA, MOEA/DD, RPD-NSGA-II, and VaEA are decomposition-based algorithms. 1by1EA and MaOEA-CSS are aggregation-based algorithms. NSGA-II/SDR modifies the conventional Pareto dominance relation. SPEA2+SDE changes density estimation of the conventional Pareto-based algorithms. Finally, SRA and AR-MOEA are two indicator-based algorithms.

All these algorithms were implemented in a recently developed Matlab platform PlatEMO[1] [133]. PlatEMO provides over 50 multi-/many-objective evolutionary algorithms and over 100 test problems (e.g., MaF test problems), as well as some widely used performance metrics (e.g., IGD metric). All the experiments were conducted on an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz with 16GB RAM, running on Windows 10. To make the comparison statistically significant, the Wilcoxon's rank sum test [161] was used throughout the experiments.

Parameter Settings

In all peer algorithms, the simulated binary crossover (SBX) [32] operator and polynomial mutation (PM) [33] operator were employed to perform variation, with both distribution indexes being set to 20. The SBX probability was set to 1.0 and PM probability was set to $1/n$, where n denotes the number of decision variables. For each test problem, each algorithm was executed 30 runs independently. The termination criterion of all algorithms was specified as the maximum number of generations ($MaxGen$)

[1] PlatEMO can be downloaded from: https://github.com/BIMK/PlatEMO.

and was set to 300.

These settings are consistent with the general settings as in [152, 96]. It is worth noting that the experiments in Chapters 3 and 4 utilized the same settings.

Additionally, the population size of four decomposition-based algorithms NSGA-III, RVEA, MOEA/DD, and RPD-NSGA-II cannot be arbitrary. The population size of the four algorithms is equal to the number of weight vectors, which is controlled by two parameters including the number of objectives (M) and the number of divisions considered along each objective (H). To generate a set of weight vectors that uniformly distributed in a simplex, the Das and Dennis's [30] systematic approach for $M \leq 5$ and the two-layer weight vectors generation approach [34] for $M > 5$ were adopted, as suggested in [34, 89]. The population size of the four algorithms was set to 105 for tri-objective problems (where $H = 13$), 126 for five-objective problems (where $H = 5$), 230 for ten-objective problems (where $H = 3$ and $H = 1$ for the boundary layer and insider layer, respectively), and 240 for fifteen-objective problems (where $H = 2$ and $H = 2$ for the boundary layer and insider layer, respectively), as suggested in [154, 50]. For a fair comparison, the population size of the other algorithms is the same as that in these four algorithms.

In addition, some of the compared algorithms require specific parameters for their execution. Here, these parameters were set according to their original papers. In RVEA, the parameter α, which controls the rate of change of the penalty function, was set to 2, and the frequency f_r, which employs reference vector adaptation, was set to 0.1, as recommended in [20]. In MOEA/DD, the neighbourhood size T was set to 10 percent of the population size, the neighbourhood selection probability δ was set to 0.9, and the penalty parameter of the PBI function θ was set to 5 according to [89]. In 1by1EA, the parameter k was set to 10 percent of the population size to balance the computational cost and the accuracy in density estimation, and the parameter R, which controls distribution threshold, was set to 1, as suggested in [104]. In MaOEA-CSS, the threshold value t of determining the difference of two closest solutions' Euclidean distance in the environmental selection was set to 0, as recommended in [60]. For SRA, the parameter p_c for the purpose of balancing the different indicators in the stochastic ranking strategy was set to 0.6, as suggested in [86].

5.3.2 Performance Comparison

Algorithm Performance on Tri-Objective Problems

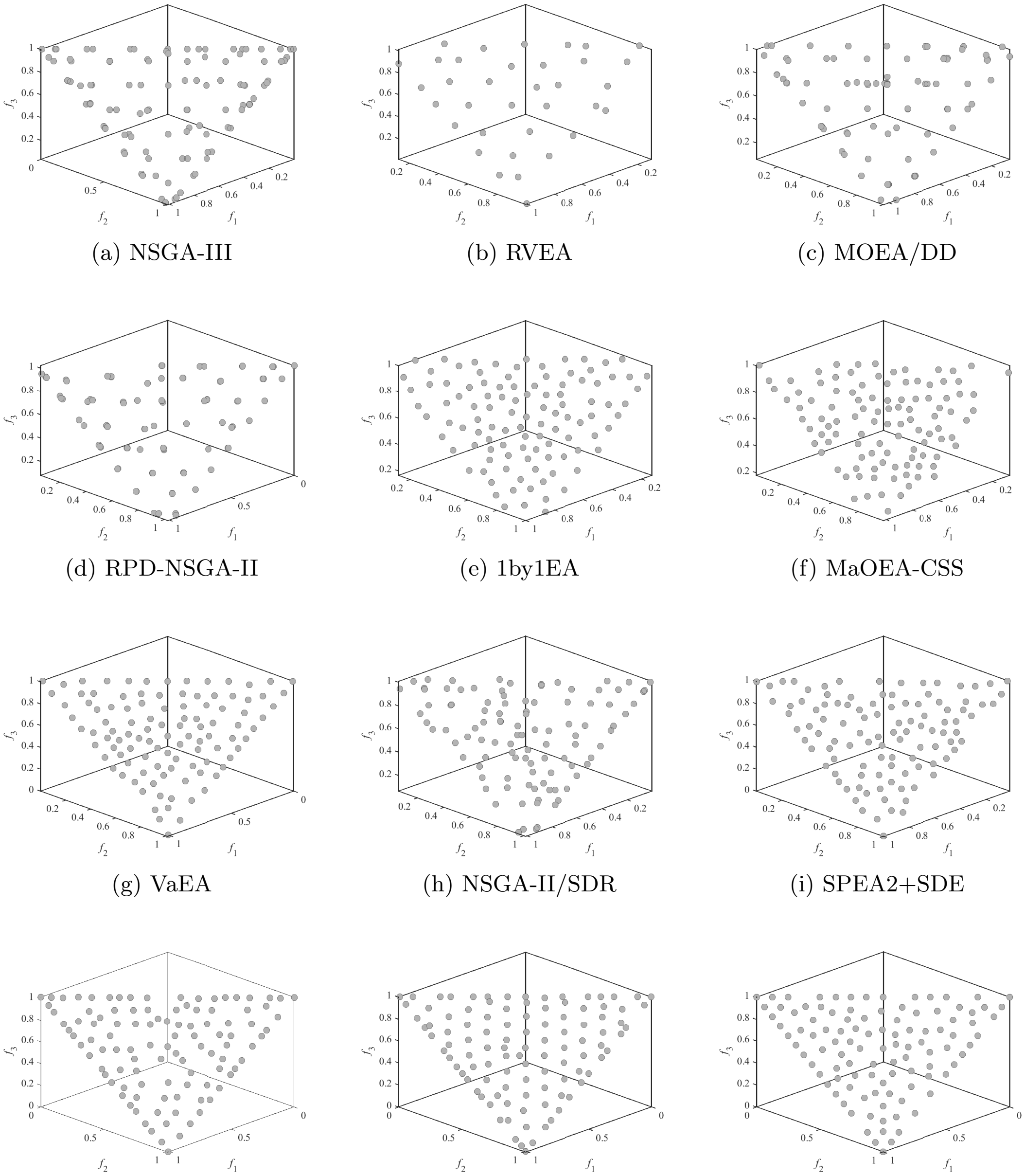
Table 5.4 shows the mean and standard deviation (in parentheses) of the IGD values obtained by BEE and compared algorithms on the tri-objective MaF problems. From Table 5.4, it can be seen that BEE obtains the best results on seven test problems (i.e., MaF1–4, MaF8–10). These seven problems are featured as a variety of characteristics, such as inverted, multimodal, badly-scaled, degenerate, and biased. Concerning pairwise comparison, BEE performs significantly better than the compared algorithms on most of the tri-objective test problems. Specifically, the proportion of the test problems where BEE significantly outperforms NSGA-III, RVEA, MOEA/DD, RPD-NSGA-II, 1by1EA, MaOEA-CSS, VaEA, NSGA-II/SDR, SPEA2+SDE, SRA, and AR-MOEA is 11/15, 12/15, 9/15, 12/15, 10/15, 10/15, 9/15, 11/15, 13/15, 12/15, and 10/15, respectively.

For a visual understanding of how solutions are distributed, Figure 5.2 plots the final solution set obtained by each algorithm on the tri-objective MaF1 with the median IGD value among all the 30 runs. As seen in Figure 5.2, VaEA, SRA, AR-MOEA, and BEE are the only four algorithms that perform well in terms of both convergence and diversity. Among them, BEE appears to be the best, with its solutions uniformly distributed on the Pareto front. For the other eight algorithms, maintaining a set of diverse solutions seems challenging, especially for four decomposition-based algorithms NSGA-III, RVEA, MOEA/DD, and RPD-NSGA-II. The main reason is that decomposition-based algorithms, which work well for MaOPs with regular Pareto front shapes, typically struggle for irregular shapes, which is the case in this instance having an inverted hyperplane. It is worth mentioning that RVEA only keeps one solution to a weight vector, which leads to an evenly distributed solution set but with fewer solutions (only 35 out of 105 solutions). It can be observed from Table 5.4 that the four algorithms (NSGA-III, RVEA, MOEA/DD, and RPD-NSGA-II) obtain poor results regarding the IGD metric with RVEA performing worst.

Table 5.4: Mean and standard deviation of the IGD values obtained by the 12 Algorithms on the tri-Objective MaF Problems.

Problem	Obj.	NSGA-III	RVEA	MOEA/DD	RPD-NSGA-II	1by1EA	MaOEA-CSS	VaEA	NSGA-II/SDR	SPEA2+SDE	SRA	AR-MOEA	BEE
MaF1	3	5.587e-2 +	7.499e-2 +	6.176e-2 +	6.348e-2 +	4.123e-2 ≈	4.135e-2 +	4.114e-2 +	3.227e-1 +	4.099e-2 +	4.519e-2 +	4.166e-2 +	4.008e-2
		(1.33e-3)	(3.01e-3)	(1.83e-3)	(1.48e-3)	(2.35e-3)	(1.16e-3)	(5.47e-4)	(3.07e-1)	(5.80e-4)	(1.25e-3)	(5.85e-4)	(2.12e-4)
MaF2	3	3.352e-2 +	3.839e-2 +	4.839e-2 +	3.486e-2 +	2.934e-2 +	2.866e-2 ≈	2.875e-2 +	3.114e-2 +	3.023e-2 +	3.501e-2 +	2.991e-2 +	2.816e-2
		(9.09e-4)	(1.10e-3)	(2.36e-3)	(2.87e-4)	(1.06e-3)	(2.69e-3)	(3.06e-4)	(6.88e-4)	(5.61e-4)	(1.78e-3)	(8.44e-4)	(2.70e-4)
MaF3	3	1.246e+0 +	2.054e+2 +	2.213e+1 +	1.176e+0 +	5.771e-1 +	1.590e+0 +	1.212e+0 +	1.203e+0 +	5.015e-1 +	1.824e+0 +	4.277e-1 +	1.536e-1
		(1.76e+0)	(6.29e+2)	(2.15e+1)	(1.89e+0)	(1.23e+0)	(1.85e+0)	(1.88e+0)	(2.46e+0)	(8.76e-1)	(2.25e+0)	(1.30e+0)	(4.06e-1)
MaF4	3	1.070e+0 +	1.875e+0 +	1.331e+0 +	5.520e-1 +	1.732e+0 +	1.056e+0 +	1.073e+0 +	9.432e-1 +	6.157e-1 +	2.226e+0 +	7.428e-1 +	4.396e-1
		(1.19e+0)	(1.87e+0)	(1.04e+0)	(4.43e-1)	(5.48e-1)	(1.05e+0)	(1.93e+0)	(9.62e-1)	(5.87e-1)	(2.43e+0)	(1.05e+0)	(5.11e-1)
MaF5	3	4.432e-1 −	2.822e-1 −	3.189e-1 −	2.465e-1 −	6.565e-1 +	3.221e-1 −	2.875e-1 −	2.147e+0 +	1.069e+0 +	6.082e-1 −	1.247e+0 +	6.326e-1
		(5.36e-1)	(2.30e-1)	(2.29e-1)	(3.81e-3)	(5.01e-1)	(2.31e-2)	(2.28e-1)	(2.92e-1)	(1.04e+0)	(6.10e-1)	(1.59e+0)	(6.25e-1)
MaF6	3	1.367e-2 +	6.180e-2 +	2.881e-2 +	5.113e-2 +	4.334e-3 −	2.396e-2 +	4.427e-3 −	7.096e-3 −	9.777e-3 +	7.738e-3 −	4.371e-3 −	8.753e-3
		(1.41e-3)	(2.06e-2)	(7.61e-4)	(1.94e-2)	(1.11e-4)	(2.76e-3)	(1.33e-4)	(9.44e-4)	(1.11e-3)	(9.93e-4)	(7.85e-5)	(4.27e-4)
MaF7	3	7.115e-2 +	1.033e-1 +	3.711e-1 +	1.024e-1 +	9.632e-2 +	1.002e-1 +	5.954e-2 −	8.177e-2 +	6.756e-2 ≈	1.258e-1 +	2.248e-1 +	6.738e-2
		(2.69e-3)	(3.58e-3)	(1.34e-1)	(6.72e-2)	(1.91e-2)	(1.24e-2)	(1.39e-3)	(4.24e-3)	(5.45e-2)	(1.06e-1)	(2.09e-1)	(6.30e-2)
MaF8	3	1.225e-1 +	1.413e-1 +	1.317e-1 +	1.233e-1 +	3.415e-1 +	7.882e-2 +	8.251e-2 +	9.021e-2 +	7.713e-2 +	5.579e-1 +	7.950e-2 +	6.767e-2
		(3.18e-2)	(1.57e-2)	(2.14e-2)	(1.18e-2)	(5.06e-2)	(1.52e-2)	(1.96e-2)	(1.16e-2)	(1.75e-2)	(1.70e+0)	(7.37e-3)	(8.87e-3)
MaF9	3	7.892e-2 +	1.949e-1 +	6.868e-2 +	2.131e-1 +	2.187e-1 +	7.818e-2 +	4.420e-1 +	8.834e-2 +	6.813e-2 +	8.101e-2 +	7.540e-2 +	6.248e-2
		(1.72e-2)	(1.19e-1)	(9.68e-3)	(2.42e-1)	(3.10e-2)	(7.06e-3)	(8.75e-2)	(1.74e-2)	(1.05e-2)	(2.43e-2)	(2.74e-2)	(4.59e-3)
MaF10	3	4.125e-1 +	5.276e-1 +	5.843e-1 +	2.704e-1 +	6.673e-1 +	4.085e-1 +	4.263e-1 +	2.573e-1 +	2.095e-1 +	6.575e-1 +	3.108e-1 +	1.880e-1
		(5.38e-2)	(7.61e-2)	(1.38e-1)	(3.83e-2)	(7.80e-2)	(3.92e-2)	(5.16e-2)	(3.79e-2)	(2.27e-2)	(8.64e-2)	(3.56e-2)	(1.97e-2)
MaF11	3	1.500e-1 −	1.764e-1 −	1.690e-1 −	1.465e-1 −	2.497e-1 +	2.380e-1 +	1.607e-1 −	1.899e-1 ≈	2.023e-1 +	2.202e-1 +	1.492e-1 −	1.894e-1
		(1.12e-3)	(8.85e-3)	(3.63e-3)	(1.70e-3)	(2.77e-2)	(2.18e-2)	(3.79e-3)	(1.42e-2)	(1.13e-2)	(1.30e-2)	(1.50e-3)	(8.31e-3)
MaF12	3	2.095e-1 −	2.170e-1 −	2.260e-1 −	2.389e-1 −	3.337e-1 +	2.786e-1 +	2.166e-1 −	2.370e-1 −	3.086e-1 +	2.827e-1 +	2.122e-1 −	2.542e-1
		(2.52e-3)	(4.89e-3)	(3.49e-3)	(4.08e-2)	(4.85e-2)	(1.78e-2)	(5.12e-3)	(7.39e-3)	(2.43e-2)	(1.25e-2)	(2.41e-2)	(5.92e-3)
MaF13	3	8.728e-2 −	1.220e-1 +	6.954e-2 −	1.317e-1 +	7.837e-2 −	8.168e-2 −	9.545e-2 ≈	9.828e-2 ≈	1.182e-1 +	1.251e-1 +	8.649e-2 −	9.465e-2
		(9.63e-3)	(2.20e-2)	(5.62e-3)	(2.18e-2)	(7.80e-3)	(6.31e-3)	(1.08e-2)	(6.96e-3)	(2.12e-2)	(1.49e-2)	(8.56e-3)	(1.14e-2)
MaF14	3	1.320e+0 +	2.945e+0 +	1.128e+0 ≈	1.597e+0 +	9.238e-1 ≈	8.823e-1 −	1.243e+0 +	1.398e+0 +	1.272e+0 +	1.369e+0 +	8.284e-1 −	1.017e+0
		(4.00e-1)	(1.07e+0)	(2.22e-1)	(5.24e-1)	(2.82e-1)	(1.24e-1)	(4.18e-1)	(4.72e-1)	(4.59e-1)	(4.55e-1)	(1.97e-1)	(2.54e-1)
MaF15	3	7.568e-1 +	8.138e-1 +	2.874e-1 −	9.375e-1 +	3.405e-1 ≈	3.426e-1 ≈	7.546e-1 +	5.902e-1 +	2.313e-1 −	2.404e-1 −	3.825e-1 +	3.585e-1
		(1.71e-1)	(2.10e-1)	(4.29e-2)	(3.93e-2)	(6.36e-2)	(4.41e-2)	(7.73e-2)	(4.34e-2)	(2.79e-2)	(2.56e-2)	(6.37e-2)	(1.18e-1)

‘+’, ‘−’, and ‘≈’ indicate that the result of BEE is significantly better than, significantly worse than, and equivalent to that of the compared algorithm at a 0.05 level by the Wilcoxon’s rank sum test, respectively.



(a) NSGA-III

(b) RVEA

(c) MOEA/DD

(d) RPD-NSGA-II

(e) 1by1EA

(f) MaOEA-CSS

(g) VaEA

(h) NSGA-II/SDR

(i) SPEA2+SDE

(g) SRA

(k) AR-MOEA

(l) BEE

Figure 5.2: The final solution set with the median IGD obtained by the 12 algorithms on the tri-objective MaF1.

Algorithm Performance on Five-Objective Problems

Table 5.5 shows the mean and standard deviation (in parentheses) of the IGD values obtained by BEE and the compared algorithms on the five-objective MaF problems. From Table 5.5, it can be seen that BEE obtains the best mean on four test problems, MaF4, MaF8, MaF10, and MaF13. Concerning pairwise comparison, BEE performs significantly better than all the compared algorithms except VaEA for more than half of the five-objective test problems. Specifically, the proportion of the test problems where BEE significantly outperforms NSGA-III, RVEA, MOEA/DD, RPD-NSGA-II, 1by1EA, MaOEA-CSS, VaEA, NSGA-II/SDR, SPEA2+SDE, SRA, and AR-MOEA is 12/15, 11/15, 14/15, 11/15, 9/15, 11/15, 7/15, 11/15, 9/15, 10/15, and 8/15, respectively.

Algorithm Performance on Ten-Objective Problems

Table 5.6 shows the mean and standard deviation (in parentheses) of the IGD values obtained by BEE and the compared algorithms on the ten-objective MaF problems. From Table 5.6, it can be seen that BEE obtains the best results on four test problems, MaF4, MaF8, MaF10, and MaF13. Concerning pairwise comparison, BEE performs significantly better than compared algorithms except SPEA2+SDE on over half of the ten-objective test problems. Specifically, the proportion of the test problems where BEE significantly outperforms NSGA-III, RVEA, MOEA/DD, RPD-NSGA-II, 1by1EA, MaOEA-CSS, VaEA, NSGA-II/SDR, SPEA2+SDE, SRA, and AR-MOEA is 12/15, 12/15, 13/15, 11/15, 12/15, 11/15, 7/15, 11/15, 5/15, 10/15, and 8/15, respectively.

For a visual understanding of how the solutions are distributed, Figure 5.3 plots the final solution set obtained by each algorithm with the median IGD value among all the 30 runs on the ten-objective MaF8, where the optimal region is a decagon in the decision space. From Figure 5.3, it can be observed that all the algorithms except RVEA, MOEA/DD, and MaOEA-CSS perform well regarding convergence (solutions located inside or very close to the polygon). Among these algorithms, NSGA-III, RPD-NSGA-II, 1by1EA, and SRA perform poorly regarding diversity, with their solutions

Table 5.5: Mean and standard deviation of the IGD values obtained by the 12 Algorithms on the five-Objective MaF Problems.

Problem	Obj.	NSGA-III	RVEA	MOEA/DD	RPD-NSGA-II	1by1EA	MaOEA-CSS	VaEA	NSGA-II/SDR	SPEA2+SDE	SRA	AR-MOEA	BEE
MaF1	5	2.507e-1 + (8.49e-3)	3.141e-1 + (3.84e-2)	3.992e-1 + (1.38e-1)	2.108e-1 + (8.58e-3)	1.197e-1 − (6.55e-3)	1.201e-1 − (2.44e-3)	1.268e-1 − (1.16e-3)	1.230e-1 − (1.70e-3)	1.253e-1 − (1.34e-3)	1.387e-1 + (2.57e-3)	1.355e-1 + (1.20e-3)	1.297e-1 (1.35e-3)
MaF2	5	1.284e-1 + (2.69e-3)	1.263e-1 + (2.06e-3)	1.605e-1 + (5.54e-3)	1.290e-1 + (1.02e-3)	9.688e-2 − (2.72e-3)	9.207e-2 − (1.46e-3)	1.058e-1 − (1.77e-3)	1.139e-1 + (2.53e-3)	1.158e-1 + (1.64e-3)	1.181e-1 + (3.11e-3)	1.115e-1 + (1.71e-3)	1.101e-1 (1.63e-3)
MaF3	5	8.795e+0 + (1.33e+1)	4.539e+1 + (4.54e+1)	2.212e+1 + (2.46e+1)	1.023e+0 + (1.35e+0)	6.951e-1 + (1.05e+0)	2.367e+0 + (3.25e+0)	5.937e+1 + (8.34e+1)	1.520e-1 − (1.27e-2)	1.913e-1 − (3.61e-1)	1.011e+0 ≈ (4.33e+0)	4.182e-1 + (7.19e-1)	2.763e-1 (5.61e-1)
MaF4	5	1.272e+1 + (1.36e+1)	7.091e+0 + (5.77e+0)	6.246e+0 + (3.16e+0)	4.493e+0 + (4.46e+0)	8.078e+0 + (6.68e-1)	5.019e+0 + (2.03e+0)	6.922e+0 + (7.75e+0)	3.685e+0 + (2.30e+0)	3.685e+0 + (1.58e+0)	4.530e+0 + (4.28e+0)	3.095e+0 + (1.62e+0)	2.630e+0 (1.46e+0)
MaF5	5	2.552e+0 + (6.64e-1)	2.589e+0 + (4.09e-1)	5.037e+0 + (6.75e-1)	2.378e+0 + (5.07e-2)	4.591e+0 + (6.37e-1)	4.639e+0 + (7.47e-1)	2.102e+0 − (4.03e-2)	1.350e+1 + (2.27e+0)	2.684e+0 + (9.26e-1)	2.647e+0 + (9.09e-1)	2.522e+0 + (6.27e-1)	2.227e+0 (3.19e-1)
MaF6	5	4.79e-2 + (8.30e-3)	1.491e-1 + (1.48e-1)	7.374e-2 + (9.37e-3)	1.490e-1 + (5.87e-2)	3.568e-3 − (7.00e-5)	6.069e-2 + (1.12e-2)	4.223e-3 − (1.59e-4)	1.747e-2 + (9.13e-3)	8.101e-3 ≈ (7.92e-4)	7.020e-3 − (2.19e-3)	3.513e-3 − (7.21e-5)	7.839e-3 (1.32e-4)
MaF7	5	3.398e-1 − (2.21e-2)	5.921e-1 + (4.14e-2)	3.001e+0 + (1.56e-6)	3.730e-1 − (1.17e-2)	4.086e-1 + (4.10e-2)	3.945e-1 + (2.42e-2)	3.408e-1 − (1.73e-2)	3.823e-1 + (3.63e-2)	3.070e-1 − (4.34e-2)	3.002e-1 − (4.47e-2)	3.259e-1 − (9.97e-3)	3.733e-1 (1.87e-1)
MaF8	5	2.034e-1 + (3.26e-2)	4.033e-1 + (3.25e-2)	3.342e-1 + (7.99e-2)	3.099e-1 + (2.98e-2)	5.191e-1 + (5.87e-2)	1.137e-1 + (8.77e-3)	1.159e-1 + (1.13e-2)	1.419e-1 + (1.67e-2)	1.184e-1 + (1.98e-2)	5.193e-1 + (9.76e-1)	1.250e-1 + (2.59e-2)	1.064e-1 (1.35e-2)
MaF9	5	5.181e-1 + (2.12e-1)	3.556e-1 + (4.95e-2)	2.669e-1 + (5.64e-2)	3.250e-1 + (6.31e-2)	2.717e-1 + (4.93e-2)	2.019e-1 + (1.92e-2)	5.023e-1 + (2.66e-1)	1.702e-1 + (1.53e-2)	9.954e-2 − (1.25e-2)	1.122e-1 − (1.94e-2)	1.250e-1 ≈ (1.94e-2)	1.206e-1 (2.02e-2)
MaF10	5	9.263e-1 + (9.27e-2)	7.799e-1 + (9.08e-2)	1.440e+0 + (1.37e-1)	5.898e-1 + (5.77e-2)	9.473e-1 + (6.61e-2)	8.581e-1 + (6.31e-2)	1.218e+0 + (1.15e-1)	8.074e-1 + (1.05e-1)	5.225e-1 + (2.75e-2)	1.213e+0 + (1.26e-1)	8.091e-1 + (7.40e-2)	4.704e-1 (2.46e-2)
MaF11	5	4.650e-1 − (3.84e-3)	4.601e-1 − (1.93e-2)	5.433e-1 + (1.35e-2)	4.549e-1 − (5.62e-3)	7.613e-1 + (6.16e-2)	6.662e-1 + (4.50e-2)	4.635e-1 − (8.76e-3)	5.723e-1 + (6.98e-2)	5.688e-1 + (2.08e-2)	5.498e-1 + (2.15e-2)	4.718e-1 − (4.02e-3)	5.323e-1 (1.50e-2)
MaF12	5	1.114e+0 − (5.38e-3)	1.136e+0 ≈ (3.98e-3)	1.207e+0 + (1.04e-2)	1.152e+0 ≈ (3.76e-2)	1.614e+0 + (1.37e-1)	1.477e+0 + (7.20e-2)	1.091e+0 − (1.69e-2)	1.150e+0 + (1.43e-2)	1.256e+0 + (2.74e-2)	1.196e+0 + (2.73e-2)	1.135e+0 ≈ (3.71e-3)	1.135e+0 (9.92e-3)
MaF13	5	2.396e-1 + (3.22e-2)	4.700e-1 + (1.27e-1)	2.534e-1 + (7.24e-2)	4.185e-1 + (5.97e-2)	1.197e-1 ≈ (1.12e-2)	1.506e-1 + (1.35e-2)	1.732e-1 + (2.68e-2)	1.821e-1 + (1.89e-2)	1.411e-1 + (1.92e-2)	1.445e-1 + (2.53e-2)	1.328e-1 + (1.27e-2)	1.195e-1 (1.47e-2)
MaF14	5	4.133e+0 + (1.33e+0)	1.669e+0 ≈ (9.68e-1)	1.585e+0 + (4.75e-1)	6.702e+0 + (2.02e+0)	1.015e+0 − (3.89e-1)	1.003e+0 − (1.73e-1)	8.138e+0 + (2.56e+0)	9.462e-1 − (2.37e-1)	1.537e+0 + (3.82e-1)	2.210e+0 + (6.28e-1)	1.318e+0 ≈ (3.81e-1)	1.223e+0 (3.34e-1)
MaF15	5	2.458e+0 + (8.30e-1)	6.161e-1 − (6.46e-2)	6.142e-1 − (1.18e-1)	1.473e+0 − (2.33e-1)	6.497e-1 − (7.31e-2)	5.510e-1 − (5.25e-2)	1.473e+0 − (1.41e-1)	8.568e-1 − (3.26e-2)	4.492e-1 − (5.34e-2)	5.260e-1 − (5.24e-2)	6.262e-1 − (5.16e-2)	1.540e+0 (1.02e+0)

‘+’, ‘−’, and ‘≈’ indicate that the result of BEE is significantly better than, significantly worse than, and equivalent to that of the compared algorithm at a 0.05 level by the Wilcoxon’s rank sum test, respectively.

Table 5.6: Mean and standard deviation of the IGD values obtained by the 12 algorithms on the ten-objective MaF Problems.

Problem	Obj.	NSGA-III	RVEA	MOEA/DD	RPD-NSGA-II	1by1EA	MaOEA-CSS	VaEA	NSGA-II/SDR	SPEA2+SDE	SRA	AR-MOEA	BEE
MaF1	10	2.960e-1 + (4.03e-3)	5.769e-1 + (7.79e-2)	3.805e-1 + (2.45e-2)	3.869e-1 + (5.75e-2)	3.936e-1 + (5.56e-2)	2.057e-1 − (1.03e-3)	2.217e-1 − (1.37e-3)	2.244e-1 − (2.17e-3)	2.311e-1 − (1.49e-3)	2.648e-1 + (3.05e-3)	2.373e-1 − (1.47e-3)	2.474e-1 (1.34e-2)
MaF2	10	2.179e-1 + (1.95e-2)	3.980e-1 + (1.85e-1)	2.572e-1 + (3.10e-2)	2.095e-1 + (3.60e-3)	4.618e-1 + (2.03e-2)	1.968e-1 − (9.51e-3)	1.800e-1 − (2.26e-3)	2.365e-1 + (1.78e-2)	1.702e-1 − (4.17e-3)	1.6155e-1 − (3.22e-3)	2.073e-1 ≈ (7.29e-3)	2.064e-1 (7.20e-3)
MaF3	10	1.947e+3 + (2.98e+3)	6.067e+0 + (6.75e+0)	6.370e+1 + (4.47e+1)	4.772e-1 + (1.01e+0)	4.235e+1 + (6.52e+1)	2.403e+0 + (6.33e+0)	3.717e+4 + (6.77e+4)	1.670e-1 + (9.34e-2)	1.066e-1 ≈ (5.51e-3)	1.862e+2 + (1.02e+3)	3.234e+0 + (1.11e+1)	1.087e-1 (1.75e-2)
MaF4	10	1.226e+2 + (8.39e+1)	1.977e+2 + (5.49e+1)	4.041e+2 + (2.17e+1)	1.403e+2 + (2.54e+1)	3.095e+2 + (2.58e+1)	1.483e+2 + (2.01e+1)	5.870e+1 + (4.76e+0)	1.961e+2 + (3.17e+1)	1.340e+2 + (1.62e+1)	1.216e+2 + (2.74e+1)	9.843e+1 + (5.11e+0)	5.416e+1 (2.35e+0)
MaF5	10	8.665e+1 + (1.58e+0)	9.436e+1 + (6.80e+0)	2.875e+2 + (1.46e+1)	7.743e+1 ≈ (4.31e+0)	2.052e+2 + (1.51e+1)	2.603e+2 + (1.70e+1)	4.705e+1 − (1.92e+0)	2.987e+2 + (2.89e+1)	6.704e+1 − (3.74e+0)	5.660e+1 − (4.90e+0)	1.006e+2 + (4.03e+0)	7.620e+1 (3.01e+0)
MaF6	10	6.623e-1 − (2.33e-1)	1.468e-1 − (8.93e-2)	1.100e-1 − (1.12e-2)	2.515e-1 − (1.64e-1)	1.918e-3 − (3.00e-5)	2.242e-1 − (8.62e-2)	1.121e+0 ≈ (4.72e-1)	7.803e-3 − (2.17e-3)	8.912e-1 − (4.56e-1)	4.159e+0 + (1.48e+0)	1.620e-1 − (2.24e-1)	1.304e+0 (5.71e-1)
MaF7	10	1.219e+0 ≈ (1.04e-1)	2.036e+0 + (5.43e-1)	2.130e+0 + (3.11e-1)	1.703e+0 ≈ (2.17e-1)	2.412e+0 + (4.15e-1)	2.426e+0 + (4.54e-1)	1.078e+0 − (4.37e-2)	1.653e+0 ≈ (3.43e-1)	8.581e-1 − (3.24e-2)	8.630e-1 − (2.40e-2)	1.418e+0 ≈ (8.22e-2)	1.675e+0 (7.06e-1)
MaF8	10	3.835e-1 + (7.49e-2)	7.819e-1 + (1.07e-1)	8.993e-1 + (1.97e-2)	5.342e-1 + (9.71e-2)	4.379e-1 + (6.95e-2)	1.222e-1 + (5.56e-3)	1.158e-1 + (2.73e-3)	1.633e-1 + (1.77e-2)	1.281e-1 + (6.73e-3)	7.649e-1 + (1.97e+0)	1.384e-1 + (4.46e-3)	1.135e-1 (1.86e-3)
MaF9	10	5.916e-1 + (1.98e-1)	8.262e-1 + (2.53e-1)	4.948e-1 + (9.87e-2)	4.154e-1 + (4.84e-2)	1.238e-1 − (1.44e-2)	1.861e-1 + (7.91e-3)	1.639e-1 − (3.65e-2)	1.880e-1 + (9.22e-3)	1.063e-1 − (1.25e-3)	1.113e-1 − (3.20e-3)	1.739e-1 ≈ (7.17e-3)	1.762e-1 (1.87e-2)
MaF10	10	1.646e+0 + (1.29e-1)	1.292e+0 + (6.00e-2)	1.712e+0 + (1.05e-1)	1.052e+0 + (2.99e-2)	1.760e+0 + (9.65e-2)	1.438e+0 + (6.38e-2)	2.453e+0 + (1.70e-1)	1.703e+0 + (1.12e-1)	1.143e+0 + (4.02e-2)	1.938e+0 + (1.46e-1)	1.489e+0 + (5.49e-2)	9.818e-1 (2.16e-2)
MaF11	10	1.262e+0 ≈ (1.31e-1)	1.148e+0 − (4.48e-2)	1.483e+0 + (2.98e-2)	1.129e+0 − (1.64e-2)	1.776e+0 + (6.44e-2)	1.680e+0 + (7.25e-2)	1.022e+0 − (1.33e-2)	1.668e+0 + (1.10e-1)	1.179e+0 − (3.97e-2)	1.182e+0 − (3.62e-2)	1.10e+0 − (3.22e-2)	1.284e+0 (1.30e-1)
MaF12	10	4.481e+0 + (4.07e-2)	4.488e+0 + (7.37e-2)	6.195e+0 + (2.09e-1)	4.632e+0 + (6.21e-2)	5.584e+0 + (2.07e-1)	5.504e+0 + (1.80e-1)	3.994e+0 − (2.48e-2)	4.500e+0 + (4.85e-2)	4.471e+0 + (4.53e-2)	4.438e+0 + (7.41e-2)	4.619e+0 + (3.47e-2)	4.214e+0 (4.85e-2)
MaF13	10	2.580e-1 + (2.71e-2)	9.350e-1 + (2.17e-1)	2.780e-1 + (2.73e-2)	5.802e-1 + (6.02e-2)	1.946e-1 + (3.22e-2)	1.506e-1 + (1.01e-2)	1.387e-1 + (1.33e-2)	1.856e-1 + (1.93e-2)	1.211e-1 + (2.58e-2)	1.276e-1 + (1.95e-2)	1.281e-1 + (7.20e-3)	1.027e-1 (1.08e-2)
MaF14	10	1.364e+1 + (5.88e+0)	1.491e+0 − (3.11e-1)	1.777e+0 ≈ (4.19e-1)	6.032e+0 + (5.79e+0)	1.627e+0 − (3.83e-1)	1.613e+0 − (1.53e-1)	1.349e+1 + (4.08e+0)	1.465e+0 − (1.60e-1)	2.656e+0 ≈ (8.41e-1)	5.655e+0 + (1.68e+0)	1.487e+0 − (4.46e-1)	2.281e+0 (1.00e+0)
MaF15	10	3.841e+0 + (1.75e+0)	1.122e+0 + (1.26e-1)	1.354e+0 + (2.06e-1)	1.384e+0 + (3.40e-1)	1.196e+0 + (6.48e-2)	9.876e-1 + (5.72e-2)	1.918e+0 + (2.90e-1)	1.196e+0 + (7.23e-2)	8.632e-1 ≈ (1.80e-1)	1.006e+0 + (1.42e-1)	1.103e+0 + (8.71e-2)	9.521e-1 (4.18e-1)

‘+’, ‘−’, and ‘≈’ indicate that the result of BEE is significantly better than, significantly worse than, and equivalent to that of the compared algorithm at a 0.05 level by the Wilcoxon’s rank sum test, respectively.

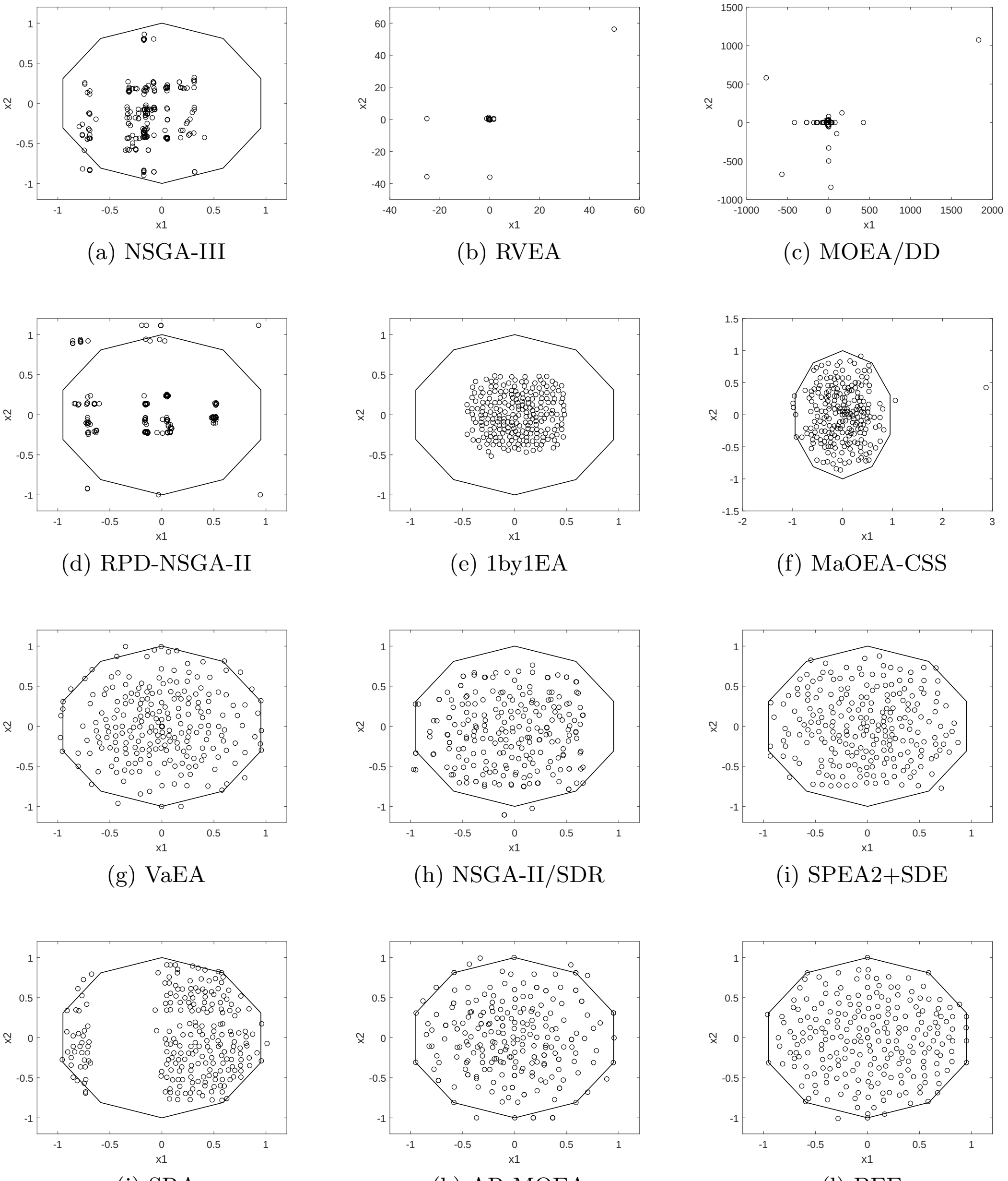
crowded in some small sub-regions, leading to a large proportion of sparse regions. One interesting observation is that 1by1EA obtains a solution set that is evenly distributed but only covers the central region of the polygon. One possible explanation is that the boundary maintenance mechanism in 1by1EA may fail to find boundary solutions when the Pareto front is irregular. This can also be seen in Figure 5.2 (e), where 1by1EA fails to obtain boundary solutions even for the tri-objective MaF1 which has an inverted simplex-like Pareto front.

Another interesting observation is that SRA sometimes provides a set of well-converged solutions, but may fail to distribute its solutions in some regions between a pair of parallel target lines within the optimal polygon. One explanation for this is that SRA could effectively guide the population into the optimal region, but may have difficulty in diversifying its solutions over the optimal polygon of the ten-objective MaF8 problem, which has many parallel target lines and constrained areas [93].

Although VaEA, NSGA-II/SDR, AR-MOEA, and SPEA2+SDE can provide a better balance between convergence and diversity, they have their own disadvantages. The solutions of VaEA and AR-MOEA are not uniformly distributed over the decagon, with some solutions crowded or even overlapping in some regions. NSGA-II/SDR and SPEA2+SDE struggle to find boundary solutions of the decagon. Finally, among these algorithms, BEE obtains very promising results on the ten-objective MaF8 problem, with a set of evenly distributed solutions covering the whole decagon.

Algorithm Performance on Fifteen-Objective Problems

Table 5.7 shows the mean and standard deviation (in parentheses) of the IGD values obtained by BEE and the compared algorithms on the fifteen-objective MaF problems. From Table 5.7, it can be seen that BEE obtains the best IGD mean on five test problems, MaF2, MaF4, MaF8, MaF10, and MaF13. Concerning pairwise comparison, BEE performs significantly better than the compared algorithms on the majority of the test problems. Specifically, the proportion of the test problems where BEE significantly outperforms NSGA-III, RVEA, MOEA/DD, RPD-NSGA-II, 1by1EA, MaOEA-CSS, VaEA, NSGA-II/SDR, SPEA2+SDE, SRA, and AR-MOEA is 11/15, 10/15, 13/15,



(a) NSGA-III (b) RVEA (c) MOEA/DD

(d) RPD-NSGA-II (e) 1by1EA (f) MaOEA-CSS

(g) VaEA (h) NSGA-II/SDR (i) SPEA2+SDE

(j) SRA (k) AR-MOEA (l) BEE

Figure 5.3: The final solution set with the median IGD obtained by the 12 algorithms on the ten-objective MaF8.

11/15, 12/15, 12/15, 9/15, 9/15, 8/15, 10/15, and 9/15, respectively.

Overall Performance

To provide an overall picture of the overall performance of the 12 different algorithms across the 60 different problem instances, the average performance scores [5] based on the IGD results are shown in Figure 5.4. For a specific problem instance, the performance score of an algorithm is the number of the peer algorithms that perform significantly better than the algorithm on the problem instance according to the statistical results. The smaller the score, the better the performance of the algorithm on the problem instance. As can be seen, the proposed obtains the best average IGD score, followed by SPEA2+SDE. Namely, in general, BEE provides competitive performance among the peer algorithms on 60 problem instances.

5.3.3 Computational Cost between SPEA2+SDE and BEE

As the time complexity of BEE is $O(MN^2)$, significantly lower than that of SPEA2+SDE ($O(MN^3)$), it is expected that this can be well reflected in their computational time required. Figure 5.5 shows the average results on the MaF10 problem with different numbers of objectives in 30 runs. Clearly, BEE always incurs lower computational cost, and their differences become clearer with the increase of the number of objectives. When the objective dimensionality is up to 15, BEE requires around 15 seconds to run, whereas SPEA2+SDE takes approximately 150 seconds.

Table 5.7: Mean and standard deviation of the IGD values obtained by the 12 algorithms on the fifteen-objective MaF Problems.

Problem	Obj.	NSGA-III	RVEA	MOEA/DD	RPD-NSGA-II	1by1EA	MaOEA-CSS	VaEA	NSGA-II/SDR	SPEA2+SDE	SRA	AR-MOEA	BEE
MaF1	15	3.196e-1 − (6.86e-3)	6.171e-1 + (8.12e-2)	5.388e-1 + (2.97e-2)	5.129e-1 + (4.24e-2)	4.846e-1 + (3.65e-2)	2.786e-1 − (2.94e-2)	2.767e-1 − (2.32e-3)	2.818e-1 − (3.45e-3)	3.034e-1 − (1.59e-2)	3.256e-1 ≈ (9.80e-3)	3.228e-1 − (8.10e-3)	3.296e-1 (1.37e-2)
MaF2	15	2.128e-1 + (1.82e-2)	4.640e-1 + (1.27e-1)	4.263e-1 + (4.64e-2)	2.233e-1 + (2.93e-3)	5.556e-1 + (2.27e-2)	2.402e-1 + (1.15e-2)	1.999e-1 + (2.27e-3)	3.492e-1 + (3.58e-2)	1.993e-1 + (1.24e-2)	1.988e-1 + (6.25e-3)	2.279e-1 + (9.56e-3)	1.930e-1 (1.13e-2)
MaF3	15	2.211e+3 + (6.90e+3)	6.271e+0 + (9.44e+0)	4.388e+1 + (4.21e+1)	3.094e-1 − (3.31e-1)	7.445e+1 + (1.05e+2)	1.540e+1 + (2.21e+1)	6.712e+4 + (1.69e+5)	1.407e-1 − (1.44e-3)	1.126e-1 ≈ (2.96e-3)	3.992e+4 + (1.84e+5)	5.024e+1 + (7.45e+1)	2.700e+0 (1.10e+1)
MaF4	15	3.358e+3 + (4.18e+2)	8.806e+3 + (1.93e+3)	1.531e+4 + (2.37e+3)	4.614e+3 + (9.87e+2)	1.058e+4 + (9.96e+2)	5.451e+3 + (9.96e+2)	1.951e+3 + (9.38e+2)	7.050e+3 + (1.26e+3)	5.513e+3 + (7.65e+2)	4.568e+3 + (1.07e+3)	3.901e+3 + (4.26e+2)	1.476e+3 (9.55e+1)
MaF5	15	2.372e+3 + (1.68e+2)	2.960e+3 + (2.96e+2)	7.302e+3 + (4.73e+1)	1.658e+3 + (1.28e+2)	6.019e+3 + (7.16e+1)	7.157e+3 + (1.11e+2)	1.385e+3 ≈ (7.86e+1)	7.304e+3 + (1.23e+2)	2.131e+3 + (1.44e+2)	1.333e+3 − (2.28e+2)	3.607e+3 + (1.59e+2)	1.413e+3 (9.01e+1)
MaF6	15	8.378e-1 − (3.71e-1)	4.023e-1 − (2.74e-1)	1.192e-1 − (1.45e-2)	3.427e-1 − (1.16e-1)	1.838e-3 − (3.97e-5)	3.294e-1 − (1.69e-1)	1.269e+0 ≈ (4.97e-1)	2.378e-2 − (7.15e-2)	1.413e+0 ≈ (8.48e-1)	4.589e+0 + (1.61e+0)	4.108e-1 − (1.43e-1)	1.342e+0 (5.41e-1)
MaF7	15	5.992e+0 + (1.42e+0)	2.521e+0 ≈ (4.21e-1)	3.430e+0 + (3.55e-2)	7.150e+0 + (6.91e-1)	3.074e+0 ≈ (3.52e-1)	5.057e+0 + (5.22e-1)	2.493e+0 ≈ (1.99e-1)	4.399e+0 + (9.35e-1)	1.452e+0 − (3.26e-2)	1.498e+0 − (1.57e-2)	3.361e+0 + (8.03e-1)	2.975e+0 (9.71e-1)
MaF8	15	3.924e-1 + (6.82e-2)	1.268e+0 + (1.88e-1)	1.266e+0 + (9.66e-2)	7.496e-1 + (1.31e-1)	4.472e-1 + (1.03e-1)	1.667e-1 + (8.05e-3)	1.579e-1 + (3.12e-3)	2.018e-1 + (2.04e-2)	1.5598e-1 + (7.37e-3)	9.902e-1 + (2.17e+0)	1.735e-1 + (5.57e-3)	1.378e-1 (1.99e-3)
MaF9	15	2.320e+0 + (4.25e+0)	1.690e+0 + (1.91e+0)	9.510e-1 + (1.52e-2)	8.189e-1 + (1.19e-1)	2.727e-1 + (2.29e-1)	2.108e-1 + (7.80e-3)	1.749e-1 ≈ (5.26e-2)	1.910e-1 + (6.42e-3)	1.316e-1 − (9.09e-4)	1.386e-1 − (2.96e-2)	1.558e-1 ≈ (7.57e-3)	1.583e-1 (3.51e-2)
MaF10	15	2.197e+0 + (9.53e-2)	1.853e+0 + (8.24e-2)	2.302e+0 + (1.21e-1)	1.588e+0 + (4.24e-2)	2.319e+0 + (1.29e-1)	2.090e+0 + (7.61e-2)	3.358e+0 + (1.92e-1)	2.418e+0 + (5.95e-2)	1.745e+0 + (4.12e-2)	2.904e+0 + (4.28e-1)	2.085e+0 + (7.12e-2)	1.570e+0 (1.69e-1)
MaF11	15	1.566e+0 ≈ (8.00e-2)	1.711e+0 + (1.28e-1)	1.913e+0 + (5.89e-2)	1.495e+0 − (4.45e-2)	2.276e+0 + (6.50e-2)	2.370e+0 + (6.60e-2)	1.665e+0 + (5.35e-2)	2.341e+0 + (9.17e-2)	1.770e+0 + (4.48e-2)	1.754e+0 + (6.30e-2)	1.549e+0 ≈ (4.37e-2)	1.562e+0 (9.50e-2)
MaF12	15	8.034e+0 + (1.45e-1)	7.665e+0 ≈ (2.51e-1)	8.765e+0 + (3.02e-1)	8.410e+0 + (6.20e-2)	9.742e+0 + (3.29e-1)	9.817e+0 + (2.44e-1)	7.026e+0 − (7.60e-2)	7.657e+0 ≈ (1.70e-1)	8.199e+0 + (4.01e-1)	7.997e+0 + (2.34e-1)	7.991e+0 + (2.10e-1)	7.624e+0 (1.90e-1)
MaF13	15	2.785e-1 + (5.05e-2)	1.198e+0 + (4.44e-1)	3.427e-1 + (3.03e-2)	6.548e-1 + (4.34e-2)	3.353e-1 + (6.67e-2)	1.978e-1 + (1.76e-2)	1.593e-1 + (2.22e-2)	1.797e-1 + (1.23e-2)	1.224e-1 + (1.53e-2)	1.325e-1 + (1.76e-2)	1.490e-1 + (9.48e-3)	1.076e-1 (1.06e-2)
MaF14	15	5.701e+0 ≈ (4.99e+0)	3.577e+0 ≈ (1.75e+0)	2.542e+0 ≈ (9.36e-1)	1.622e+0 − (4.41e-1)	4.412e+0 + (1.99e+0)	4.153e+0 + (1.23e+0)	1.454e+1 + (5.92e+0)	1.539e+0 − (2.99e-1)	2.683e+0 ≈ (9.00e-1)	5.558e+0 + (1.64e+0)	1.848e+0 − (6.14e-1)	2.961e+0 (1.29e+0)
MaF15	15	1.530e+1 + (4.44e+0)	1.391e+0 − (6.35e-2)	3.401e+0 + (9.21e-1)	4.488e+0 + (5.85e-1)	1.764e+0 − (1.12e-1)	1.520e+0 ≈ (9.81e-2)	4.231e+0 + (6.98e-1)	1.347e+0 − (4.00e-2)	1.168e+0 − (5.47e-2)	1.262e+0 − (4.51e-2)	1.956e+0 − (2.07e-1)	2.225e+0 (1.78e+0)

‘+’, ‘−’, and ‘≈’ indicate that the result of BEE is significantly better than, significantly worse than, and equivalent to that of the compared algorithm at a 0.05 level by the Wilcoxon’s rank sum test, respectively.

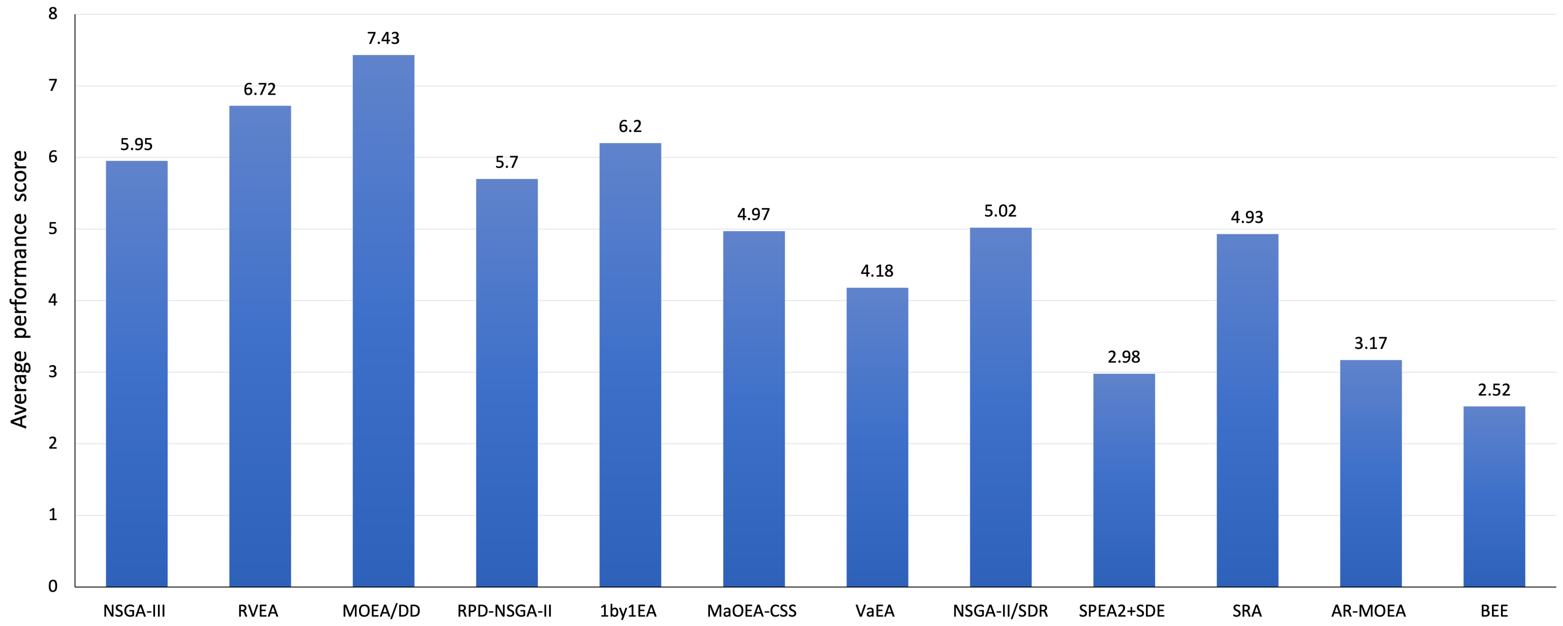


Figure 5.4: Average performance scores of the 12 algorithms on all 60 problem instances in terms of IGD. The smaller the score, the better the overall performance of the algorithm in terms of IGD.

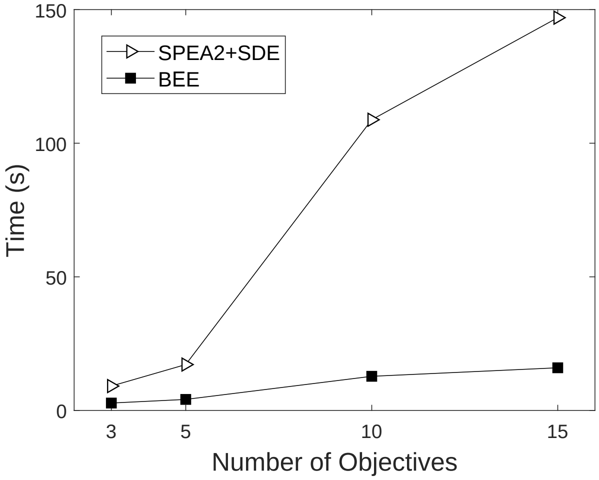


Figure 5.5: The average computational time of BEE and SPEA2+SDE on MaF10 with different numbers of objectives in 30 runs.

5.4 Summary

Balancing effectiveness and efficiency can be a challenging task in many-objective optimization, particularly in a real-world application, where the problem's Pareto front is typically irregular. In this chapter, a parameter-free many-objective evolutionary algorithm (called BEE) was proposed, with the aim of achieving high effectiveness and efficiency on problems with various Pareto front shapes. The proposed algorithm was compared with 11 state-of-the-art algorithms on 60 problem instances with up to 15 objectives. The experimental results have shown that the proposed algorithm significantly outperforms all the compared algorithms on the majority of the instances. Altogether, BEE represents a significant advance in evolutionary many-objective optimization, which provides innovative ideas for developing better algorithms.

Chapter 6

Conclusion

This chapter first summarizes the research carried out in this thesis and the contributions of each chapter in Section 6.1. Then, this chapter outlines some future research directions in Section 6.2.

6.1 Thesis Summary

This thesis aimed to address the effectiveness and efficiency of evolutionary algorithms for many-objective optimization. After reviewing key concepts and major advances in many-objective optimization, this thesis has set out to achieve three research objectives, including how to enhance the effectiveness of Pareto-based algorithms to solve many-objective optimization problems (via a case study on a challenging problem in software engineering); how to make this type of algorithms more effective to handle more generic situations (via a case study on a very popular and widely cited algorithm called BiGE); and how to develop an effective and efficient evolutionary algorithm for many-objective optimization.

To fulfill the research aim, three innovative approaches have been presented in this thesis. Specifically, a novel dominance relation in Chapter 3 has been introduced to make conventional Pareto-based algorithms more effective in many-objective optimization; an angle-based crowding degree estimation approach in Chapter 4 has been proposed to further enhance the effectiveness of BiGE for a class of many-objective optimization problems where the search process involves dominance resistant solutions; and a parameter-free many-objective evolutionary algorithm in Chapter 5 has been developed to solve optimization problems with various Pareto front shapes both effectively and efficiently. In the following, the contributions for each of these chapters are summarized.

Chapter 3 has proposed a novel aggregation-based dominance (ADO) to address the effectiveness of the conventional Pareto-based algorithms for many-objective optimization. ADO focuses on the convergence enhancement of Pareto-based algorithms. Instead of modifying Pareto dominance relation (e.g., [84, 61, 134]), ADO is used as a secondary selection criterion when Pareto dominance fails to differentiate between solutions in a population, thus providing additional selection pressure towards the Pareto front in the high-dimensional objective space.

The proposed ADO was applied to two popular Pareto-based algorithms NSGA-II and SPEA2+SDE. It has been observed that ADO can accelerate the convergence process in many-objective optimization, as shown in the experiments on the optimal feature selection problem for software product lines. Furthermore, the two modified algorithms have been demonstrated to be competitive in searching for a set of valid and diverse product configurations by comparing them with four different types of evolutionary algorithms.

Chapter 4 has presented a variant of BiGE (called aBiGE) to handle the many-objective optimization problems with dominance resistant solutions in an effective manner. In aBiGE, a novel angle-based crowding distance estimation method has been developed to replace the distance-based crowding distance estimation method in BiGE. The basic idea of angle-based crowding distance estimation is simple – by assigning a larger (worse) crowding degree to dominance resistant solutions, such solutions have a high possibility to be eliminated in the bi-goal space of proximity and diversity using the Pareto dominance relation. Therefore, aBiGE could avoid the negative influence of dominance resistant solutions during the evolutionary process.

To test the performance of the proposed aBiGE, the experiments were carried out by comparing it with the original BiGE on three representative problems involving dominance resistant solutions. The experimental results have shown that aBiGE significantly outperforms BiGE on all test problems in terms of convergence and diversity. Nevertheless, the performance of aBiGE may be slightly degraded on problems without dominance resistant solutions. This indicates that the proposed aBiGE can address the challenges in handling dominance resistant solutions during the search process while inheriting the advantages of the bi-goal evolution framework.

Chapter 5 has proposed a many-objective evolutionary algorithm (called BEE), which aims to achieve high effectiveness and efficiency for many-objective optimization. BEE focuses on the design of environmental selection, where its two operations, i.e., selecting boundary solutions and selecting non-boundary solutions work collaboratively. Specifically, the former is to determine the range of the Pareto front approximation, from which the latter is to preserve a set of well-converged and diverse solutions. During the selection procedure, the diversity is calculated based on the distance between unselected solutions and the solutions that have already been selected for the next-generation evolution. In particular, the unselected solution that is far away from the current selected solutions (i.e., solutions with good diversity, relative to the current selected solutions) is preferred. Furthermore, the convergence is measured by considering how the unselected solutions perform with respect to their closeness to the Pareto front, relative to other unselected solutions.

Systematic experiments were carried out to make an extensive comparison of BEE with 11 state-of-the-art algorithms on 60 problem instances with up to 15 objectives. The comparative results have shown that the proposed algorithm is highly practical in many-objective optimization in view of 1) competitive performance in converging its solutions into the Pareto front and also diversifying them on the front (i.e., effectiveness); 2) a reasonable amount of execution time (i.e., efficiency); 3) suitability for MaOPs with various Pareto front shapes; 4) no additional parameter except those associated with an evolutionary algorithm.

6.2 Future Work

There are several directions for future research:

First, further analysis of the parameter settings for ADO needs to be carried out. In Chapter 3, the sensitivity of the parameter k in ADO has been evaluated on the experiments with NSGA-II-ADO on six feature models with four objectives. The experiments have shown that almost all of the feature models reach roughly the highest HV values when the parameter k value is between 0.5–0.6. The influence of parameter (k) settings in ADO for more Pareto-based algorithms should be studied.

Second, the scalability of ADO in terms of the size of feature models should be investigated. Chapter 3 has shown that ADO could assist Pareto-based algorithms in searching for valid and diverse product configurations on eight published feature models with a maximum of 290 features and a randomly generated feature model with $10,000$ features. In this empirical study, there is only one large feature model out of nine feature models, but more complex and large feature models could be investigated, e.g., LVAT [1] repositories with real-world feature models including the aforementioned Linux X86 kernel model with $6,888$ features.

Third, performance investigation of aBiGE on problems without dominance resistant solutions can be conducted. In Chapter 4, aBiGE algorithm has been introduced to address the issue of effectiveness on the many-objective problems with a high probability to produce dominance resistant solutions. The experiments were carried out on nine DTLZ [38] test problems. One area of further research is to focus on the problems without dominance resistant solutions, aiming at a deeper understanding of the behaviour of the proposed aBiGE algorithm as well as a comprehensive improvement of the algorithm on both types of problems. Other well-known benchmark suites could be used, such as the walking fish group (WFG) toolkit [66] and multiline distance minimization problem (ML-DMP) [93].

Fourth, a thorough empirical investigation of different mating selection strategies in BEE should be performed. Like most many-objective evolutionary algorithms, BEE focuses on the environmental selection component of an evolutionary algorithm, leaving mating selection flexible to be implemented (Chapter 5). It would be interesting to explore the mating selection component in many-objective optimization further, particularly on how to select solutions based on their shifted position for the crossover operation.

Fifth, how the preference information of decision makers may be integrated into many-objective evolutionary algorithms is worth studying. A set of trade-off solutions approximating the entire Pareto front are usually generated by traditional many-objective evolutionary algorithms, but it may be difficult for a decision maker to understand these solutions when the number of objectives is large [90]. One direction for future research is to incorporate the preferences of the decision maker into the proposed many-objective evolutionary algorithms, which would be helpful in directing the search for solutions of interest.

Finally, the performance of the proposed innovative algorithms on more real-world applications needs to be tested. In this thesis, well-defined benchmark suites were mainly used to investigate the effectiveness of the presented many-objective evolutionary algorithms including aBiGE and BEE. Although ADO has been applied to the optimal feature selection for software product lines, it is interesting to apply the proposed many-objective approaches to more real-world scenarios in engineering [22] and manufacturing [23], and understand their characteristics in more depth.

References

[1] Linux Variability Analysis Tools (LVAT) Repository. url = https://code.google.com/archive/p/linux-variability-analysis-tools/. Accessed: 2020-11-06.

[2] S. F. Adra and P. J. Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, 2010.

[3] M. Asafuddoula, T. Ray, and H. K. Singh. Characterizing Pareto front approximations in many-objective optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 607–614, 2015.

[4] M. Asafuddoula, H. K. Singh, and T. Ray. An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. *IEEE Transactions on Cybernetics*, 48(8):2321–2334, 2018.

[5] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

[6] S. Bandyopadhyay, S. K. Pal, and B Aruna. Multiobjective GAs, quantitative indices, and pattern classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2088–2099, 2004.

[7] D. Batory. Feature models, grammars, and propositional formulas. In *International Conference on Software Product Lines*, pages 7–20. Springer, Berlin, Heidelberg, 2005.

[8] D. Benavides, S. Segura, and A. Ruiz-Cortés. Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6):615–636, 2010.

[9] B. Beume, N.and Naujoks and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[10] N. Beume. S-metric calculation by considering dominated hypervolume as klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.

[11] P. A. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2003.

[12] K. Bringmann and T. Friedrich. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3):383–402, 2010.

[13] K. Bringmann and T. Friedrich. Approximation quality of the hypervolume indicator. *Artificial Intelligence*, 195:265–290, 2013.

[14] K. Bringmann, T. Friedrich, C. Igel, and T. Voß. Speeding up many-objective optimization by Monte Carlo approximations. *Artificial Intelligence*, 204:22–29, 2013.

[15] D. Brockhoff, T. Wagner, and H. Trautmann. On the properties of the R2 indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, pages 465–472, 2012.

[16] D. Brockhoff, T. Wagner, and H. Trautmann. R2 indicator-based multiobjective search. *Evolutionary Computation*, 23(3):369–395, 2015.

[17] X. Cai, Y. Li, Z. Fan, and Q. Zhang. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):508–523, 2015.

[18] X. Cai, Z. Mei, and Z. Fan. A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors. *IEEE Transactions on Cybernetics*, 48(8):2335–2348, 2018.

[19] R. D. Carvalho, R. R. Saldanha, B. N. Gomes, A. C. Lisboa, and A. X. Martins. A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas. *IEEE Transactions on Magnetics*, 48(2):803–806, 2012.

[20] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, 2016.

[21] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, 2017.

[22] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin. Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(2):97–111, 2017.

[23] T. Chugh, N. Chakraborti, K. Sindhya, and Y. Jin. A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem. *Materials and Manufacturing Processes*, 32(10):1172–1178, 2017.

[24] P. Clements and L. Northrop. *Software product lines: Practices and patterns.* Addison Wesley, 2001.

[25] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

[26] C. A. C. Coello and M. R. Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican International Conference on Artificial Intelligence*, pages 688–697. Springer, 2004.

[27] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 283–290, 2001.

[28] D. W. Corne and J. D. Knowles. Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 773–780, 2007.

[29] D. W. Corne, J. D. Knowles, and M. J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 839–848. Springer, 2000.

[30] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[31] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[32] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[33] K. Deb and M. Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[34] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[35] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. 2002.

[36] K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. *Evolutionary Computation*, 13(4):501–525, 2005.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[38] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 105–145. Springer-Verlag, 2005.

[39] F. di Pierro, S. T. Khu, and D. A. Savi. An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, 2007.

[40] E. Dilettoso, S. A. Rizzo, and N. Salerno. A weakly Pareto compliant quality indicator. *Mathematical and Computational Applications*, 22(1):25, 2017.

[41] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y. S. Ong. A new decomposition-based nsga-ii for many-objective optimization. *IEEE transactions on systems, man, and cybernetics: systems*, 48(7):1191–1210, 2018.

[42] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer, 2005.

[43] N. Esfahani, S. Malek, and K. Razavi. GuideArch: Guiding the exploration of architectural solution space under uncertainty. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 43–52. IEEE Press, 2013.

[44] Y. Fang, Q. Liu, M. Li, Y. Laili, and D. T. Pham. Evolutionary many-objective optimization for mixed-model disassembly line balancing with multi-robotic workstations. *European Journal of Operational Research*, 276(1):160–174, 2019.

[45] A. Farhang-Mehr and S. Azarm. An information-theoretic entropy metric for assessing multi-objective optimization solution set quality. *J. Mech. Des.*, 125(4):655–663, 2003.

[46] M. Fleischer. The measure of Pareto optima applications to multi-objective metaheuristics. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 519–533. Springer, Berlin, Heidelberg, 2003.

[47] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Icga*, volume 93, pages 416–423. Citeseer, 1993.

[48] G. Fu, Z. Kapelan, J. R. Kasprzyk, and P. Reed. Optimal design of water distribution systems using many-objective visual analytics. *Journal of Water Resources Planning and Management*, 139(6):624–633, 2013.

[49] J. García-Galán, P. Trinidad, O. F. Rana, and A. Ruiz-Cortés. Automated configuration support for infrastructure migration to the cloud. *Future Generation Computer Systems*, 55:200–212, 2016.

[50] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. P. Chen. A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning. *IEEE Transactions on Evolutionary Computation*, 23(4):572–586, 2019.

[51] C. K. Goh and K. C. Tan. An investigation on noisy environments in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 11(3):354–381, 2007.

[52] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Pro-*

ceedings of the Second International Conference on Genetic Algorithms, pages 41–49. Hillsdale, NJ: Lawrence Erlbaum, 1987.

[53] R. H. Gómez and C. A. C. Coello. MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator. In *2013 IEEE Congress on Evolutionary Computation*, pages 2488–2495. IEEE, 2013.

[54] M. Gong, L. Ma, Q. Zhang, and L. Jiao. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A: Statistical Mechanics and its Applications*, 391(15):4050–4060, 2012.

[55] M. L. Griss. Software reuse architecture, process, and organization for business success. In *Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering*, pages 86–89. IEEE, 2002.

[56] F. Gu and Y. Cheung. Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 22(2):211–225, 2018.

[57] J. Guo, J. White, G. Wang, J. Li, and Y. Wang. A genetic algorithm for optimized feature selection with resource constraints in software product lines. *Journal of Systems and Software*, 84(12):2208–2221, 2011.

[58] D. Hadka and P. Reed. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evolutionary Computation*, 20(3):423–452, 2012.

[59] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the nondominated set. Imm-rep-1998-7, Institute of Mathematical Modeling, Technical University of Denmark, 1998.

[60] Z. He and G. G. Yen. Many-objective evolutionary algorithms based on coordinated selection strategy. *IEEE Transactions on Evolutionary Computation*, 21(2):220–233, 2017.

[61] Z. He, G. G. Yen, and J. Zhang. Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(2):269–285, 2014.

[62] C. Henard, M. Papadakis, M. Harman, and Y. Le Traon. Combining multi-objective search and constraint solving for configuring large software product lines. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, pages 517–528. IEEE, 2015.

[63] R. Hernández Gómez and C. A. Coello Coello. Improved metaheuristic based on the r2 indicator for many-objective optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 679–686, 2015.

[64] R. M. Hierons, M. Li, X. Liu, S. Segura, and W. Zheng. SIP: Optimal product selection from feature models using many-objective evolutionary optimization. *ACM Transactions on Software Engineering and Methodology*, 25(2):1–39, 2016.

[65] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87. IEEE, 1994.

[66] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. 10(5):477–506, 2006.

[67] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[68] K. Ikeda, H. Kita, and S. Kobayashi. Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 2, pages 957–962. IEEE, 2001.

[69] H. Ishibuchi, N. Akedo, and Y. Nojima. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, 2015.

[70] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 110–125. Springer, 2015.

[71] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190, 2016.

[72] H. Ishibuchi, Y. Tanigaki, H. Masuda, and Y. Nojima. Distance-based analysis of crossover operators for many-objective knapsack problems. In *International Conference on Parallel Problem Solving from Nature*, pages 600–610. Springer, 2014.

[73] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426. IEEE, 2008.

[74] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 527–534, 2010.

[75] S. Jiang and S. Yang. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts. *IEEE Transactions on Cybernetics*, 46(2):421–437, 2016.

[76] S. Jiang and S. Yang. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 21(3):329–346, 2017.

[77] S. Jiang, S. Yang, and M. Li. On the use of hypervolume for diversity measurement of Pareto front approximations. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[78] S. Jiang, J. Zhang, Y. S. Ong, A. N. Zhang, and P. S. Tan. A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, 45(10):2202–2213, 2014.

[79] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[80] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[81] J. B. Kollat and P. M. Reed. The value of online adaptive search: A performance comparison of NSGAII, ε-NSGAII and εMOEA. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 386–398. Springer, 2005.

[82] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1179–1186. IEEE, 2006.

[83] S. Q. Lau. *Domain analysis of E-commerce systems using feature-based model templates*. PhD thesis, University of Waterloo Ontario, Canada, 2006.

[84] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

[85] M. Laumanns and R. Zenklusen. Stochastic convergence of random search methods to fixed size pareto front approximations. *European Journal of Operational Research*, 213(2):414–421, 2011.

[86] B. Li, K. Tang, J. Li, and X Yao. Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Transactions on Evolutionary Computation*, 20(6):924–938, 2016.

[87] H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[88] K. Li, K. Deb, Q. Zhang, and S. Kwong. Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization. *Department of Electtrical and Computer Engineering, Michigan State University, East Lansing, USA, Tech. Rep. COIN Report*, 2014014, 2014.

[89] K. Li, K. Deb, Q. Zhang, and S. Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, 2015.

[90] K. Li, M. Liao, K. Deb, G. Min, and X. Yao. Does preference always help? a holistic study on preference-based evolutionary multi-objective optimisation using reference points. *IEEE Transactions on Evolutionary Computation*, 2020.

[91] K. Li, R. Wang, T. Zhang, and H. Ishibuchi. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access*, 6:26194–26214, 2018.

[92] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 18(6):909–923, 2014.

[93] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao. Multiline distance minimization: A visualized many-objective test problem suite. *IEEE Transactions on Evolutionary Computation*, 22(1):61–78, 2017.

[94] M. Li, S. Yang, and X. Liu. Diversity comparison of Pareto front approximations in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12):2568–2584, 2014.

[95] M. Li, S. Yang, and X. Liu. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365, 2014.

[96] M. Li, S. Yang, and X. Liu. Bi-goal evolution for many-objective optimization problems. *Artificial Intelligence*, 228:45–65, 2015.

[97] M. Li, S. Yang, and X. Liu. A performance comparison indicator for pareto front approximations in many-objective optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 703–710, 2015.

[98] M. Li, S. Yang, X. Liu, and R. Shen. A comparative study on evolutionary algorithms for many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 261–275. Springer, 2013.

[99] M. Li and X. Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–38, 2019.

[100] M. Li and X. Yao. What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evolutionary Computation*, 28(2):227–253, 2020.

[101] M. Li, L. Zhen, and X. Yao. How to Read Many-Objective Solution Sets in Parallel Coordinates [Educational Forum]. *IEEE Computational Intelligence Magazine*, 12(4):88–100, 2017.

[102] M. Li and J. Zheng. Spread assessment for evolutionary multi-objective optimization. In *International Conference on Evolutionary Multi-criterion Optimization*, pages 216–230. Springer, 2009.

[103] H. L. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455, 2014.

[104] Y. Liu, D. Gong, J. Sun, and Y. Jin. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Transactions on Cybernetics*, 47(9):2689–2702, 2017.

[105] G. Lizarraga-Lizarraga, A. Hernandez-Aguirre, and S. Botello-Rionda. G-Metric: an M-ary quality indicator for the evaluation of non-dominated sets. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pages 665–672, 2008.

[106] A. López Jaimes and C. A. Coello Coello. Study of preference relations in many-objective optimization. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 611–618, 2009.

[107] R. J. Lygoe, M. Cary, and P. J. Fleming. A real-world application of a many-objective optimisation complexity reduction process. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 641–655. Springer, 2013.

[108] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu. On tchebycheff decomposition approaches for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244, 2018.

[109] M. Mendonça, T. T. Bartolomei, and D. Cowan. Decision-making coordination in collaborative product configuration. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 108–113, New York, New York, USA, 2008.

[110] M. Mendonca, M. Branco, and D. Cowan. S.P.L.O.T.: Software product lines online tools. In *Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications - OOPSLA '09*, pages 761–762. ACM, 2009.

[111] S. Mostaghim and J. Teich. A new approach on many objective diversity measurement. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2005.

[112] R. Olaechea, D. Rayside, J. Guo, and K. Czarnecki. Comparison of exact and approximate multi-objective optimization for software product lines. In *Proceedings of the 18th International Software Product Line Conference - Volume 1*, pages 92–101, New York, New York, USA, 2014.

[113] D. H. Phan and J. Suzuki. R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1836–1845. IEEE, 2013.

[114] R. C. Purshouse and P. J. Fleming. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784, 2007.

[115] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu. MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, 22(2):231–264, 2014.

[116] A. Ramirez, J. Romero, and S. Ventura. A survey of many-objective optimisation in search-based software engineering. *Journal of Systems and Software*, 149:382–395, 2019.

[117] A. B. Sánchez, S. Segura, J. A. Parejo, and A. Ruiz-Cortés. Variability testing in the wild: The Drupal case study. *Software & Systems Modeling*, 16(1):173–194, 2017.

[118] H. Sato, H. Aguirre, and K. Tanaka. Variable space diversity, crossover and mutation in MOEA solving many-objective knapsack problems. *Annals of Mathematics and Artificial Intelligence*, 68(4):197–224, 2013.

[119] S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.

[120] A. S. Sayyad, J. Ingram, T. Menzies, and H. Ammar. Scalable product line configuration: A straw to break the camel's back. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 465–474. IEEE, 2013.

[121] A. S. Sayyad, T. Menzies, and H. Ammar. On the value of user preferences in search-based software engineering: A case study in software product lines. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 492–501. IEEE, 2013.

[122] J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.

[123] O. Schutze, X. Esquivel, A. Lara, and C. A C. Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012.

[124] K. Shang and H. Ishibuchi. A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 24(5):839–852, 2020.

[125] N. Siegmund, M. Rosenmüller, M. Kuhlemann, C. Kästner, S. Apel, and G. Saake. SPL Conqueror: Toward optimization of non-functional properties in software product lines. *Software Quality Journal*, 20(3-4):487–517, 2012.

[126] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[127] Y. Sun, G. G. Yen, and Z. Yi. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 23(2):173–187, 2019.

[128] K. C. Tan, E. F. Khor, and T. H. Lee. *Multiobjective evolutionary algorithms and applications*. Springer Science & Business Media, 2006.

[129] K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial intelligence review*, 17(4):251–290, 2002.

[130] T. H. Tan, Y. Xue, M. Chen, J. Sun, Y. Liu, and J. S. Dong. Optimizing selection of competing features via feedback-directed evolutionary algorithms. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis - ISSTA 2015*, pages 246–256, New York, New York, USA, 2015. ACM Press.

[131] T. Thum, D. Batory, and C. Kastner. Reasoning about edits to feature models. In *2009 IEEE 31st International Conference on Software Engineering*, pages 254–264. IEEE, 2009.

[132] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin. An indicator-based multi-objective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, 22(4):609–622, 2018.

[133] Y. Tian, R. Cheng, X. Zhang, and Y. Jin. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87, 2017.

[134] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(2):331–345, 2019.

[135] Y. Tian, X. Zhang, R. Cheng, and Y. Jin. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *2016 IEEE congress on evolutionary computation (CEC)*, pages 5222–5229. IEEE, 2016.

[136] A. Trivedi, D. Srinivasan, K. Pal, C. Saha, and T. Reindl. Enhanced multiobjective evolutionary algorithm based on decomposition for solving the unit commitment problem. *IEEE Transactions on Industrial Informatics*, 11(6):1346–1357, 2015.

[137] D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999.

[138] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary computation and convergence to a pareto front. In *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, 1998.

[139] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary computation and convergence to a pareto front. In *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228. Citeseer, 1998.

[140] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 742–756. Springer, 2007.

[141] H. Wang, L. Jiao, and X. Yao. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):524–541, 2015.

[142] H. Wang, Y. Jin, and X. Yao. Diversity assessment in many-objective optimization. *IEEE Transactions on Cybernetics*, 47(6):1510–1522, 2016.

[143] Z. Wang, K. Tang, and X. Yao. Multi-objective approaches to optimal testing resource allocation in modular software systems. *IEEE Transactions on Reliability*, 59(3):563–575, 2010.

[144] L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2011.

[145] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, 2006.

[146] J. Wu and S. Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *J. Mech. Des.*, 123(1):18–25, 2001.

[147] Y. Xiang, Y. Zhou, M. Li, and Z. Chen. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):131–152, 2017.

[148] Y. Xiang, Y. Zhou, Z. Zheng, and M. Li. Configuring software product lines by combining many-Objective optimization and SAT solvers. *ACM Transactions on Software Engineering and Methodology*, 26(4), 2018.

[149] H. Xu, W. Zeng, D. Zhang, and X. Zeng. MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition. *IEEE Transactions on Cybernetics*, 49(2):517–526, 2019.

[150] Y. Xue, M. Li, and X. Liu. Angle-based crowding degree estimation for many-objective optimization. In *International Symposium on Intelligent Data Analysis*, pages 574–586. Springer, 2020.

[151] Y. Xue, J. Zhong, T. H. Tan, Y. Liu, W. Cai, M. Chen, and J. Sun. IBED: Combining IBEA and DE for optimal feature selection in software product line engineering. *Applied Soft Computing*, 49:1215–1231, 2016.

[152] S. Yang, M. Li, X. Liu, and J. Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

[153] G. G. Yen and H. Lu. Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation*, 7(3):253–274, 2003.

[154] J. Yuan, H. L. Liu, F. Gu, Q. Zhang, and Z. He. Investigating the properties of indicators and an evolutionary many-objective algorithm based on a promising region. *IEEE Transactions on Evolutionary Computation*, 25(1):75–86, 2021.

[155] Y. Yuan, H. Xu, B. Wang, and X. Yao. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37, 2016.

[156] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[157] X. Zhang, Y. Tian, and Y. Jin. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(6):761–776, 2014.

[158] Z. Zhu, G. Zhang, M. Li, and X. Liu. Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1344–1357, 2016.

[159] E. Zitzler. *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, Zurich, Switzerland: Swiss Federal Institute of Technology (ETH), 1999.

[160] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[161] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of Pareto set approximations. In *Multiobjective Optimization*, pages 373–404. Springer, Berlin, Heidelberg, 2008.

[162] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842. Springer, Berlin, Heidelberg, 2004.

[163] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report*, 103, 2001.

[164] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithmsa comparative case study. In *International Conference on Parallel Problem Solving from Nature*, pages 292–301. Springer, 1998.

[165] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[166] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.