Evolutionary Algorithms for Static and Dynamic Multiobjective Optimization

PhD Thesis

By

Shouyong Jiang

Faculty of Technology

De Montfort University

This dissertation is submitted for the degree of Doctor of Philosophy

December 2016

To my loving parents, my wife and daughter . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. Some of the work has been previously published in journals and conferences. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

By

Shouyong Jiang
December 2016

Acknowledgements

The completion of this dissertation could not have been possible without the help and support of the following. I thank all of them with my deepest gratitude.

To begin with, I would like to express my sincere thanks to Prof. Shengxiang Yang and Prof. Ferrante Neri for providing persistent support and valuable suggestions within the past three years. Prof. Shengxiang Yang has been a tremendous mentor for me and always provides encouragement, help and guidance whenever I encounter difficulties. His kindness, advice and encouragement are priceless for me to complete the PhD project. Prof. Ferrante Neri is very supportive and hard-working. His invaluable input and guidance has been supporting me during the whole research.

I also wish to thank my enlightenment supervisor Prof. Hongru Li for fostering my interest in academic research, directing my career development, and offering immense help over the years.

Special thanks belongs to my colleagues and friends who have been involved in discussions that have helped to conduct a variety of research studies over the past years: Dr Michalis Mavrovouniotis, Jayne Eaton, Muhanad Tahrir Younis, Conor Fahy, Phuong Thi Mai Nguyen, Manal Alghieth, Xiaobin Liu and academic visitors Prof. Yong Wang, Prof. Jinglei Guo, Dr. Shuzhen Wan, Mr. Rui Hu and Dr. Zhong Ma. Without them, the research would not have been such a pleasant and enjoyable experience in my life.

Furthermore, I would like to extend my sincere gratitude to the following people I have come across during the PhD study for providing invaluable help, advice, and support for my research: Prof. Xin Yao, Prof. Xiaoli Li, Prof. Qingfu Zhang, Dr. Miqing Li, Dr. Ke Li, Dr. Hu Zhang, Prof. Aiming Zhou, Prof. Andries Engelbecht, and Dr. Marde Greeff.

I am extremely grateful to my family especially my parents, my brothers, my wife's family, my wife Huichao and my daughter Jiayi for their endless love and support. Special acknowledgement goes to my wife. Without her love, support, encouragement and patience I would not have been able to finish this work.

Finally, I would like to thank School of Computer Science and Informatics, De Montfort University and the "Evolutionary Computation for Dynamic Optimisation in Network Environments" (ECDONE) project sponsored by the Engineering and Physical Science Research Council (EPSRC) of the UK for funding my three-year PhD research.

Abstract

Many real-world optimization problems consist of a number of conflicting objectives that have to be optimized simultaneously. Due to the presence of multiple conflicting objectives, there is no single solution that can optimize all the objectives. Therefore, the resulting multiobjective optimization problems (MOPs) resort to a set of trade-off optimal solutions, called the Pareto set in the decision space and the Pareto front in the objective space. Traditional optimization methods can at best find one solution in a single run, thereby making them inefficient to solve MOPs. In contrast, evolutionary algorithms (EAs) are able to approximate multiple optimal solutions in a single run. This strength makes EAs good candidates for solving MOPs. Over the past several decades, there have been increasing research interests in developing EAs or improving their performance, resulting in a large number of contributions towards the applicability of EAs for MOPs. However, the performance of EAs depends largely on the properties of the MOPs in question, e.g., static/dynamic optimization environments, simple/complex Pareto front characteristics, and low/high dimensionality. Different problem properties may pose distinct optimization difficulties to EAs. For example, dynamic (time-varying) MOPs are generally more challenging than static ones to EAs. Therefore, it is not trivial to further study EAs in order to make them widely applicable to MOPs with various optimization scenarios or problem properties.

This thesis is devoted to exploring EAs' ability to solve a variety of MOPs with different problem characteristics, attempting to widen EAs' applicability and enhance their general performance. To start with, decomposition-based EAs are enhanced by incorporating two-phase search and niche-guided solution selection strategies so as to make them suitable for solving MOPs with complex Pareto fronts. Second, new scalarizing functions are proposed and their impacts on evolutionary multiobjective optimization are extensively studied. On the basis of the new scalarizing functions, an efficient decomposition-based EA is introduced to deal with a class of hard MOPs. Third, a diversity-first-and-convergence-second sorting method is suggested to handle possible drawbacks of convergence-first based sorting methods. The new sorting method is then combined with strength based fitness assignment, with the aid of reference directions, to optimize MOPs with an increase of objective dimensionality. After that, we study the field of dynamic multiobjective optimization where objective functions and constraints can change over

time. A new set of test problems consisting of a wide range of dynamic characteristics is introduced at an attempt to standardize test environments in dynamic multiobjective optimization, thereby aiding fair algorithm comparison and deep performance analysis. Finally, a dynamic EA is developed to tackle dynamic MOPs by exploiting the advantages of both generational and steady-state algorithms. All the proposed approaches have been extensively examined against existing state-of-the-art methods, showing fairly good performance in a variety of test scenarios.

The research work presented in the thesis is the output of initiative and novel attempts to tackle some challenging issues in evolutionary multiobjective optimization. This research has not only extended the applicability of some of the existing approaches, such as decomposition-based or Pareto-based algorithms, for complex or hard MOPs, but also contributed to moving forward research in the field of dynamic multiobjective optimization with novel ideas including new test suites and novel algorithm design.

Table of contents

List of figures

List of tables

List of Abbreviations

Algorithms

ACD	Adaptive Constrained Decomposition
AGR	Adaptive Global Replacement
dCOEA	Dynamic Competitive-cooperative Coevolutionary Algorithm
DFCS	Diversity First and Convergence Second
DU	Distance-based Update
EA	Evolutionary Algorithm
MOEA	Multiobjective Evolutionary Algorithm
MOEA/D	Multiobjective Evolutionary Algorithm Based on Decomposition
MSF	Multiplicative Scalarizing Function
PBI	Penalty-based Boundary Intersection
PICEA-g	Preference-inspired Coevolutionary Evolutionary Algorithm
PSF	Penalty-based Scalarizing Function
SDE	Shift-based Density Estimation
SGEA	Steady-state and Generational Evolutionary Algorithm
SPEA/R	Strength Pareto Evolutionary Algorithm Based on Reference
STM	Stable Matching
TCH	Weighted Tchebycheff
TPN	Two-Phase and Niching Strategy
WS	Weighted Sum

Acronyms

DCI	Diversity Comparison Indicator
DE	Differential Evolution
DMOP	Dynamic Multiobjective Problem
EDMO	Evolutionary Dynamic Multiobjective Optimization
EMO	Evolutionary Multiobjective Optimization
FDA	Farina-Deb-Amoto test suite
GD	Generational Distance
HV	Hypervolume
HVD	Hypervolume Difference
IGD	Inverted Generational Distance
JY	Jiang-Yang test suite
MaOP	Many-objective Optimization Problem
MOP	Multiobjective Optimization Problem
MS	Maximum Spread
NS	Nondominated Set
PCI	Performance Comparison Indicator
PF	Pareto Front
PS	Pareto Set
R(PM)	Robustness of Performance Measure
RMS	Revised Maximum Spread
SBX	Simulated Binary Crossover
UDF	Unconstrained Dynamic Functions
WFG	Walking-Fish-Group test suite
ZDT	Zitzler-Deb-Thiele test suite

Symbols

Δ_p	Hausdorff indicator
Ω_f	objective space
Ω_t	time space
Ω_x	search space
$\preceq$	Pareto dominance relation
σ_{share}	niche radius in niche sharing
$\Delta(c)$	the size of improvement region
$B(i)$	the neighborhood of subproblem i
D_{ext}	crowdedness for extreme populations
D_{mid}	crowdedness for intermediate populations
f_i	the i-th objective value
gen	generation counter
H_M^k	the number of reference directions
M	the number of objectives
$MaxGen$	the maximum number of generations
N	population size
n	the number of decision values
n_r	the maximum number of replaceable individuals in MOEA/D
p_m	crossover probability
r_i	the i-th component of nadir point
w_i	the i-th component of a weight vector
x_i	the i-th decision variable
z_i^*	the i-th component of ideal point

Chapter 1

Introduction

Many real-life problems ranging from engineering to economics involve multiple objectives to be optimized [46]. For example, the decision on the purchase of a flight ticket depends on departure/arrival places and time, flight duration, safety, airline service, and the cost. The objectives are often in conflict with each other, and problems having this conflicting nature are referred to multiobjective optimization problems (MOPs). As a result, there is no single solution that could minimize or maximize all the objectives simultaneously. Instead, the optima of an MOP are a set of trade-off solutions that compromise objectives, known as the Pareto set (PS) in the decision space and Pareto front (PF) in the objective space. Solutions in the PS are incomparable as each of them represents a certain compromise between the objectives.

Mimicking the process of nature evolution, i.e., the survival of the fittest [38], evolutionary algorithms (EAs) are an important method applied to solve MOPs [34]. The popularity of using EAs for multiobjective optimization is due to the following advantages. First, EAs do not require much knowledge about problem properties, e.g., continuity and differentiability, compared with traditional mathematical programming methods [158]. Second, EAs can provide a set of solutions by employing a population of candidates and evolving them simultaneously in a single run. Third, EAs have the ability to handle complex search environments, e.g., the search space is very large or has disconnected regions.

MOPs are a general term to refer to problems with at least two objectives. However, different MOPs may have different problem properties, resulting in distinct optimization difficulties for EAs. Thus, it is not trivial to classify MOPs into different categories according to their problem properties. In the community of evolutionary computation, there are three popular subcategories of MOPs: (static) MOPs, (static) many-objective optimization problems (MaOPs), and dynamic MOPs (DMOPs).

MOPs or static MOPs are often related to problems with two or three objectives. After decades of development, the research on MOPs has achieved fruitful results. Theoretical

foundation and algorithm design have been deeply studied. Despite that, some work related to multiobjective optimization has not been fully understood. Wider problem types should be investigated and more improvements are expected to be made toward existing EAs.

Many-objective problems differentiates from MOPs in the number of objectives. Many-objective problems involve more than three objectives. Due to the increase in the number of objectives, existing EAs that are originally designed for MOPs suffer from significant loss of selection pressure and their performance deteriorates dramatically [43]. In order to make EAs applicable to many-objective problems, effective selection methods are needed to increase selection pressure during the search.

Dynamic MOPs are a kind of MOPs in which objective functions and/or constraints change over time. Due to environmental changes in dynamic MOPs, feasible solutions can become infeasible, and promising ones can become unexpectedly poor. The consequences of dynamic environments give rise to new challenges to EAs. However, the research on evolutionary dynamic multiobjective optimization is now at the very early stage. There is a lack of diverse test environments, suitable performance metrics, and effective algorithms in the field of dynamic multiobjective optimization. These open issues need to be addressed in order to promote the development of this field.

This chapter is organized as follows. First, the motivation of undertaking this research is explained. Thereafter, the main objectives of this research are stated, followed by an outline of contributions. Finally, the overall structure of this thesis is provided.

1.1 Motivation

Evolutionary multiobjecitve optimization (EMO) in static and dynamic environments is a challenging research topic because it not only involves the simultaneous optimization of multiple complex objectives, but also requires multiobjective EAs (MOEAs) to be capable of addressing many issues related to different optimization environments. There have been great advances made in multiobjective optimization, many-objective optimization and dynamic multiobjective optimization in recent years. However, the ability and applicability of EAs to various optimization environments have not yet been well understood.

The primary motivation of this work is to extend EAs' applicability to a wide variety of MOPs, including both static and dynamic optimization environments, and facilitate theoretical foundations for dynamic multiobjective optimization. The following paragraphs are devoted to explaining the incentive of this research work in detail.

A large number of EAs have been proposed for multiobjective optimization, and they have been shown to be very promising for solving a variety of MOPs [43, 120, 188]. Among them, decomposition-based EAs [188] are a popular class of methods and have

become a baseline algorithm by winning the continuous multiobjective optimization competition in the 2009 IEEE Congress on Evolutionary Computation [190]. However, some recent studies have shown that the decomposition-based EAs could be influenced by PF geometries of MOPs, particularly when optimizing MOPs with complex PFs [139]. As a result, they fail to provide a good coverage and distribution of solutions. This issue needs to be addressed in order to improve the applicability of this kind of EAs.

Scalarizing functions are widely used in multiobjective optimization. Scalarizing functions are an important tool to convert a MOP into a number of MOPs or sing-objective problems [188]. Through conversion, the MOP becomes easier to be handled. Despite great success, scalarizing functions are not yet fully understood. A particular question related to scalarizing functions is: how do they affect the search behaviour of EAs? To answer this question soundly, a deep investigation is required.

Many-objective problems are more challenging than MOPs for EAs. Due to the increase in the number of objectives, Pareto dominance relation becomes ineffective to discriminate solutions, leading to a dramatical loss of selection pressure during the search [43]. Facing this issue, a natural question arises – is there any other effective method in place of the Pareto dominance based selection method? This is an interesting question and worthwhile of investigation.

Dynamic MOPs frequently appear in real-world applications [55]. This kind of problems brings new challenges to EAs. Dynamic test problems play an important role in deeply studying and understanding dynamic environments. They are also helpful for algorithm design and development. However, in the field of dynamic multiobjective optimization, there is a lack of standard test environments that can be used to deeply and comprehensively study the challenges caused by dynamic environments and assess EAs' ability to deal with these challenges. Therefore, standard test problems are desirable in order to advance the development of EAs for dynamic MOPs.

While EAs have been shown to be powerful and efficient optimizers for static MOPs, they encounter difficulties in solving dynamic MOPs. Due to environmental changes, EAs are very likely to lose population diversity, and nondominated solutions discovered in the previous environment may be no longer nondominated [61]. The changing environments require EAs to be capable of maintaining diversity, detecting changes, and converging quickly. In other words, a good EA should be able to track the changing PS/PF and provide a set of well-diversified solutions for each environmental change. In this research field, there is a great need of good EAs that are able to handle dynamic environments and serve as baseline algorithms for algorithm comparison.

1.2 Objectives

The overall purpose of this work is to make EAs applicable to various MOPs with different problem characteristics and optimization difficulties. To accomplish the overall objective some specific objectives are proposed:

- A review of decomposition-based EAs for MOPs with complex PFs will be conducted. This will result in a further and extensive understanding of strengths and weaknesses of decomposition-based EAs when solving this kind of problems. Novel techniques will be proposed to alleviate or overcome the observed drawbacks, and these new techniques will be examined on complex-PF MOPs and compared with other peer methods to show the effectiveness.

- EAs with scalarizing functions for MOPs will be systematically studied. First, potential drawbacks of existing scalarizing functions will be analysed qualitatively. In view of the drawbacks of incorporating these scalarizing functions into EAs for MOPs, new scalarizing functions will be proposed. These new scalarizing functions will be studied quantitatively and qualitatively. They will also be integrated into EAs to solve MOPs, and experimental studies will be carried out to show the superiority of the proposed methods.

- Different selection methods will be investigated in the field of many-objective optimization. Particularly, the difference between diversity-biased and convergence-biased selection methods will be studied. An extensive experimental study will be conducted to identify the most suitable one from these two kinds of selection methods for many-objective optimization. The selection method identified will be combined with new strategies to form a new algorithm for many-objective optimization. The new algorithm will be validated through fair comparisons with state of the arts.

- An extensive investigation and thorough analysis in test suites for dynamic multi-objective optimization will be carried out to identify the common dynamic features among existing test suites. Based on the analysis, new and representative dynamic characteristics that are rarely considered will be assessed, and this will lead to the development of a new dynamic test suite. The new test suite will be used to study the ability of EAs to handle environmental changes. The strengths and weaknesses of some EAs for dynamic multiobjective optimization will be summarized.

- On the basis of the above testing, a new EA framework will be developed for dynamic multiobjective optimization. The new EA will take the advantage of steady-state EAs in promoting convergence and the advantage of generational EAs in maintaining diversity, thereby having the ability to react to changes quickly and search

new PFs effectively. The proposed EA will be examined on a wide range of test problems, and its performance will be validated by comparing existing popular dynamic multiobjective optimizers.

1.3 Contributions

The following summarises the main contributions of this thesis:

- A new two-phase search method and a niche-guided selection method are integrated into decomposition-based EAs for solving MOPs with complex PFs (Chapter 3). The popular decomposition-based EA, e.g., MOEA based on decomposition (MOEA/D) [188], maintains diversity on the assumption that uniform weight vectors provide a set of uniformly-distributed solutions. However, this assumption hardly holds when the PF to be approximated is irregularly shaped, e.g., MOEA/D is likely to generate duplicate solutions if the PF has disconnected segments. Moreover, MOEA/D uses a simple solution selection method for mating and replacement without considering the density of solutions, which can easily lead to overexploitation/underexploitation in some regions, particularly for complex PFs. In view of these drawbacks, a two-phase search method is proposed to divide the search into two phases, where the first phase is devoted to a coarse search and the second phase helps refine the solutions obtained from the first phase and improve their distribution. A niche-guided selection is introduced to reduce the chance of selecting solutions from overcrowded regions and enhance the search in underexplored regions, thereby guaranteeing good population diversity during the search.

  The two-phase and niche-guided (TPN) strategy is tested on a number of irregular MOPs with different PF characteristics, e.g., sharp-peak/long-tail PFs, disconnected PF segments and multimodal PFs. The experimental study has shown the effectiveness of TPN in improving MOEA/D for solving complex MOPs. Furthermore, MOEA/D with TPN has been compared with other peers and state-of-the-art algorithms, demonstrating that it is very promising for finding well-converged and uniformly-distributed solutions for MOPs with various and complex PFs.

- New scalarizing functions (i.e., the multiplicative scalarizing function (MSF) and penalty-based scalarizing function (PSF)) are introduced for hard MOPs and their impact on search behaviour is deeply investigated (Chapter 4). Unlike the commonly-used existing scalarizing functions, e.g., the weighted sum (WS), the weighted Chebycheff (TCH) and the penalty-based boundary intersection (PBI) [188], which are difficult to maintain the balance between diversity and convergence for hard search environments, the MSF and PSF can induce adjustable improvement regions

so that diversity can be well controlled during the search. Also, MSF and PSF promote the similar size of improvement regions. As a result, all the solutions to the subproblems can make equally good progress during the evolution. Besides, an efficient EA based on the proposed scalarizing functions (eMOEA/D) is proposed for multiobjective optimization.

The eMOEA/D algorithm has been investigated on a number of difficult MOPs where local attractors can cause evolutionary stagnation. Compared with nine state-of-the-art scalarizing-based or decomposition-based algorithms, eMOEA/D is more capable of balancing diversity and convergence, thereby producing high-quality solutions at the end of the search. This implies the usefulness of the new scalarizing functions in dealing with hard MOPs.

- A novel diversity-first-and-convergence-second (DFCS) selection approach is proposed to handle many-objective optimization (Chapter 5). Unlike most existing Pareto-based EAs that approximate the PF in a convergence-first-and-diversity-second manner, DFCS considers diversity as the first selection criterion. This way, the loss of selection pressure resulting from Pareto dominance can be rescued by the increase of diversity emphasis. On the basis of DFCS, a new EA, called SPEA/R, is introduced to deal with many-objective problems. SPEA/R employs a set of diverse weight vectors to partition the objective space into a number of subspaces and uses a Pareto-based method to do fitness assignment. As a result, SPEA/R makes it possible to use Pareto dominance in many-objective optimization.

  The effectiveness and promise of SPEA/R has been verified on both MOPs and many-objective problems. This refutes the common belief that Pareto dominance is ineffective in many-objective optimization. An interesting finding from empirical studies is that diversity may be more important than convergence in the case of optimizing many-objective problems considered in this thesis.

- A new test suite (i.e., JY) is developed for dynamic multiobjective optimization (Chapter 6). The JY test suite is proposed to meet the need of standard test environments in the field of dynamic multiobjective optimization. Unlike some existing test suites, JY contains test problems with typical and diverse dynamic characteristics, e.g., mixed PFs in terms of convexity and concavity that change over time, and non-monotonic and time-varying linkages between variables instead of static monotonic variable-linkage used in the literature. The JY test suite plays an important role in furthering theoretical analysis and widening insights in the understanding of dynamics involved in changing optimization environments.

  The JY test suite has been adopted to study and assess the ability of different MOEAs to deal with dynamic environments. According to the used performance

metrics including both existing and new ones, JY is able to facilitate a comprehensive understanding of MOEAs in response to environmental changes. Therefore, JY is a promising toolbox for the research and development of dynamic multiobjective optimization.

- A new EA based on the combination of generational and steady-state search methods is suggested for dynamic multiobjective optimization (Chapter 7). In the field of dynamic multiobjective optimization, an important topic is to design effective and efficient EAs that can handle dynamic environments well. In other words, EAs should be able to maintain diversity and convergence well whenever there is an environmental change. The proposed EA (i.e., SGEA) is inspired by the fast convergence of steady-state EAs and the good diversity maintenance of generational EAs. Unlike most existing approaches, SGEA detects and reacts to changes in a steady-state manner and maintains population diversity in a generational manner. It also uses history information and the new information about the population to relocate the current population so that the population is close to the optima in new environments. This way, SGEA is easy to track the change of the PF and solve the current problem before new environments arrive.

  The SGEA has been studied on a wide variety of test suites with different characteristics. Empirical studies have revealed that SGEA reacts to environmental changes faster and more stably than its competitors. SGEA works generally well on most of the considered test problems. This work will attract more research interests in dynamic multiobjective optimization.

1.4 Overview

The main purpose of this work is to investigate the suitability of EAs for solving different kinds of MOPs and make subsequent improvements on the applicability of EAs for these MOPs. As the performance of EAs is very problem-specific, the thesis is organized such that (roughly) each chapter covers one kind of MOPs and/or an EA designed specifically for this kind of MOPs. Thus, there may be a lack of strong links between some chapters. Generally, however, Chapters 3, 4, and 5 can be in one group and are devoted to studying static MOPs with different characteristics, whereas Chapters 6 and 7 are another group which focuses on dynamic MOPs. Static MOPs from Chapter 3 to Chapter 5 have an increasing level of optimization difficulties in the first group, and all the corresponding EAs designed use the same idea of decomposition for dealing with these increasingly difficult MOPs. The second group starts from constructing dynamic test environments to compare existing EAs in handling dynamic characteristics (which are presented in Chapter 6), and

ends with designing a new EA (in Chapter 7) to overcome the drawbacks identified by the previous chapter.

To be specific, the thesis is organized as follows.

First, in Chapter 2, background knowledge and related work are presented. This chapter introduces three main topics related to evolutionary multiobjective optimization, i.e., static multiojective optimization, many-objective optimization and dynamic multiobjective optimization. In each topic, related work is described, including basic concepts, test suites, mainstream approaches, and performance metrics.

Chapter 3 starts with the motivation of improving decomposition-based EAs for solving MOPs with complex PFs. Then, a new two-phase search strategy and a niche-guided selection strategy are suggested to be used in decomposition-based EAs. After that, extensive experiments are carried out to verify the effectiveness of the proposed method, followed by deep sensitivity analysis.

Chapter 4 presents two new scalarizing functions after the illustration of potential drawbacks of existing scalarizing functions. Based on the new scalarizing functions, an efficient MOEA/D (i.e., eMOEA/D) is suggested. Broad algorithm comparisons are then conducted to show the promise of the proposed algorithm, followed by a further investigation into the impact of different strategies.

In Chapter 5, a new diversity-first-and-convergence-second (DFCS) selection method is introduced to overcome the disadvantages of convergence-first based methods. With the aid of DFCS and reference directions, a new optimizer, i.e., SPEA/R, is then suggested. Experimental studies are conducted on both multiobjective and many-objective optimization. Finally, further investigations are made to show the strengths and weaknesses of SPEA/R and peer methods.

Chapters 6 and 7 focus on addressing open issues in the field of dynamic multiobjective optimization. Specifically, Chapter 6 attempts to construct a diverse dynamic multiobjective test environments by introducing a new test-bed consisting of diverse and typical dynamic characteristics. The test bed is then used to examine the performance of some existing dynamic optimizers, providing a deep understanding of dynamics in time-changing environments.

Based on the analysis of empirical results provided in the previous chapter, Chapter 7 introduces a dynamic EA to deal with dynamic environments. The proposed EA, i.e., SGEA, is tested and evaluated on various test environments, and general concerns about this algorithms is discussed.

Chapter 8 summarizes the work presented in this thesis and points out contributions to corresponding research field. Future research directions are also outlined.

Chapter 2

Background

This chapter is devoted to presenting preliminary knowledge of this thesis. The structure of this chapter is organized as follows. Section 2.1 describes related work on evolutionary multiobjective optimization, including problem definition and evolutionary approaches. Section 2.2 reviews the work on evolutionary many-objective optimization, followed by a review of the research on evolutionary dynamic multiobjective optimization (EDMO) in Section 2.3. Section 2.4 summarizes this chapter.

2.1 Evolutionary Multiobjective Optimization

2.1.1 Multiobjective Optimization Problems

2.1.1.1 Problem Definition

A multiobjective optimization problem (MOP) can be mathematically described as follows:

$$\begin{aligned} \min \quad & f(x) = \Big(f_1(x), f_2(x), \ldots, f_M(x)\Big)^T \\ s.t. \quad & x \in \Omega_x, \end{aligned} \tag{2.1}$$

where $\Omega_x \subseteq \mathbb{R}^n$ is the decision space and $x = (x_1, \ldots, x_n)^T$ is a candidate solution. $f : \Omega_x \mapsto \Omega_f \subseteq \mathbb{R}^M$ contains M objective functions, and Ω_f is the attainable objective space. In this thesis, we only focus on MOPs with box constraints. That is, Ω_x can be written as $\Omega_x = \prod_{i=1}^{n} [l_i, u_i]$, where l_i and u_i are upper and lower bounds of x_i for all $i = 1, \ldots, n$.

More often than not, the objectives of the problem (2.1) are in conflict with each other. This means, any improvement in one objective will inevitably result in deterioration of another objective. Thus, no single solution exists that makes all the objectives reach their optima. Instead, the optimality of the problem (2.1) consists of a set of trade-off solutions (called Pareto-optimal set) that compromise all the objectives. Concepts related to Pareto optimality [46] are described in the next section.

2.1.1.2 Basic Concepts

Definition 2.1. *A solution x is said to dominate another solution y if x is not worse than y in all objectives and is better than y in at least one objective. This is denoted* $x \preceq y$.

Definition 2.2. *A solution* x^* *is said to be Pareto optimal if no another solution x in the decision space satisfies* $x \preceq x^*$.

Definition 2.3. *The Pareto-optimal set (PS) is a set of Pareto-optimal solutions, i.e.,*

$$PS = \{x \in \Omega_x | x \text{ is Pareto optimal}\}.$$

Definition 2.4. *The Pareto-optimal front (PF), the image of PS in the objective space, is defined as,*

$$PF = \{f(x) \in \Omega_f | x \in PS\}.$$

Definition 2.5. *The nondominated set (NS) of a set S is a subset of S, i.e.,* $NS \subset S$*, and consists of all the solutions that cannot be dominated by any other solution in S. NS is expressed as*

$$NS = \{s \in S | \nexists t \in S, t \preceq s\}.$$

The above definitions are basic concepts in the field of multiobjective optimization and will be frequently used throughout the thesis.

2.1.1.3 Test Suites

Many real-world optimization problems share various common features. Using artificial test suites as a representative of these common features would make it possible to assess different approaches in a much broader context than the usual empirical set for a single application. Besides, artificial test suites also allow to ease theoretical analysis of algorithms. So far, a number of test suites focusing on different types of features have been proposed. In the following, we briefly introduce several test suites that are used for continuous multiobjective optimization.

ZDT Test Suite ZDT [200] is one of the earliest test suites used in multiobjective optimization. This test suite contains six biobjective problems with different characteristics, such as convex or concave PFs, continuous or discontinuous PFs, and unimodal or multimodal PFs.

DTLZ Test Suite The DTLZ [48] test suite is developed to meet the need of scalable test problems where the number of objectives can be easily scaled up. This test suite has nine instances. DTLZ has various features, such as multimodality and disconnectivity. There are also two instances that have inequality constraints.

WFG Test Suite The WFG [82] test suite contains nine distinctive problems, each of which has one or several typical characteristics. This test suite is constructed in view of limitations of the ZDT and DTLZ test suites. In this test suite, more features like mixed PF shapes, degeneration of PFs, biases, deception, and dependencies between variables, are recommended to be used for performance assessment.

LZ Test Suite The LZ [120] test suite features nonlinear correlation between variables, resulting in problems having complicated PS shapes. In practice, LZ test problems can specify arbitrary PS shapes where dependencies between variables can be adjusted to control the difficulty of convergence. Following this idea, Zhang *et al.* [190] proposed a test suite of 23 test problems for *IEEE Congress on Evolutionary Computation (IEEE CEC2009)* competition, in which more characteristics like constraints and deceptive search spaces are recommended.

MOP Test Suite The MOP [127] test suite is an extension of ZDT and DTLZ but more difficult than its predecessors. It originally has seven problems, each of which has local attractors on the PF. Due to the existence of these local attractors, the MOP problems cannot be solved well by many existing approaches, such as NSGA-II [41] and MOEA/D [188]. Considering the lack of three-objective problems with local attractors in intermediate regions of the PF, Jiang *et al.* [95] have recently added two more instances to this test suite.

In addition to the above-mentioned test suites, there are also another kinds of test suites. The OKA [135] test suite is a set of problems having complicated PSs, which are constructed through a series of transformations. TYP-MOP [31] is a set of truly disconnected multiobjective problems where the true PF and PS are in the form of multiple disconnected segments and has been used to examine algorithms' ability to deal with disconnectivity. Cheng *et al.* [29] proposed a set of test problems for large-scale multi- and many-objective optimization.

2.1.2 Related EAs

2.1.2.1 Introduction

EAs belong to the class of randomized search heuristics that mimic evolution by natural selection. Multiobjective optimization EAs (MOEAs) extend the applicability of EAs to solving MOPs. Over the past twenty years, there have been increasing research interests in the design of MOEAs for solving MOPs. This is mainly because MOEAs have several advantages over other optimization methods like mathematical programming. First, MOEAs have low requirements on problem characteristics, e.g., differentiability, and they

can deal with large and complex search spaces. Second, MOEAs can be used in the situation that there are not enough computational resources in terms of money, time, or knowledge to construct a problem-dependent algorithm [15]. Third, MOEAs can provide a set of trade-off solutions close to the PF/PS in a single run, whereas other mathematical approaches like normal boundary intersection [39] can compute only one solution at a time.

It is widely accepted that if no preference information is provided, MOEAs are expected to reach the following main optimization goals when approximating the PF.

- The approximation should be as close to the PF as possible. In other words, MOEAs should provide good convergence performance.
- The approximation solutions should be well-diversified across the PF. That is, MOEAs should provide a good distribution of solutions.
- The approximation should cover the PF well. That means, MOEAs should spread solutions widely over the PF.

The first goal often refers to convergence whereas the second and the third are related to diversity. Therefore, convergence and diversity are widely used to assess and measure the performance of MOEAs. They are generally assumed to be conflicting in EMO, and designing MOEAs is to reach the balance between convergence and diversity. Apart from these goals, in practice MOEAs are also required to have low computational complexity. This makes much sense if there are very limited computational resources.

2.1.2.2 General MOEA Framework

A general MOEA framework is illustrated in Fig. 2.1. MOEAs start with an initial population of candidate individuals. Very often, the initial population is generated in a random manner. However, if we have some knowledge about the characteristics of a good solution, it is wise to use this information to create the initial population. The population evaluation provides solutions with the exact objective values. The environmental selection is intended to preserve the best solutions in terms of convergence and diversity for the next generational evolution. The population reproduction generates an offspring population. The population evaluation, environmental selection and population reproduction run in turn until the stopping criterion is met.

According to the selection strategies used in the environmental selection, MOEAs can be classified into different categories. The following sections provide a broad overview of several main categories of MOEAs proposed in the literature.

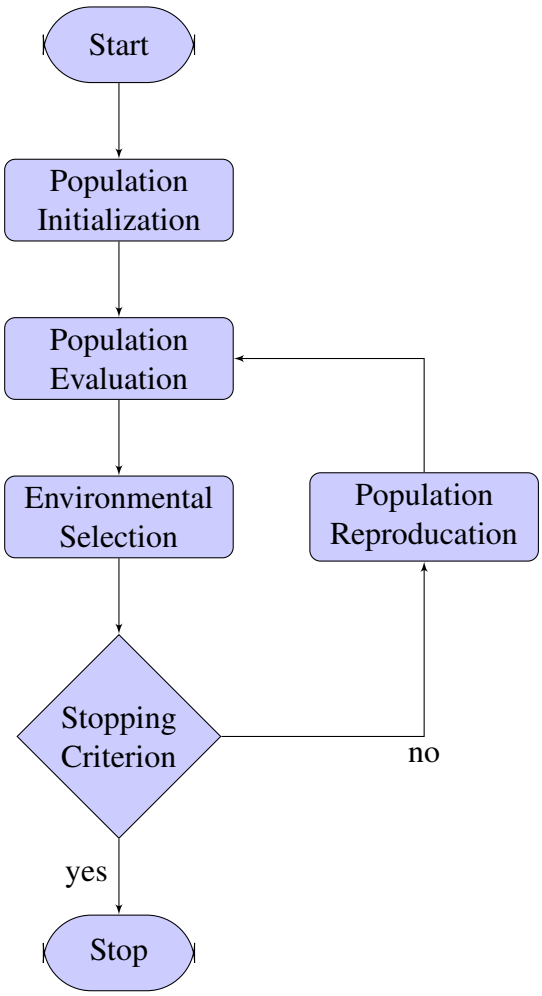


Fig. 2.1 The flowchart of general MOEAs.

2.1.2.3 Pareto-based Approaches

Pareto-based MOEAs employ the (weak) Pareto-dominance relation (i.e., "$\preceq$") [46], a kind of notion that defines a partial order in the objective space, to discriminate individuals in the population.

After decades of effort, a large number of Pareto-based MOEAs have been proposed. The multiobjective genetic algorithm (MOGA) [56] is generally considered the first MOEA that applies the concept of Pareto-based selection for mulitobjective optimization. Following the idea of MOGA [56], some popular MOEAs with Pareto-based selection emerged, including the niched Pareto genetic algorithm (NPGA) [80] and the nondominated sorting genetic algorithm (NSGA) [155]. These algorithms established the utility of MOEAs for solving MOPs. Then, new MOEAs, such as the strength Pareto evolutionary algorithm (SPEA) [203], Pareto envelope based evolutionary algorithm (PESA) [35] and the Pareto archived selection algorithm (PAES) [103], verified the importance of elitism, diversity maintenance and external archiving. These algorithms are commonly referred to as first-

generation. In the early 2000s, some second-generation algorithms were developed, and many of them are an updated version of their first-generation counterparts, e.g., NSGA-II [41], SPEA2 [202] and PEAS-II [36].

Pareto-based selection in MOEAs often has two stages. In the first stage, the population is ranked into different fronts. Solutions in the same front usually have the same rank value. The rank assignment can be conducted by dominance rank [155], dominance count [56] or dominance strength [202, 203]. In the second stage, each solution from the same font is assigned a density value. The density of solutions can be estimated by niching and fitness sharing [56, 80], gridding [103], or crowding distance [41]. Density information is used because it can help preserve solutions in sparse areas. Usually, solution ranking is considered the first criterion for solution selection, while solution density is the second criterion. The first criterion can promote population convergence, and the second one helps to maintain good population diversity.

2.1.2.4 Decomposition-based Approaches

Decomposition-based MOEAs, such as multiple single objective Pareto sampling (MSOPS) [83] and cellular multiobjective genetic algorithm (C-MOGA) [131], are a popular class of metaheuristics for EMO. They decompose an MOP into a number of subproblems[1] and simultaneously solve them in a collaborative manner. The MOEA based on decomposition (MOEA/D) [188] is a representative of this class of metaheuristics. MOEA/D decomposes an MOP by scalarizing functions (or termed decomposition approaches in some works [188]) into a set of subproblems, each of which is associated with a search direction (or weight vector) and assigned a candidate solution. In every generation, parents from a mating pool are selected to generate an offspring solution for each subproblem. Then, the offspring replaces certain existing solutions if it achieves better scalarizing values.

There are three popular scalarizing functions used for decomposition in decomposition-based approaches, which are listed as follows:

(a) Weighted Sum (WS) Method[188]

Assume that $w = (w_1, \ldots, w_m)^T$ is a weight vector where all components are non-negative and should satisfy $\sum_{i=1}^{m} w_i = 1$. The WS method defines the following single-objective problem:

$$\begin{array}{ll} \min & g^{ws}(x|w,z^*) = \sum_{i=1}^{m}(w_i|f_i(x) - z_i^*|) \\ s.t. & x \in \Omega_x. \end{array} \tag{2.2}$$

[1] Note that, subproblems can be not only single-objective optimization problems [188] but multiobjective ones [127].

If necessary, throughout the paper, $f_i(x) - z_i^*$ should be replaced by $(f_i(x) - z_i^*)/(z_i^{nadir} - z_i^*)$ where z_i^* and z_i^{nadir} are the i-th objective values of ideal point and nadir point found so far [188], respectively. The WS method can obtain a set of PF points by different weight vectors. The method can approximate the PF if it is convex, but will miss some PF points if the PF is nonconvex [188].

(b) Weighted Tchebycheff (TCH) Method[188]

The TCH method converts an MOP into a scalar problem in the following form:

$$\begin{aligned} \min \quad & g^{te}(x|w,z^*) = \max_{1\le i\le m}\left(\tfrac{1}{w_i}|f_i(x) - z_i^*|\right) \\ s.t. \quad & x \in \Omega_x, \end{aligned} \tag{2.3}$$

where $w_i = 10^{-4}$ is used in this method if $w_i = 0$. In Eq. (2.3), $1/w_i$ instead of w_i is adopted in order to obtain a set of uniformly-distributed solutions from a set of uniformly-distributed weight vectors [122]. The TCH method has the advantage in approximating nonconvex PFs compared with the WS method. It has been widely employed as a decomposition approach in MOEA/D variants [90, 120, 122, 175].

(c) Penalty-based Boundary Intersection (PBI) Method [188]

The PBI method converts an MOP into a scalar problem as follows:

$$\begin{aligned} \min \quad & g^{pbi}(x|w,z^*) = d_1 + \theta d_2 \\ s.t. \quad & x \in \Omega_x, \end{aligned} \tag{2.4}$$

where

$$d_1 = \frac{\|(f(x) - z^*)^T w\|}{\|w\|}, \tag{2.5}$$

$$d_2 = \|f(x) - (z^* + d_1 \frac{w}{\|w\|})\|. \tag{2.6}$$

In PBI, θ is a user-defined penalty factor. d_1 and d_2 are the length of the projection of vector $(f(x) - z^*)$ on the weight vector w and the perpendicular distance from $f(x)$ to w, respectively. θ is a key parameter for balancing convergence (measured by d_1) and diversity (measured by d_2). Recent studies [147] have shown that, when PBI approximates convex PFs, large diversity is likely to be obtained from minute and large θ values, and small θ values are beneficial to convergence.

The performance of MOEA/D was compared with the improved nondominated sorting genetic algorithm (NSGA-II) in [41] and [120] for MOPs with simple and complicated PSs, respectively, showing that MOEA/D is able to generate the best set of diverse nondominated solutions close to the PF in all tested cases. Furthermore, the efficiency of MOEA/D was confirmed by winning the unconstrained MOEA competition in the 2009

IEEE Congress on EC (IEEE CEC 2009) [190]. Since then, MOEA/D has attracted increasing research interest and various modified versions have been proposed in the literature [86, 127, 139]. Besides, the idea has also been integrated into hybrid algorithms [21, 102, 121, 153].

2.1.2.5 Indicator-based Approaches

The main idea behind indicator-based approaches is that quality indicators are able to quantify the quality of an approximate PF obtained. Indicator-based MOEAs often use quality indicators to guide the search, particularly in the process of environmental selection.

The indicator-based evolutionary algorithm (IBEA) [201] is the first implementation of indicator-based approaches. IBEA compares a pair of candidate solutions by an arbitrary indicator. As a result, high-quality solutions are preserved for further evolution. Later, Beume *et al.* [8] proposed a steady-state MOEA, called SMS-EMOA, based on the hypervolume indicator. In environmental selection, the hypervolume contribution of each candidate solution is computed, and the one with the least hypervolume contribution is excluded from the evolving population. Hyervolume-based MOEAs have received increasing research interests because hypervolume is the only indicator complying with Pareto dominance. However, a major drawback is that hypervolume is computationally demanding, particularly when the number of objectives is high, which is the case with many-objective optimization.

Apart from hypervolume, other indicators have also been successfully applied in indicator-based approaches. The averaged Hausdorff indicator [151] is used in a number of studies [141, 143] to replace hypervolume in the pursue of a computationally cheap indicator-based MOEA. The R2 indicator [18] has also been suggested to be in place of hypervolume in existing indicator-based MOEAs like SMS-EMOA.

2.1.2.6 Preference-based Approaches

Preference-based MOEAs originate from the fact that the number of Pareto optimal solutions may be very large or even infinite and the decision maker may be only interested in preferred solutions instead of the whole solutions. When preference information is provided, the search can be directed toward the region of interest to the decision maker.

Some early attempts on preference-based MOEAs are the studies of [56, 64, 157], in which preference information is used to rank the population. In [47], Deb *et al.* proposed a preference-based MOEA. In this method, the decision maker's performance information is used to model an approximate value function after every few generations of an MOEA. Then, the constructed value function is used to direct the search to more preferred solutions.

In a recent work, Wang *et al.* [172] have proposed to coevolve a family of preferences simultaneously with a population of candidate solutions, which leads to preference-inspired coevolutionary algorithms (PICEAs). Following this idea, they suggested a realization of PICEAs, called PICEA-g, and demonstrated that this method provides highly competitive performance for MOPs.

2.1.3 Performance Assessment

Performance measures are of vital importance to indicate whether an algorithm can achieve the optimization goals mentioned earlier. The following lists some widely used indicators in EMO.

2.1.3.1 Generational Distance

The generational distance (GD) metric [41] is one of commonly-used performance measures in EMO. It measures how close an approximation is to the true PF. Let PF be a set of uniformly distributed points in the true PF, and PF^* be an approximation of the PF. The GD is calculated as follows:

$$GD = \frac{(\sum_{i=1}^{n_{PF^*}} d_i^q)^{1/q}}{n_{PF^*}}, \tag{2.7}$$

where $n_{PF^*} = \|PF\|$, d_i is the Euclidean distance between the ith member in PF^* and its nearest member in PF. Very often, $q = 2$ is used.

2.1.3.2 Inverted Generational Distance

The inverted generational distance (IGD) metric in [183, 192] measures both the convergence and diversity of solutions obtained by an algorithm. The IGD is calculated as follows:

$$IGD = \frac{\sum_{i=1}^{n_{PF}} d_i}{n_{PF}}, \tag{2.8}$$

where $n_{PF} = \|PF\|$, d_i is the Euclidean distance between the ith member in PF and its nearest member in PF^*. To have a low IGD value, PF^* must be very close to PF and cannot miss any part of the whole PF.

2.1.3.3 Averaged Hausdorff Distance

Averaged Hausdorff distance (Δ_p) [151] is a recently developed metric that can somewhat handle the outlier tradeoff. The metric is calculated as follows:

$$\Delta_p(PF^*,PF) = \max\left\{\left(\frac{\sum_{x\in PF} d^p(x,PF^*)}{|PF|}\right)^p, \left(\frac{\sum_{x\in PF^*} d^p(x,PF)}{|PF^*|}\right)^p\right\}, \tag{2.9}$$

where $d(x, PF)$ is the distance between the member x of PF^* and the nearest member of PF, and $d(x, PF^*)$ is the distance between the member x of PF and the nearest member of PF^*. In this thesis, $p = 2$ is used.

2.1.3.4 Schott's Spacing

Schott [150] developed a metric with regard to the distribution of the discovered PF, called the spacing metric (S). S measures how evenly the members in a PF approximation (denoted PF^*) obtained by an algorithm are distributed, and is computed as:

$$\begin{aligned} S &= \sqrt{\frac{1}{n_{PF^*}-1}\sum_{i=1}^{n_{PF^*}}(D_i-\overline{D})^2} \\ \overline{D} &= \frac{1}{n_{PF^*}}\sum_{i=1}^{n_{PF^*}} D_i, \end{aligned} \tag{2.10}$$

where D_i is the Euclidean distance between the ith member and its nearest member in PF^*.

2.1.3.5 Maximum Spread

The maximum spread (MS), first introduced by Zitzler *et al.* [200], measures to what extent the extreme members (usually boundary points) in PF have been reached. Goh and Tan [60] proposed a modified version of MS by taking into account the proximity of PF^* towards PF:

$$MS = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left[\frac{min[\overline{PF_k}, \overline{PF_k^*}] - max[\underline{PF_k}, \underline{PF_k^*}]}{\overline{PF_k} - \underline{PF_k}}\right]^2}, \tag{2.11}$$

where $\overline{PF_k}$ and $\underline{PF_k}$ is the maximum and minimum of the kth objective in PF, respectively; Similarly, $\overline{PF_k^*}$ and $\underline{PF_k^*}$ is the maximum and minimum of the kth objective in PF^*, respectively.

2.1.3.6 Hypervolume

The hypervolume (HV) [203] metric measures the size of the objective space dominated by the approximated solution set S and bounded by a reference point $R = (R_1, \ldots, R_M)^T$ that is dominated by all points on the PF, and is computed by:

$$HV(S) = Leb(\bigcup_{x \in S}[f_1(x), R_1] \times \cdots \times [f_M(x), R_M]), \tag{2.12}$$

where $Leb(A)$ is the Lebesgue measure [98] of a set A.

Based on HV, an alternative indicator is defined as follows:

$$HVD = HV(PF) - HV(PF^*), \tag{2.13}$$

where HVD is called hypervolume difference. Thus, minimization of HVD is equivalent to maximization of HV.

2.2 Evolutionary Many-objective Optimization

2.2.1 Many-objective Optimization Problems

2.2.1.1 Problem Definition

In general, many-objective optimization problems (MaOPs) are an extension of MOPs. There is no agreed definition for MaOPs. However, one thing for sure is that MaOPs are often related to MOPs with more than three objectives, that is, $M > 3$ in Eq. (2.1). Due to the increase in the number of objectives, MaOPs bring about new challenges to MOEAs. A typical challenge is that dominance mentioned earlier becomes less effective and even unable to discriminate solutions. As a result, existing MOEAs specially designed for MOPs cannot induce sufficient selection pressure during the search. MaOPs are significantly different from MOPs, although both have the same mathematical description.

2.2.1.2 Test Suites

Test problems for many-objective optimization problems should have more than three objectives. It is often desirable that the problems are scalable in terms of the number of objectives, because this can allow a deep investigation into the impact of dimension increase. So far, DTLZ [48] and WFG [82] are top two most popular test suites in many-objective optimization.

There are also some other many-objective problems in the literature. Saxena *et al.* [148] adapted several DTLZ problems so as to generate redundant objectives which can evaluate dimensionality reduction techniques. Li *et al.* [115] proposed a test problem for easing the difficulty of visualization in many-objective optimization. In this problem, the performance of EAs on many objectives can be reflected by the distribution of solutions on a 2-dimensional decision space. However, this problem is hard to be generalized to any number of objectives.

Similar to the work of Li *et al.* [115], Ishibuchi *et al.* [84] also proposed to generate many-objective test problems in a two- or three-dimensional decision space. A main advantage of their test problems is that many-objective approximations can be easily as-

sessed in the decision space. However, these kinds of test problems fail to exactly exhibit the performance of objective vectors through the visualization of decision vectors.

2.2.2 Related EAs

2.2.2.1 Modified Dominance Based Approaches

In many-objective optimization, the Pareto-dominance relation between solutions becomes less discriminating for MaOPs as most solutions become incomparable or nondominated, and for over ten objectives, almost all the solutions are nondominated [87]. For a geometrical interpretation, the reader is referred to [99, 156]. As a consequence, the Pareto-dominance relation becomes of limited use for MaOPs, since it cannot induce sufficient selection pressure towards the PF.

A straightforward approach is to modify or develop the definition of the Pareto-dominance relation so as to increase the selection pressure. In an early attempt, a relaxed version of Pareto-dominance, known as ε-dominance, was proposed by Laumanns *et al.* [108] to combine both the convergence and diversity of solutions in a compact form. This modification makes it possible for Pareto-based MOEAs to strengthen the selection pressure among solutions and has shown to be very promising for MaOPs [68, 101, 167]. Other studies along this direction, such as cone ε-dominance [7], k-optimality [54], preference order ranking [51], fuzzy-dominance [54, 72], θ-dominance [186], and generalized Pareto-optimality [199], have also been shown to provide competitive results.

2.2.2.2 Decomposition Based Approaches

Decomposition-based MOEAs, such as multiple single objective Pareto sampling (MSOPS) [83] and MOEA/D [188], are originally designed for multiobjective optimization, but they have great potential to handle many-objective optimization. Taking MOEA/D for example, it can decompose an MaOP into a set of scalar subprolems. It maintains population diversity by a set of evenly-distributed weight vectors. This way, MOEA/D is capable of solving different types of optimization problems with varying degrees of success [59, 85, 90, 120, 188]. Besides, the decomposition-based idea has also been exploited in some recently-developed MOEAs, e.g., NSGA-III [43], MOEA based on dominance and decomposition (MOEA/DD) [112] and MOEA/D with a distance-based updating strategy [185], to maintain population diversity or control convergence for many-objective optimization.

2.2.2.3 Indicator-based Approaches

Indicator-based approaches are promising for many-objective optimization because solution selection is guided by quality indicators instead of Pareto dominance based selection

methods. For this reason, their search ability does not deteriorate severely in the presence of many objectives. In indicator-based approaches, the hypervolume indicator is always preferred to any other indicators due to its Pareto dominance compatibility. IBEA [201] based on this indicator was reported to obtain good results for many-objective optimization [167]. However, hypervolume-based approaches are very computationally expensive, which prevents their wide application in many-objective optimization. A number of attempts have been made to alleviate this shortcoming. Ishibuchi *et al.* [88] proposed to search for only a small number of representative solutions in the hope to reduce hypervolume computations in hypervolume-based approaches. Brockhoff and Zitzler [16] suggested to conduct objective reduction before the use of hypervolume-based approaches. Bader and Zitzler [6] proposed to use Monte-Carlo sampling methods to approximate hypervolume so that the computational complexity can be significantly reduced.

Despite those efforts to improve the efficiency of hypervolume-based approaches, it should be acknowledged that the efficiency gained is very limited, and most of hypervolume-based approaches are still computationally demanding. This has motivated increasing research in which other efficient quality indicators are explored in place of hypervolume. A typical example is the R2 indicator [18], which is weakly Pareto-dominance compatible and is very cheap in terms of computational efficiency. R2 has been successfully applied to solve MaOPs in a recent study [63].

2.2.2.4 Diversity Enhancement Approaches

Since Pareto dominance becomes less effective in producing selection pressure for many-objective optimization, it may be wise to de-emphasize convergence and pay more attention to diversity maintenance or diversity promotion [2, 117, 169]. This is because diversity enhancement can help to alleviate the loss of selection pressure. In [2], a diversity management operator was introduced to manage the activation/deactivation of diversity promotion on the crowding distance of NSGA-II [41]. Wagner *et al.* [167] reported a significant improvement on the convergence performance of NSGA-II after modifying the assignment of crowding distance values for boundary solutions. Recently, Li *et al.* [117] proposed a shift-based density estimation (SDE) strategy to increase selection pressure for MaOPs. For fitness assignment, SDE takes into account both the distribution and convergence information of solutions, and nondominated solutions with poor convergence are penalized. The empirical study in [117] showed a clear improvement for MOEAs incorporating this strategy.

2.2.2.5 Dimensionality Reduction Approaches

This kind of approaches (i.e., objective reduction) focuses on the reduction of the number of objectives [19, 30, 148, 154], which attempts to circumvent the problems of MaOPs

by means of identification and removal of redundant objectives. As a result, the reduced multiobjective problems can be solved effectively using existing MOEAs.

Dimensionality reduction approaches have three main advantages. First, they help to reduce the computational complexity of MOEAs when dealing with MaOPs. Second, they help the decision maker understand MaOPs better by identifying the redundant objectives. Third, they function like data preprocessing methods and can be very independent of the framework of MOEAs. As a result, they are very easy to be incorporated into any other MOEAs.

However, dimensionality reduction approaches are based on the assumption that there are redundant objectives in the MaOP under discussion. Therefore, this kind of approach is of limited use in the situation where all the objectives are completely conflicting and have equal importance.

2.2.3 Performance Assessment

2.2.3.1 Reference-point Based IGD

Due to the increase in the number of objectives, the PF of an MaOP is usually spread over a very large objective space. As a result, a small-sized reference set is no longer capable of representing the whole PF, which in turn affects the accuracy of reference set based performance metrics, e.g., GD and IGD, when evaluating MOEAs' performance. On the other hand, it is often undesirable to use a large-sized reference set as it will lead to other issues like space shortage and computational inefficiency. In view of this problem, Deb and Jain [43] proposed a reference-point IGD for performance assessment of many-objective optimization. In the reference-point IGD, the reference set is a set of points which are the intersections of the PF and predefined search directions.

However, the reference-point IGD depends largely on predefined search directions. If the used search directions are not sufficiently diversified, then any conclusions based on this performance metric will be biased and unreliable. Also, this metric is applicable only to reference-point based EAs.

2.2.3.2 R2 Indicator

The R2 indicator [17] is a recently-developed indicator at an attempt to replace hypervolume [203] for many-objective optimization. The R2 indicator is based on utility functions which map a vector to a scalar value in order to measure the quality of PF approximations. In R2, the Tchebycheff function is used most. Therefore, R2 is defined as:

$$R2(P,W,z^*) = \frac{1}{|W|}\sum_{w\in W} \min_{p\in P}\{\max_{1\le i\le M}\{w_i|z_i^* - p_i|\}\}, \tag{2.14}$$

where W is a given set of uniformly-distributed weight vectors and z^* is an utopian point. P is a PF approximation. In recent studies, R2 has been reported to have desirable properties, i.e., it is weakly monotonic, produces well-distributed solutions and can be computed in a fast manner.

2.2.3.3 Other Comparison Indicators

In recent years, there have been increasing research interests in diversity-based indicators for many-objective optimization. Li *et al.* [113] proposed the diversity comparison indicator (DCI) that employs a grid environment to assess both spread and uniformity. Later, the same authors proposed another relative indicator called performance comparison indicator (PCI) [114] to assess PF approximations obtained by many-objective algorithms. PCI constructs a reference set with approximation sets and then uses this reference set to aid quality assessment. The points in the reference set are divided into many clusters, and PCI estimates the minimum moves of solutions in the approximation sets to weakly dominate these clusters. Very recently, Wang *et al.* [170] have also developed a new diversity metric for many-objective optimization, which is an accumulation of the dissimilarity in the population.

2.3 Evolutionary Dynamic Multiobjective Optimization

2.3.1 Dynamic Multiobjective Optimization Problems

2.3.1.1 Problem Definition

There are many dynamic characteristics involved in dynamic MOPs (DMOPs), and different DMOPs may have different mathematical definitions. This thesis considers the DMOPs defined as:

$$\begin{aligned} min \quad & F(x,t) = (f_1(x,t),...,f_M(x,t))^T \\ s.t. \quad & \begin{cases} h_i(x,t) = 0, & i = 1,...,n_h \\ g_i(x,t) \geq 0, & i = 1,...,n_g \\ x \in \Omega_x, \quad t \in \Omega_t, \end{cases} \end{aligned} \tag{2.15}$$

where M is the number of objectives, n_h and n_g are the number of equality and inequality constraints, respectively. $\Omega_x \subseteq R^n$ is the decision space, t is the discrete time instance defined as $t = \frac{1}{n_t}\lfloor\frac{\tau}{\tau_t}\rfloor$ (where n_t, τ_t, and τ represent the severity of change, the frequency of change, and the iteration counter, respectively) and $\Omega_t \subseteq R$ is the time space. $F(x,t)$: $\Omega_x \times \Omega_t \rightarrow R^M$ is the objective function vector that evaluates the solution x at time t.

2.3.1.2 Classification of DMOPs

Having a sound and clear classification is of great importance not only for a better understanding of dynamism but also for an easier way to define or construct dynamic test problems. Thus, before developing dynamic multiobjective test problems, one should answer the question of what classes can be defined for them. Fortunately, there already have been two classification criteria proposed in the literature, and they can be roughly termed as effect-based and cause-based criteria, respectively.

(a) Effect-based Criterion

In most real-life MOPs, e.g. vehicle routing, the environments often change over time. According to the induced effects on the PF/PS, Farina *et al.* [55] classified dynamic environments into four different types:

- **Type I** - the PS changes over time while the PF remains stationary.
- **Type II** - both the PF and PS change over time.
- **Type III** - the PF changes over time while the PS remains stationary.
- **Type IV** - both the PF and PS remain stationary, though the objective functions or the constraints may change over time.

Farina *et al.* further noted that, when the environment is dynamic, more types of the above changes could occur simultaneously in the time scale.

(b) Cause-based Criterion

Tantar *et al.* [160] argued that effect-based criterion, although of undisputed importance, does not capture or does not describe where dynamic changes in DMOPs come from or the cause of the dynamic changes. Accordingly, they suggested four cases (originally termed order, but we use case here to avoid misleading interpretation) of dynamic environments:

- **Case 1** - the decision variables change over time.
- **Case 2** - the objective functions change over time.
- **Case 3** - the current values of the decision variables or the objective functions depend on their previous values.
- **Case 4** - parts of or the entire environments change over time.

2.3.1.3 Test Suites

As evolutionary dynamic multiobjective optimization (EDMO) is a quite new research field, there is a lack of standard test suites to facilitate broad investigations on EAs' performance. So far, some attempts have been made to meet the demand, thereby generating the following important test-beds:

FDA Test Suite The FDA [55] test suite is one of the most widely-used test-bed in EDMO. It is derived from a modification of ZDT [200] and DTLZ [48]. Dynamic characteristics involved in this test suite are time-dependent variations of PFs/PSs.

dMOP Test Suite Similar to FDA [55], the dMOP [61] test suite is also a modification of ZDT [200]. However, dMOP contains a special problem, i.e., dMOP3. dMOP3 can generate random selection of diversity-related variables. As a result, population diversity will drop dramatically when an environmental change occurs, making many EAs difficult to track the change quickly.

UDF Test Suite UDF [11] is exactly a counterpart of its static version, i.e., LZ's test suite [190]. That is, UDF is obtained by adding some dynamic components to LZ's test suite. This test suite features nonlinear, time-varying and monotonic variable linkages, and variables in each problem can have different amount of change. Besides, it also has problems that can induce uncertain PF variations whenever there is a change.

HE Test Suite The HE [76] test suite borrows the framework of ZDT [200], LZ [190] and WFG [82] and adds some dynamic features to these predecessors. Therefore, this test suite inherits isolated or deceptive properties, which are hard for EAs to handle.

SJY Test Suite SJY [93] is a recently developed test suite in which the number of objectives for each problem can be easily scaled up. This test suite proposes different kinds of PF-related functions. It can facilitate theoretical studies for evolutionary dynamic many-objective optimization.

GTA Test Suite GTA [58] is a test suite based on LZ's framework [120]. This test suite addresses dynamic properties that are rarely considered in existing test problems, such as time-varying fitness landscape modality, tradeoff connectedness and tradeoff degeneracy.

Apart from these, there are also several other kinds of dynamic test suites. Jin and Sendhoff [96] proposed a generic idea that can construct DMOPs by dynamically changing the weights used to aggregate different objectives of static MOPs, but they did not

provide clearly defined problems. Guan *et al.* [67] suggested that DMOPs can be the replacement of some objectives by new objectives during the evolution. Mehnen *et al.* [130] suggested DSW DMOPs and DTF problems. The T [81] test suite proposed by Huang *et al.* introduce some Type-IV problems where the current found PS may affect future PSs and/or PFs.

Despite fruitful contributions to DMOPs, no single test suite can capture all possible features in dynamic environments. Standard test suites are much in need to further studies and developments in EDMO.

2.3.2 Related EAs

In dynamic environments, it is often inefficient to restart the optimization process from scratch whenever a change occurs, although the restart strategy may be a good choice if the environmental change is considerably severe [14]. In the literature, various approaches have been proposed to handle environmental changes. They can be roughly categorized into the following groups, according to their diversity/convergence handling features.

2.3.2.1 Diversity Introduction

This approach aims to increase diversity after the detection of environmental changes. Diversity introduction can be achieved by the mutation of selected old solutions [32, 44] or random generation of some new solutions [44, 65]. According to [44], random generation is helpful for handling dynamic environments with severe changes but may misguide the search to new areas of the search space that is far from the current optima. In contrast, hybermuation [32] may be beneficial in environments with small changes but may cause evolutionary stagnation if environments change severely. However, diversity introduction approaches can be easily incorporated into existing MOEA frameworks.

2.3.2.2 Diversity Maintenance

Instead of generating diversity after environmental changes, diversity maintenance approaches seek to maintain diversity through the run. Amongst these, random immigration [25] is a popular technique that can be easily incorporated in MOEAs. Random immigrants are introduced into the evolving population at fixed intervals and only a small portion of the population is replaced. This way, population diversity can be always maintained at a high level so that the search process will not be affected too much by environmental changes.

2.3.2.3 Memory Approaches

In EDMO, an important issue is how to reuse past knowledge obtained so far to help EAs to adapt to the new environment quickly. Memory approaches are a kind of techniques that can improve the speed of convergence. One benefit of memory approaches is that solutions previously discovered can be reused for tracking future environmental changes, thereby saving computational time. In memory approaches, the open issue is when to retrieve and how to reuse memory. An early attempt along this direction is Wang and Li' work [171], where four memory-based schemes are proposed. Later, Goh and Tan [61] explored the use of memory to improve the tracking ability of EAs for DMOPs.

2.3.2.4 Prediction Approaches

When environmental changes follow a certain pattern, it is desirable to use prediction techniques to compute or at least guess the location of the new optima. Prediction approaches are widely used in EDMO because most test problems designed so far are kind of periodic and can be predicted. Prediction models, such as autoregression (AR) [196, 197], feedback prediction [71], and Kalman filter [132], have been successfully applied to EDMO. Usually, prediction approaches are used to predict gradient [105], population centroid [136, 196], and search direction [182]. When this information is available, population can be re-initialized close to the new optima. Prediction approaches are helpful to save computational resources and speed up the tracking process. However, these approaches may fail if environmental changes does not exhibit regular patterns.

2.3.2.5 Multi-population Approaches

Using multiple populations to solve dynamic optimization problems is not very new in evolutionary computation, particularly in the field of dynamic single-objective optimization [119]. Multiple populations can explore different search regions simultaneously so that they are able to track any change or emergence of new optimal solutions. An advantage of multi-population approaches is that one population can exploit the current optimal solution while the other populations are encouraged to explore the search space. Dynamic competitive-cooperative coevolutionary EA (dCOEA) [61] is a first attempt to apply multi-population approaches to deal with DMOPs.

2.3.3 Performance Assessment

EDMO is a relatively new research topic, so it has not yet had well-established performance metrics. Some performance metrics used in EDMO are actually adapted from their static counterparts. More specifically, they are an average of static metric values

over a number of time steps. These commonly used dynamic performance metrics are briefly described as follows.

2.3.3.1 Averaged IGD

Zhou *et al.* [196] modified the IGD metric and adopted the average of the IGD values in some (e.g., T_s) time steps over a run as a performance indicator for EDMO, which is computed as:

$$\overline{IGD} = \frac{1}{T_s}\sum_{t=1}^{T_s} IGD(t), \tag{2.16}$$

where $IGD(t)$ refers to the IGD metric at time instance t and is calculated just before the next change occurs.

2.3.3.2 HV Difference

Zhou *et al.* [197] suggested to use the hypervolume difference (HVD) to measure the quality of the found PF, HVD is defined as:

$$HVD(t) = HV(PF_t) - HV(PF_t^*). \tag{2.17}$$

However, when the true PF is unknown, the HVD cannot be used. Facing this limitation, Zeng [195] considered the maximum HV as a reference when measuring the quality of the found PF.

Camara *et al.* [23] used the HVD to define an accuracy measure when the true PF is known in advance, which is defined as

$$acc(t) = |HV(PF_t) - HV(PF_t^*)|, \tag{2.18}$$

where the equation ensures that the accuracy measure is always a positive value.

2.3.3.3 Stability

The effect of the changes in the environment on the accuracy of the algorithm can be quantified by the measure of stability that was introduced by Weicker [177] for dynamic single-objective optimization and adapted for EDMO by Camara *et al.* [22]. Stability is computed as:

$$stab(t) = max\{0, acc(t-1) - acc(t)\}, \tag{2.19}$$

where $acc(t)$ is defined in Eq. (2.18). A low stability value indicates better robustness performance.

2.3.3.4 Reactivity

In the meanwhile, Camara *et al.* [22] also took into account the response time for an algorithm to recover after a change in the environment, thus presenting a measure of reactivity based on the reactivity performance measure introduced by Weicker [177] for dynamic single-objective optimization. Reactivity measure determines how long it takes for an algorithm to reach a specified accuracy threshold ε, and is defined as:

$$react(t,\varepsilon) = min\{t' - t | t < t' < t_{max}, acc(t') \geq (1-\varepsilon)acc(t)\}, \tag{2.20}$$

where $acc(t)$ is defined in Eq. (2.18), and t_{max} is the maximum number of iterations or generations.

2.4 Summary

This chapter briefly reviews three main topics in the field of EMO, i.e., multiobjective optimization, many-objective optimization and dynamic multiobjective optimization. In each topic, the research progress regarding test environments, EAs, and performance measures is presented.

Chapter 3

EAs for MOPs with Complex Pareto Fronts

In real-world applications, the properties of MOPs can be very complex regarding their PFs. Some MOPs may have a discontinuous PF whereas others may have an extremely-shaped one. These irregular PF geometries affect much the distribution of solutions provided by EAs. In order for EAs to deliver maximum information about the irregular PF, it is desired that EAs are able to learn from the MOP to be optimized and finally provide uniformly-distributed solutions across the PF. In this chapter, we focus on MOEA/D [188] and equip it with additional strategies so that it can deal with MOPs with complex PFs.

The rest of the chapter is organized as follows. Section 3.1 is devoted to delivering related work and the incentive of this research work. In Section 3.2, the proposed method is presented. Section 3.3 presents some experimental studies with regard to the proposed method, followed by further investigation on the sensitivity of some key parameters. Section 3.5 summarizes the work and points out related future research directions.

3.1 Introduction

In recent years, there have been increasing research interests in improving MOEA/D for complex MOPs. As the uniformity of weight vectors plays a fundamental role in MOEA/D, some researchers [66, 139, 159] employ weight adjustment schemes to achieve the best approximation possible. However, the authors of [59] argued that an even or uniform distribution of weight vectors does not necessarily produce evenly-distributed solutions on the PF. Another feasible way is to develop effective decomposition approaches [85, 86, 189]. One of the most popular decomposition approaches is the weighted Tchebycheff method, as mentioned in previous chapter. It has been recently demonstrated that this approach cannot guarantee a good distribution of solutions even though uniformly-distributed weight vectors are provided [59]. To overcome this drawback, the authors of

[189] proposed a modified Tchebycheff approach based on normal boundary intersection (NBI) [39] to overcome the sensitivity to scales of the objectives, thereby expecting a good distribution of PF approximations. While it works well on bi-objective problems, this approach can not be easily extended to higher dimensional problems. Another interesting method for handling complex problems is to adopt an objective transform strategy [126] if the geometry of the PF is below the hyperplane $\sum_{i=1}^{M} f_i = 1$. The basic idea behind it is to identify the PF geometry, getting the transformed PF as close to this hyperplane as possible. The idea seems helpful. However, if there is noise in the approximate PF used in identifying the PF shape, this method will fail. Besides, the identification of the PF geometry itself is an optimization problem, which requires extra computational resources.

To overcome these shortcomings, an improved MOEA/D with a two-phase (TP) strategy and a niche-guided scheme, denoted MOEA/D-TPN, is proposed in this chapter. In MOEA/D-TPN, TP is conditional and will be activated when there are less boundary or extreme solutions than intermediate solutions in the population, which helps the algorithm to dedicate computational resources to finding boundary solutions, and the niche-guided scheme is used to perform the mating operation with parents in the less crowded regions, thereby avoiding duplicated solutions in the offspring. MOEA/D-TPN is tested on some existing and newly designed complex MOPs, showing better performance than its predecessor. Additionally, the performance of MOEA/D-TPN is also compared with two many-objective optimizers, mainly to see if MOEA/D-TPN is competitive in three-objective cases.

3.2 Proposed Method

In this section, the MOEA/D-TPN algorithm is presented. MOEA/D-TPN improves MOEA/D with two strategies: the TP strategy, which conditionally divides the whole optimization process into two phases, and the niche-guided strategy, which helps increase the population diversity in a more reliable way. The two strategies are described below.

3.2.1 Two-phase Optimization

As stated in [139], there are two main issues in MOEA/D. One is the nonuniformity of approximate solutions along the convex PF that has complex shapes with a sharp peak and long tail, where a small variation in one objective results in a large gap in another objective. In this case, MOEA/D offers dense solutions in the intermediate region of the PF and can hardly achieve well-distributed solutions in the extreme region of the PF although a set of uniform weight vectors is provided. Fortunately, MOEA/D is free from this drawback in concave MOPs [126]. Thus, it is natural to achieve uniform solutions for

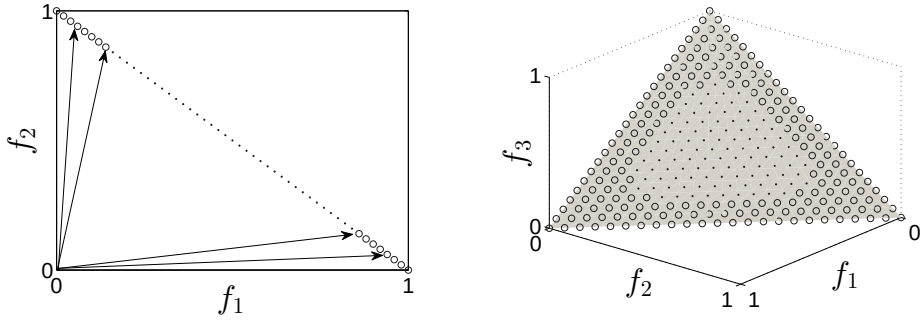


Fig. 3.1 Distribution of the extreme weight vectors (circles) and intermediate weight vectors (black dots): (a) the 2-objective case; (b) the 3-objective case.

a convex MOP by solving the scalar optimization subproblems in a reversed form

$$\begin{aligned} \max \quad & g^r(x|w,r^*) = \min_{1\le j\le M} w_j(r_j^* - f_j(x)) \\ \text{s.t.} \quad & x \in \Omega_x, \end{aligned} \tag{3.1}$$

where r^* is the nadir point constructed from the worst objective values for the entire PS, i.e., $r_j^* = max\{f_j(x)|x \in \Omega_x\}$ for each $j = 1,\ldots,M$.

When handling an MOP, however, its convexity-concavity is not always known beforehand. This gives rise to the problem on how to properly choose the subproblem form. Besides, the nadir point is also not available before the evolution. For these reasons, we propose the TP method that divides the whole evolution procedure into two phases. The first $M_r\%$ of the entire computing resources is called the first phase, where MOEA/D uses the scalar subproblem form of Eq. (**??**) and concentrates on convergence and diversity. At the end of the first phase, a crowding-based method is used to evaluate whether MOEA/D has obtained a set of uniform solutions or not. The crowding-based method assesses the density of solutions in both the intermediate region and the extreme region of the PF. If the solutions in the intermediate region is denser than those in the extreme region, it implies that the MOP is probably convex, and the use of the reversed scalar subproblem form will be more suitable in the remaining optimization phase. Note that the nadir point used in Eq. (3.1) for the second phase can be constructed by the obtained nondominated set (NS) from the first phase. The following illustrates how the crowding-based method works.

Any weight vector w used in MOEA/D should satisfy the restriction $w_1 + \cdots + w_M = 1$, where each ingredient w_j is non-negative. According to the Arithmetic Mean-Geometric Mean Inequality (AM-GM) theorem, $\prod_{j=1}^{M} w_j \le (\sum_{j=1}^{M} w_j/M)^M$ or simply $\prod_{j=1}^{M} w_j \le (1/M)^M$ always holds, and the equality condition of this inequality is that if and only if all ingredients of w are equal, which means w is located in the centre of the hyperplane $w_1 + \cdots + w_M = 1$. In other words, the larger $\prod_{j=1}^{M} w_j$ is, the closer the weight vector is

to the central weight vector. Therefore, we can categorize the set of N weight vectors employed in MOEA/D into two subsets, the intermediate subset W_m and the extreme subset W_e, as follows:

$$\begin{aligned} W_m &= \{w^i \in \Omega_w | \prod_{j=1}^{M} w_j^i \geq 0.5(\tfrac{1}{M})^M\} \\ W_e &= \Omega_w / W_m, \end{aligned} \tag{3.2}$$

where Ω_w is the set of N weight vectors generated by the simplex-lattice design used in the original MOEA/D. A value of 0.5 is used in Eq. (3.2) as it can generate a good classification between W_m and W_e. A too large or too small value of this parameter will cause size imbalance of the classified subsets. Fig. 3.1 shows examples in 2-D and 3-D cases.

Then, according to the weight vector subsets W_m and W_e, the population can be divided into two sub-populations, called the intermediate sub-population P_m and the extreme sub-population P_e, respectively. The crowdedness of the two sub-populations are estimated by

$$D_{mid} = \frac{1}{|P_m|} \sum_{i \in P_m} \gamma(i), \tag{3.3}$$

$$D_{ext} = \frac{1}{|P_e|} \sum_{i \in P_e} \gamma(i), \tag{3.4}$$

where D_{mid} and D_{ext} are the crowdedness values for the intermediate and extreme sub-populations, respectively. $\gamma(i)$ quantifies the crowding level of the ith solution in a sub-population, which is defined as follows:

$$\gamma(i) = \frac{1}{T} \sum_{j \in B(i)} d_{ij}, \tag{3.5}$$

where d_{ij} is the distance between the solutions i and j, $B(i)$ and T are the neighbouring members of solution i and the neighbourhood size in MOEA/D, respectively. The definition of $\gamma(i)$ is not trivial because it measures the closeness of the T neighbouring solutions to the ith solution.

It should be noted that the second phase is an alternative, which means if there is no much difference between D_{mid} and D_{ext}, the second phase can be eliminated from the evolution procedure, and the first phase will last for the whole evolution.

3.2.2 Niche-guided Mating/Update Selection

Another issue regarding MOEA/D is that it will produce many similar solutions due to the neighbourhood mating and updating strategy. The neighbourhood mating plays an important role in MOEA/D. In the original version of MOEA/D [188], the authors used

the simulated binary crossover (SBX) [42] operator to produce new solutions. Later in MOEA/D-DE [120], the same authors employed the DE operator [138], instead of SBX, to perform the mating operation and increased the population diversity by selecting three parent solutions from the whole population with a small probability $1-\delta$. This modification, however, can not guarantee that the selected solutions are distinct and may result in producing similar solutions in each generation. When an MOP has disconnected subregions on the PF, the effect is more severe and many scalar subproblems may obtain similar solutions on breakpoints, leading to a decrease in the population diversity. Thus, the selection of mating range is also of great importance to the performance of MOEA/D.

In the following, we propose a niche-guided scheme for the selection of the mating/update range. The scheme computes the niche count of each individual over its T neighbouring individuals instead of all members of the population. The niche count, $nc(i)$, for each individual i, is calculated by summing a sharing function over its T neighbouring individuals as:

$$nc(i) = \sum_{j=1}^{T} sh(d_{ij}), \tag{3.6}$$

where d_{ij} is the distance between individuals i and j, and $sh(d_{ij})$ is the sharing function that measures the similarity level between individuals i and j, which is defined as:

$$sh(d_{ij}) = \begin{cases} 1-(\frac{d_{ij}}{\sigma_{share}})^{\alpha}, & \text{if } d_{ij} \leq \sigma_{share} \\ 0, & \text{otherwise,} \end{cases} \tag{3.7}$$

where σ_{share} is a predefined niche radius and α is a constant, called the sharing level [62].

If the niche count of an individual is over a given threshold, it means that the individual is similar to its T neighbouring individuals and it is desirable to choose individuals outside the neighbourhood as the mating parents. Hence, the mating/update range can be defined as follows:

$$P'' = \begin{cases} P, & \text{if } nc(i) < \beta, \\ P', & \text{otherwise.} \end{cases} \tag{3.8}$$

where β is the threshold that is closely related to the niche radius σ_{share}, i.e., β is mainly determined by σ_{share}. As the maximum value of $nc(i)$ is T, we set β to be $T/2$ in this work. P and P' are calculated by

$$P = \begin{cases} B(i), & \text{if } rand_1 < \delta, \\ \{1,\ldots,N\}, & \text{otherwise.} \end{cases} \tag{3.9}$$

$$P' = \begin{cases} P, & \text{if } rand_2 < 0.5, \\ \{1,\ldots,N\}/B(i), & \text{otherwise.} \end{cases} \tag{3.10}$$

where $rand_1$ and $rand_2$ generate independently random numbers in the range $[0,1]$, P remains the same as in MOEA/D-DE [120]. The definition of P' means that, if the niche count of individual i reaches the threshold β, it has a 50% chance to select individuals outside its neighbourhood as the mating parents. This makes sense when MOEA/D pursues a high level of diversity but the neighbouring solutions are very similar.

3.2.3 The Framework of MOEA/D-TPN

For simplicity, we apply the DE operator [138] and polynomial mutation operator [46] to produce offspring in the proposed algorithm, which is the case with MOEA/D-DE [120]. The DE operator generates a candidate solution $\hat{y}$ by:

$$\hat{y_k} = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}), & \text{with probability } CR, \\ x_k^{r_1}, & \text{with probability } 1 - CR. \end{cases} \tag{3.11}$$

where $\hat{y_k}$ is the kth component of $\hat{y}$, and $x_k^{r_1}$, $x_k^{r_2}$, and $x_k^{r_3}$ are three distinct individuals randomly chosen from the population. CR and F are two control parameters.

The polynomial mutation produces a solution $y = (y_1, \ldots, y_n)$ from $\hat{y}$ as follows:

$$y_k = \begin{cases} \hat{y_k} + \sigma_k \times (u_k - l_k), & \text{with probability } p_m, \\ \hat{y_k}, & \text{with probability } 1 - p_m. \end{cases} \tag{3.12}$$

with

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}}, & \text{if } rand < 0.5, \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}}, & \text{otherwise.} \end{cases}$$

where $rand$ is a uniform random number from [0, 1]. The distribution index η and the mutation rate p_m are two control parameters. u_k and l_k are the lower and upper bounds of the kth decision variable, respectively.

The framework of MOEA/D-TPN is given in Algorithm 3.1. We would like to make the following remarks on the algorithm:

1. Since the distribution of the N weight vectors greatly affects the performance of MOEA/D, we use the WS-transformation scheme in [139] to generate a uniform spread of N weight vectors. Besides, as shown in Line 10 of Algorithm 3.1, at the beginning of the second phase, the N weight vectors should be reinitialized since the scalar subproblem form (i.e., the search direction) has changed. To obtain a uniform distribution of weight vectors for the reversed scalar subproblem form, we firstly use the simplex-lattice design to generate N points $\{w^1, \ldots, w^N\}$ on the hyperplane $w_1 + \cdots + w_M = 1$, then calculate the N search directions $\hat{w}^i (i = 1, \ldots, N)$ by $\hat{w}^i = (1 - w_1^i, \ldots, 1 - w_M^i)$. After that, the WS-transformation is applied on $\hat{w}^i (i =$

Algorithm 3.1: MOEA/D-TPN

Input: $MaxGen$: the stopping criterion
M_r: computational resources allocated for the first phase
N: the number of subproblems considered in MOEA/D
T: the neighbourhood size
δ: the probability for selecting neighbourhood
n_r: the maximum number of replaceable solutions
σ_{share}: the niche radius

Output: An approximated PF

Generate a uniform spread of N weight vectors: $w^1,\ldots,w^N$ and then compute the T closest weight vectors to each weight vector by the Euclidean distance. For each $i = 1,\ldots,N$, set $B(i) = \{i_1,\ldots,i_T\}$ where $w^{i_1},\ldots,w^{i_T}$ are the T closest weight vectors to w^i;
Generate an initial population $x^1,\ldots,x^N$ by uniformly randomly sampling from the decision space;
Set $FV^i := F(x^i)$, $EP_1 := \emptyset$, and $EP_2 := \emptyset$;
for $gen \leftarrow 1$ **to** $MaxGen$ **do**
    if $gen == M_r * MaxGen$ **then**
        Calculate D_{mid} and D_{ext} by Eqs. (3.3) and (3.4), respectively;
        if $D_{mid} < 0.9D_{ext}$ **then**
            Save the population in EP_1;
            Set the nadir point $r = \{r_1,\ldots,r_M\}$ where $r_j = max_{1\le i\le N} f_j(x^i)$;
            Re-initialize the N weight vectors for the second evolution phase and re-calculate the T closest weight vectors to each weight vector;
        end
    end
    for $i \leftarrow 1$ **to** N **do**
        Calculate the niche count of individual i and set the mating/update range as P'' by Eq. (3.8);
        Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P'';
        Apply the DE operator on individuals r_1, r_2 and r_3 by Eq. (3.11) to generate a solution $\bar{y}$, and perform the polynomial mutation operator on $\bar{y}$ by Eq. (3.12) to produce a new solution y;
        Update the reference point z if the evolution is in the first phase;
        Check if y is better than any individual x^j in P'' ($g(y|w^j,z) \le g(x^j|w^j,z)$ for the first phase and $g(y|w^j,r) \ge g(x^j|w^j,r)$ for the second phase). If y is better and no more than n_r individuals in P'' have been replaced, x^j is replaced by y;
    end
end
Save the final population in EP_2;
Output the non-dominated solutions from $EP_1 \cup EP_2$;

$1,\ldots,N$) to generate the working weight vectors. Due to the variation of the weight vectors, we have to recalculate the T closest weight vectors to each weight vector (Line 10).

2. In Line 5, we set the proportion of the computational resources allocated for the first evolution phase to be M_r. Intuitively, a large M_r gives the algorithm a large proportion of computational resources to converge toward the PF and makes the estimated nadir point (Line 9) more reliable. If M_r is too large, it may lead to the algorithm not having enough computational resources to execute the second-phase evolution. Besides, for easy-to-converge problems, $M_r = 0.5$ may be enough for estimating the crowdedness of the population as well as the nadir point. Accordingly, M_r is suggested to be in [0.5, 0.8]. In this work, M_r is set to 0.7 based on some preliminary experiments.

3. In some cases, the algorithm has already achieved a set of well-distributed solutions along the target PF at the end of the first phase, but D_{mid} may be slightly smaller than D_{ext} (Line 6), which gives an illusion that there are more solutions in the intermediate region than in the extreme region. To reduce the risk of being misled, D_{mid} is compared with 90% of the D_{ext} value when comparing the crowdedness of the two regions (Line 7).

4. Similar to MOEA/D-DE [120], the proposed algorithm adopts the DE operator to produce new solutions (Line 16). But, it selects parent individuals in a reliable way that guarantees a wide exploration when the neighbouring solutions are overcrowded. In other words, the niche-guided scheme helps maintain the population diversity.

5. In some sense, the second phase can be regarded as a stage of refining the already obtained solutions in the first phase since the second phase is expected to find a set of well-distributed solutions. Thus, in the second phase, instead of reinitializing the population, we use the final population of the first phase as the initial population. Besides, this will not interrupt the evolution process, which is quite important to the convergence performance of any MOEAs.

6. The algorithm uses two external population archives, EP_1 and EP_2, to store the approximated solutions in the two phases. Then, at the end of the whole evolution, nondominated solutions are identified from EP_1 and EP_2 and served as the final optimization results. Intuitively, EP_1 tends to keep intermediate solutions, while EP_2 is more likely to store boundary solutions.

3.2.4 Computational Cost of One Generation of MOEA/D-TPN

MOEA/D-TPN has the same framework as MOEA/D-DE [120], thus the increased computation cost is attributed to its detection step for the two-phase scheme and calculation of the niche count for niche-guided mating selection. The detection step (line 6 in Algorithm 1) requires $O(MNT)$ computations. Reinitialization (line 10 in Algorithm 1) would also requires $O(MNT)$. Calculation of the niche count for each individual (line 14 in Algorithm 1) requires $O(MT)$ computations. Other operations have smaller complexity. Therefore, the computational cost of MOEA/D-TPN in each generation is $O(MNT)$.

3.3 Experimental Studies

3.3.1 Test Problems and Performance Metrics

To investigate the performance of MOEA/D-TPN on problems with complex PF shapes, we develop some new test instances (F1-F4), each of which has a non-concave PF shape. Besides, two more commonly used test functions, UF4 [190] and convex DTLZ2 [43], are also included in the experiments. UF4 is a 2-objective problem with a complicated PS shape, whilst the convex DTLZ2 is adapted from the original DTLZ2 problem [49], whose Pareto-optimal surface is almost flat at the edges, but changes sharply in the intermediate region. Deb *et al.* [43] have shown in their experimental study that, on the convex DTLZ2, some MOEA/D variants fail to find points on the boundary of the PF. The details of these test instances are presented in Table 3.1.

In our experimental studies, we adopt two widely used metrics, i.e., IGD [204] and HV [203], for performance assessment. The reference point R is set to $(2.0, 2.0)^T$ for bi-objective test instances, which meets the requirement that R should be worse than the nadir point of the problems when calculating HV.

3.3.2 Parameter Settings

Four MOEA/D variants, i.e., MOEA/D, MOEA/D-N (i.e., MOEA/D with the niche scheme only), MOEA/D-TP (i.e., MOEA/D with the TP scheme only), and MOEA/D-TPN, are tested on the test instances. Note that, the term “MOEA/D" here refers to the MOEA/D-DE optimizer except that the WS-transformation method [139] is used for weight vector initialization. The control parameters for the compared algorithms were set the same as in [120]. In the last three variants, the sharing radius σ_{share} was set to 0.005, and the two-phase control parameter M_r was set to 0.7.

The population size in each algorithm was set to 200 for the 2-objective instances and 300 for the 3-objective instances. Each algorithm was executed 30 runs independently for

Table 3.1 Test Instances

Instance	Description	Domain	Notes
F1	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})^5$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2-\cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ PF: $f_2 = (1-\sqrt{f_1})^5$ PS: $x_i = \sin(0.5\pi x_i), i = 2,\ldots,n$	$[0,1]^n$ $n = 30$	Unimodal Convex Separable
F2	$f_1(x) = (1+g(x))(1-x_1)$ $f_2(x) = \frac{1}{2}(1+g(x))(x_1+\sqrt{x_1}\cos^2(4\pi x_1))$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2-\cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ PF: $f_2 = \frac{1}{2}(1-f_1+\sqrt{1-f_1}\cos^2(4\pi(1-f_1)))$ PS: $x_i = \sin(0.5\pi x_i), i = 2,\ldots,n$	$[0,1]^n$ $n = 30$	Multimodal Disconnected Separable
F3	$f_1(x) = (1+g(x))x_1$ $f_2(x) = \frac{1}{2}(1+g(x))(1-x_1^{0.1}+(1-\sqrt{x_1})^2\cos^2(3\pi x_1))$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2-\cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ PF: $f_2 = \frac{1}{2}(1-f_1^{0.1}+(1-\sqrt{f_1})^2\cos^2(3\pi f_1))$ PS: $x_i = \sin(0.5\pi x_i), \forall x_i \in \mathbf{x_{II}}$	$[0,1]^n$ $n = 30$	Multimodal Disconnected Separable
F4	$f_1(x) = (1+g(x))(\frac{x_1}{\sqrt{x_2x_3}})$ $f_2(x) = (1+g(x))(\frac{x_2}{\sqrt{x_1x_3}})$ $f_3(x) = (1+g(x))(\frac{x_3}{\sqrt{x_1x_2}})$ $g(x) = \sum_{i=4}^{n}(x_i-2)^2$ PF: $f_1f_2f_3 = 1$ PS: $x_i = 2, i = 3,\ldots,n$	$[1,4]^n$ $n = 30$	Unimodal Convex Separable
UF4	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}h(y_j)$ $f_2(x) = 1-x_1^2+\frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}h(y_j)$ $y_j = x_j - \sin(6\pi x_1+\frac{j\pi}{n}), j = 2,\ldots,n$ $h(y_j) = \frac{\lvert y_j\rvert}{1+e^{2\lvert y_j\rvert}}, j = 2,\ldots,n$ where $J_1 = \{j\mid j$ is odd and $2 \le j \le n\}$, $J_2 = \{j\mid j$ is even and $2 \le j \le n\}$ PF: $f_1^2+f_2 = 1$ PS: $x_j = \sin(6\pi x_1+\frac{j\pi}{n}), j = 2,\ldots,n$	$[0,1]\times$ $[-2,2]^{n-1}$ $n = 10$	Multimodal Concave Nonseparable
Convex DTLZ2	$f_1(x) = ((1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2))^4$ $f_2(x) = ((1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2))^4$ $f_3(x) = ((1+g(x))\sin(0.5\pi x_1))^2$ $g(x) = \sum_{i=3}^{n}(x_i-0.5)^2$ PF: $\sqrt{f_1}+\sqrt{f_2}+f_3 = 1$ PS: $x_i = 0.5, i = 3,\ldots,n$	$[0,1]^n$ $n = 10$	Unimodal Convex Separable

each test instance on a computer with a configuration of Intel(R) Core(TM)2 Duo CPU 2.4GHz processor and 4.00GB memory. The maximum number of generations was set to 500 for all the test instances.

3.3.3 Comparison Among MOEA/D Variants

Figures 3.2 and 3.3 plot the approximated PF achieved by each MOEA/D variant with the best IGD value on each test instance. F1 and F3 are problems with complex PF shapes of a sharp tail and a long tail. So, the use of TP evidently provides a good distribution in the extreme regions. F2 has several disconnected regions on the PF, and the original MOEA/D only finds a small part of solutions on the PF. Nevertheless, the use of the niche-guided scheme helps MOEA/D to solve this difficulty. It is also interesting that MOEA/D-TP also achieves a good approximation on this problem. One possible reason for this is that in the second phase, the weight vectors need to be re-initialized, which may cause the variation of search directions, and thus increase the population diversity. In Fig. 3.3, again, TP is very helpful in identifying boundary points for F4 and convex DTLZ2. UF4 is a concave problem, and the proposed TP scheme will not be activated in this case, which helps us focus analysis on the influence of the niche-guided scheme. Besides, UF4 also has strong dependencies between variables. Thus, to obtain a good approximation of this problem, an algorithm must be capable of maintaining population diversity. As shown in the middle column of Fig. 3.3, MOEA/D variants with the use of the niche-guided scheme have improved population diversity, thus achieving a better distribution of approximated solutions than those without this strategy. It can be also observed that TP is not activated in this case since MOEA/D-TPN and MOEA/D-N achieve similar results in Fig. 3.3 on UF4.

The plots of MOEA/D (which uses only WS-transformation) on F1, F3 and convex DTLZ2 in Figs. 3.2 and 3.3 also clearly indicate that a uniform distribution of weight vectors will not necessarily lead to a set of evenly distributed solutions on the PF. This may motivate researchers to seek other possible improvements like advanced decomposition approaches, instead of pursuing a better distribution of weight vectors, to enhance MOEA/D.

Table 3.2 shows the results regarding the IGD and HV metrics. Specially, reference points $(5,5,5)^T$ and $(2,2,2)^T$ are selected for the computation of the HV metric for the F4 and convex DTLZ2 test problems according to their nadir points, respectively. Best values are marked in boldface. For each instance, the two-sided Wilcoxon signed rank test [179] at the 95% confidence level is performed between MOEA/D-TPN and each of the other compared algorithms. '+', '-' or '*' denotes that the performance of the compared algorithm is significantly better than, worse than, or equivalent to that of MOEA/D-TPN, respectively, and they are marked on the median values. It is clear that MOEA/D-TPN

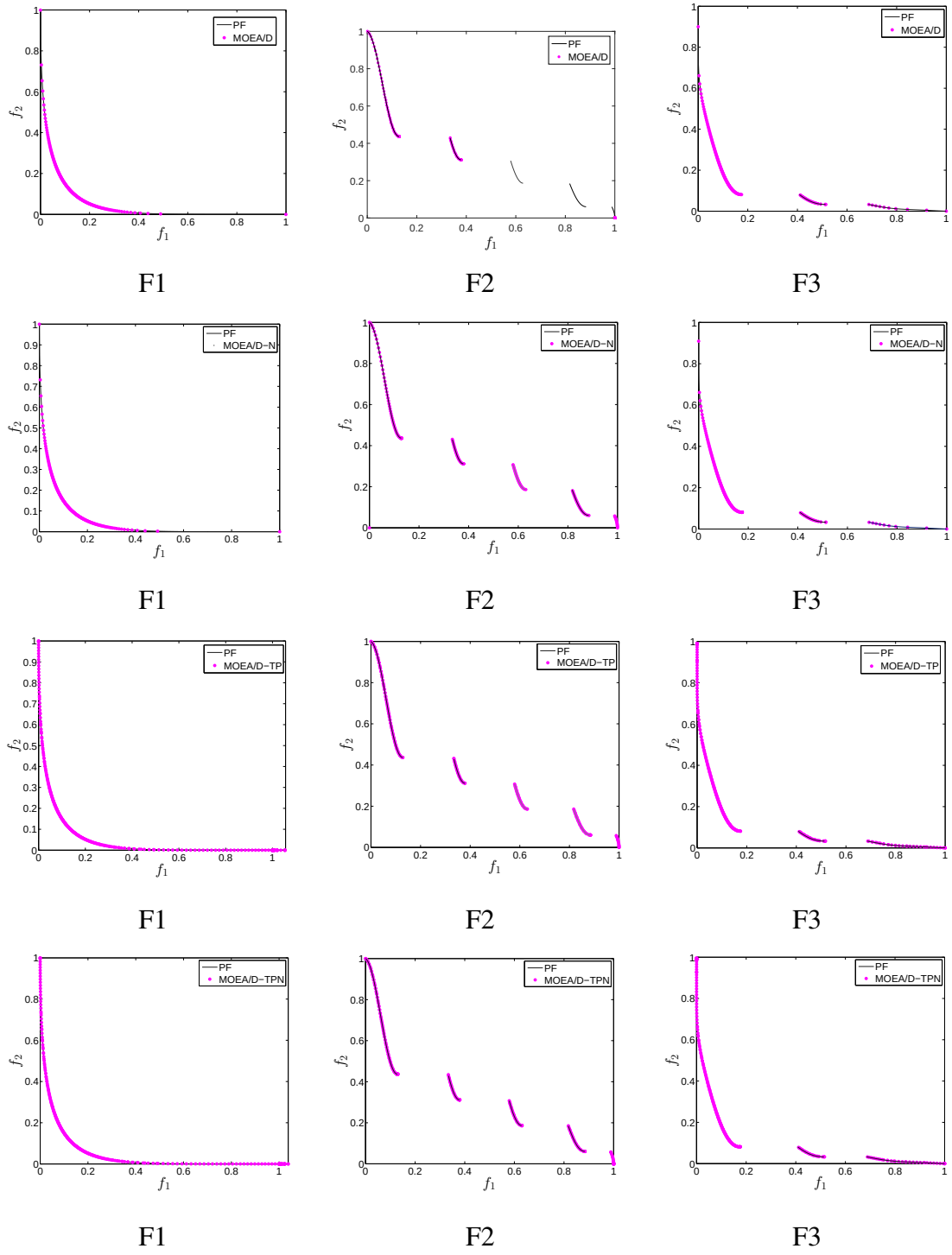


Fig. 3.2 PF approximations with the lowest IGD values among 30 runs on F1-F3.

performs significantly better than MOEA/D and MOEA/D-N on all the test instances, and performs similarly to MOEA/D-TP on F3, F4, UF4 and convex DTLZ2. This further validates the efficiency of the proposed algorithm in solving complex problems tested in this work.

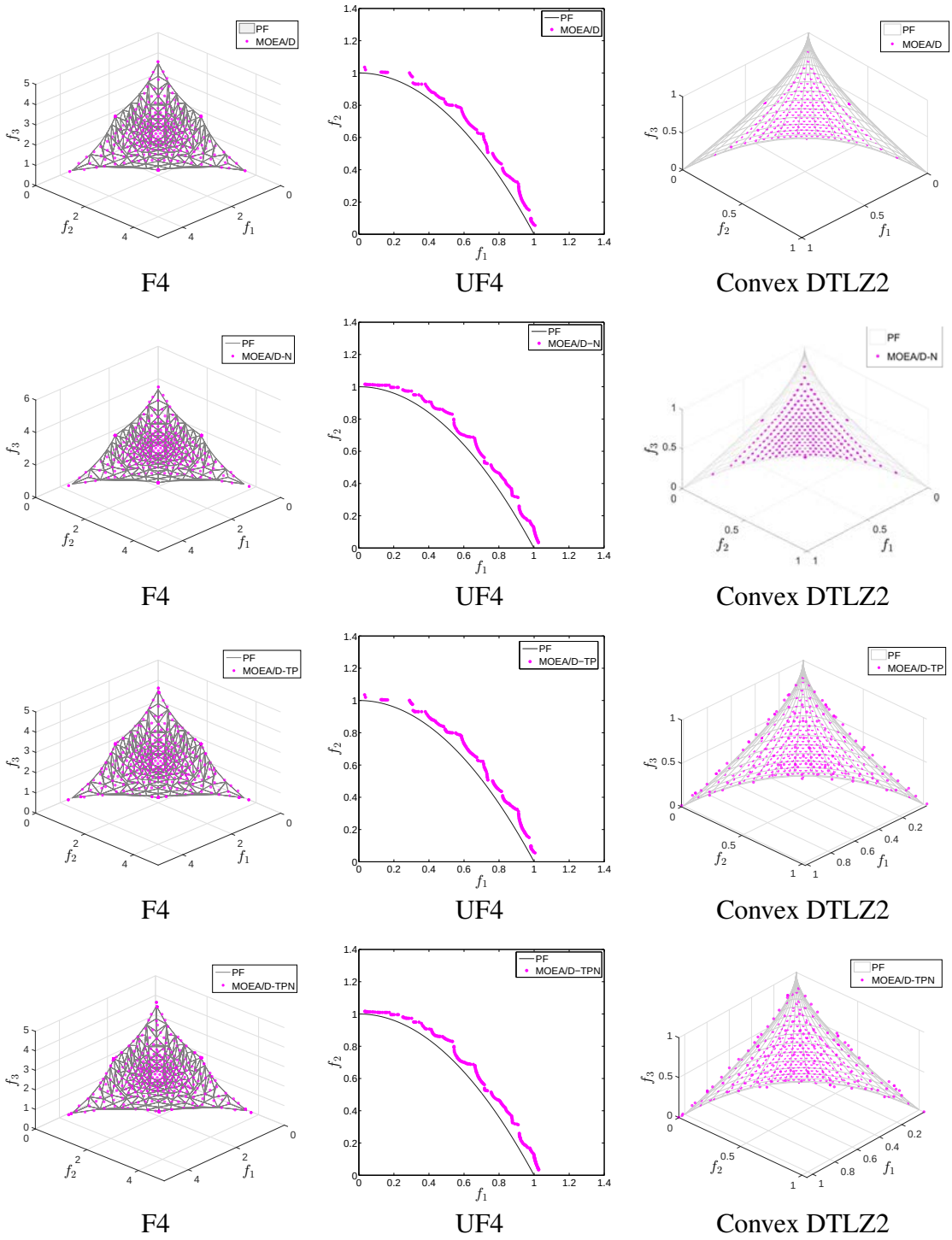


Fig. 3.3 PF approximations with the lowest IGD values among 30 runs on F4, UF4 and convex DTLZ2.

3.3.4 Comparison with Peer Algorithms

To have a fair comparison, peer algorithms, such as MOEA/D-NBI [189] and MOEA/D with objective transform (TMOEA/D) [126], that are specially designed for solving MOPs

Table 3.2 Best, median and worst IGD and HV values of the four algorithms on the test problems

Metric	Prob.	MOEA/D	MOEA/D-N	MOEA/D-TP	MOEA/D-TPN
IGD	F1	4.4032E-02	4.4105E-02	2.8330E-03	**2.8260E-03**
		(4.4287E-02)$^-$	(4.4278E-02)$^-$	(2.8465E-03)$^-$	**2.8455E-03**
		4.4504E-02	4.4482E-02	2.8670E-03	**2.8610E-03**
	F2	2.2880E-03	2.2830E-03	1.6630E-03	**1.6480E-03**
		(2.2930E-03)$^-$	(2.2890E-03)$^-$	(1.6995E-03)$^-$	**1.6780E-03**
		1.5622E-01	2.3370E-03	1.5570E-01	**1.8000E-03**
	F3	1.1038E-02	1.0936E-02	2.2620E-03	**2.2344E-03**
		(1.1181E-02)$^-$	(1.1245E-02)$^-$	(**2.3545E-03**)*	2.3835E-03
		1.1739E-02	1.3015E-01	**2.4950E-03**	2.6090E-03
	F4	8.3978E-02	8.4353E-02	7.5811E-02	**7.1559E-02**
		(8.4398E-02)$^-$	(8.4962E-02)$^-$	(**7.6760E-02**)*	7.6872E-02
		8.5074E-02	8.5784E-02	7.8385E-02	**7.8119E-02**
	UF4	3.6898E-02	3.5806E-02	3.6853E-02	**3.5704E-02**
		(4.2410E-02)$^-$	(4.2400E-02)$^-$	(**4.2323E-02**)*	4.2361E-02
		4.8510E-02	4.8282E-02	4.8409E-02	**4.8264E-02**
	Convex DTLZ2	9.1408E-02	8.9932E-02	**2.8870E-02**	2.9198E-02
		(9.3433E-02)$^-$	(9.3486E-02)$^-$	(3.3324E-02)*	**3.2598E-02**
		9.6259E-02	9.5988E-02	3.5477E-02	**3.5458E-02**
HV	F1	3.9568	3.9564	3.9568	**3.9592**
		(3.9508)$^-$	(3.9500)$^-$	(3.9496)$^-$	**3.9536**
		3.9432	3.9400	3.9424	**3.9436**
	F2	3.6968	3.6964	3.6968	**3.6986**
		(3.6744)$^-$	(3.6734)$^-$	(3.6742)$^-$	**3.6774**
		3.5120	3.6588	3.5184	**3.6608**
	F3	3.9224	3.9272	3.9252	**3.9280**
		(3.9146)$^-$	(3.9176)$^-$	(3.9195)*	**3.9198**
		3.9012	3.8724	**3.9032**	3.9024
	F4	99.8375	100.1625	**100.9213**	100.9125
		(98.9938)$^-$	(98.6813)$^-$	(**99.9688**)*	99.8337
		98.2250	98.1250	99.0000	**99.0250**
	UF4	3.2328	3.2272	3.2312	**3.2416**
		(3.1866)$^-$	(3.1932)$^-$	(3.1934)*	**3.1946**
		3.1524	3.1536	3.1512	**3.1604**
	Convex DTLZ2	7.9096	7.9232	7.9688	**7.9696**
		(7.8932)$^-$	(7.9020)$^-$	(7.9544)*	**7.9572**
		7.8696	7.8856	7.9400	**7.9472**

with complex PF geometries should be tested in our experiments. Since MOEA/D-NBI is only applicable to bi-objective cases, we choose F1 to F3, and UF4 as the test instances in this part. The parameter settings of each algorithm are derived from the referenced paper.

The statistical results of the three algorithms are presented in Table 3.3. It can be clearly seen from the table that MOEA/D-TPN significantly outperforms the other two algorithms on the four tested problems in terms of IGD and HV. For TMOEA/D, the poor performance can be attributed to that the shapes of the test instances are extremely irregular, and TMOEA/D may not be able to accurately identify them, thus it cannot achieve a good approximation. For MOEA/D-NBI, apart from the extremely irregular PF geometries of the test problems, the lack of effective diversity maintenance strategy may be another reason for its poor performance. Nevertheless, MOEA/D-TPN considers the distribution of solutions and uses TP to adjust search directions. Accordingly, it achieves a better approximation for each instance. To have a better understanding of differences among the three different kinds of algorithms, we also plot the approximations obtained by TMOEA/D and MOEA/D-NBI for F1, F2 and UF4 in Fig. 3.4.

3.3.5 Comparison with Other Algorithms

In this subsection, SPEA2+SDE [117] and NSGA-III [43] are used for comparison. The algorithms compared are newly-developed techniques and have shown to be very promising for solving MOPs. To increase difficulties for these algorithms, four extra test problems with complex PF shapes are used in the experiment, which are presented in Table 3.4. POL is derived from Poloni's study [137], whose PF is discontinuous and has a long tail. mF4 is a modified version of F4, and is more difficult for MOEAs to find boundary solutions on the PF since both the search space and the objective space have been scaled. The modification leads to the variation of the objective scale, i.e., $f_i \in [0.1, 10], i = 1, 2, \ldots, M$. F5 has an irregular PF shape and poses difficulties for MOEAs to approximate extreme regions. F6 is a multi-modal convex problem and has deceptive properties. It challenges algorithms in finding a global PF as well as extreme regions. In other words, the four test functions from Table 3.4 are more challenging than those used in our previous experiment, imposing various effects on the performance of MOEAs. Specially, reference points $(20, 30)^T$, $(12, 12, 12)^T$, and $(2, 2, 2)^T$, which should be slightly worse than the nadir point, are selected for the HV calculation of POL, mF4, and another three-objective problems, respectively. The parameter settings in each algorithm are derived from the referenced paper.

Table 3.5 gives the statistical results of the IGD, HV and T values on these test functions, where "T" represents the run-time of an algorithm, recorded in seconds. It can be observed that, for the tested problems, MOEA/D-TPN achieves significantly better results than the other two algorithms in terms of IGD and HV. Specifically, judging from the IGD values, SPEA2+SDE struggles to converge toward the PF on mF4 and F5, while NSGA-III performs poorly on POL and F6. As for the run-time performance, MOEA/D-TPN

Table 3.3 Best, median and worst IGD and HV values of the peer algorithms on four test problems

	IGD			HV		
Prob.	TMOEA/D	MOEA/D -NBI	MOEA/D -TPN	TMOEA/D	MOEA/D -NBI	MOEA/D -TPN
F1	3.4720E-03	2.8690E-03	**2.8260E-03**	3.9592	3.9563	**3.9592**
	(3.8615E-03)$^-$	(2.8720E-03)$^-$	**2.8455E-03**	(3.9514)$^-$	(3.9496)$^-$	**3.9536**
	4.5490E-03	2.8800E-03	**2.8610E-03**	3.9432	3.9428	**3.9436**
F2	2.9960E-03	**1.6410E-03**	1.6480E-03	3.6960	3.7008	**3.6986**
	(4.2245E-03)$^-$	(1.9650E-03)$^-$	**1.6780E-03**	(3.6720)$^-$	(3.6744)$^-$	**3.6774**
	1.8772E-02	1.5566E-01	**1.8000E-03**	3.6532	3.5316	**3.6608**
F3	3.4610E-03	2.4550E-03	**2.2344E-03**	3.9280	3.9272	**3.9280**
	(4.3285E-03)$^-$	(2.6590E-03)$^-$	**2.3835E-03**	(3.9174)$^-$	(3.9174)$^-$	**3.9198**
	1.3451E-02	2.9640E-03	**2.6090E-03**	3.8996	3.9016	**3.9024**
UF4	4.9577E-02	3.8695E-02	**3.5704E-02**	3.2136	3.2402	**3.2416**
	(6.4952E-02)$^-$	(4.2808E-02)$^-$	**4.2361E-02**	(3.1356)$^-$	(3.1894)$^-$	**3.1946**
	8.7681E-02	4.9567E-02	**4.8264E-02**	3.0544	3.0812	**3.1604**

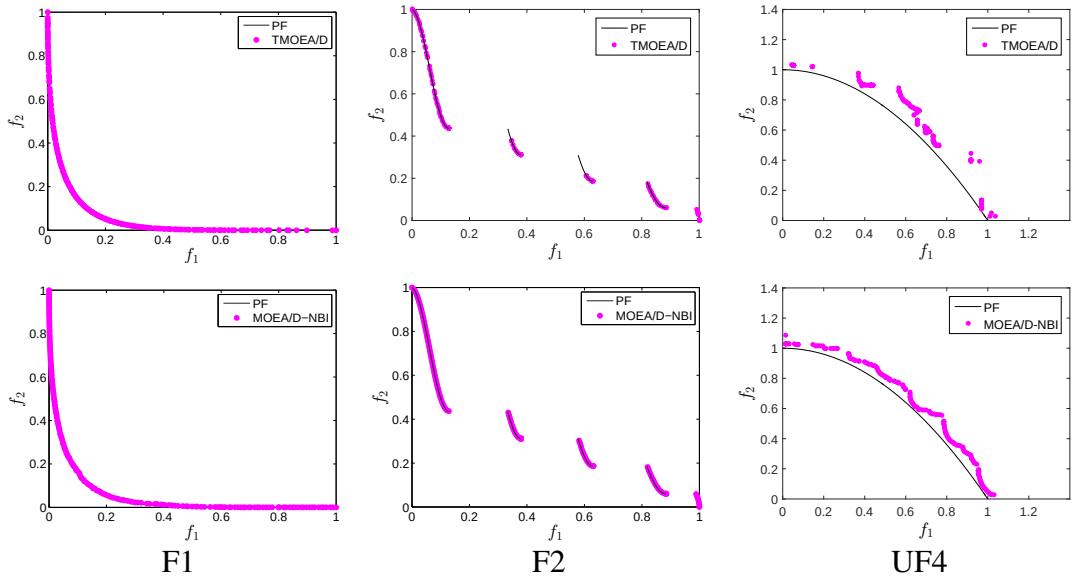


Fig. 3.4 PF approximations with the lowest IGD values among 30 runs on F1, F2 and UF4.

clearly exceeds the other two algorithms by a significant margin, SPEA2+SDE ranks the second, and NSGA-III ranks the last.

Fig. 3.5 shows the approximate PF of the lowest IGD value over 30 runs obtained by each algorithm. It is clear to see that MOEA/D-TPN is able to converge to the PFs and cover the entire PFs of the first three test functions, and finds most of solutions for F6. On F6, SPEA2+SDE loses the boundary areas, while MOEA/D-TPN keeps most of them. However, MOEA/D-TPN loses extreme areas, while SPEA2+SDE keeps well.

Table 3.4 Extra Test Instances

Instance	Description	Domain	Notes
POL	$f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2$ $A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2)$ $A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2)$ $B_1 = 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2)$ $B_2 = 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2)$	$[-\pi, \pi]^n$ $n = 2$	Unimodal Non-convex Nonseparable
mF4	$f_1(x) = (1 + g(x))(\frac{x_1}{\sqrt{x_2 x_3}})$ $f_2(x) = (1 + g(x))(\frac{x_2}{\sqrt{x_1 x_3}})$ $f_3(x) = (1 + g(x))(\frac{x_3}{\sqrt{x_1 x_2}})$ $g(x) = \sum_{i=4}^{n} (x_i - 5)^2$ PF: $f_1 f_2 f_3 = 1$ PS: $x_i = 5, i = 3, \ldots, n$	$[1, 10]^n$ $n = 30$	Unimodal Convex Separable
F5	$f_1(x) = (1 + g(x))\,((1 - x_1)x_2)$ $f_2(x) = (1 + g(x))\,(x_1(1 - x_2))$ $f_3(x) = (1 + g(x))\,(1 - x_1 - x_2 + 2x_1 x_2)^6$ $g(x) = \sum_{i=3}^{n} (x_i - 0.5)^2$ PF: $f_3 = (1 - f_1 - f_2)^6$ PS: $x_i = 0.5, i = 3, \ldots, n$	$[0, 1]^n$ $n = 30$	Unimodal Convex Nonseparable
F6	$f_1(x) = \cos^4(0.5\pi x_1)\cos^4(0.5\pi x_2)$ $f_2(x) = \cos^4(0.5\pi x_1)\sin^4(0.5\pi x_2)$ $f_3(x) = \left(\frac{1+g(x)}{1+\cos^2(0.5\pi x_1)}\right)^{\frac{1}{1+g(x)}}$ $g(x) = \frac{1}{10}\sum_{i=3}^{n} (1 + x_i^2 - \cos(2\pi x_i)))$ PF: $f_3(1 + \sqrt{f_1} + \sqrt{f_2}) = 1$ PS: $x_i = 0, i = 3, \ldots, n$	$[0, 1]^n$ $n = 30$	Multimodal Convex Separable

While SPEA2+SDE performs well on POL and F6, it struggles to converge to the PFs of mF4 and F5. NSGA-III achieves fair results on POL, mF4 and F5, but it fails to cover the whole extreme regions of these problems. NSGA-III performs poorly and obtains many points on a local PF surface on F6 since this is a deceptive problem with a complex PF, but the other algorithms have successfully found many global solutions on this problem.

Fig. 3.6 shows the evolution curves of the average IGD values for the test problems against generation. Clearly, all the algorithms have a fast convergence performance on the test functions. Meanwhile, a notable improvement of the IGD values can be observed in MOEA/D-TPN when TP is activated (at the 350th generation), leading to MOEA/D-TPN achieving better late-stage performance than SPEA2+SDE and NSGA-III on these problems.

Table 3.5 Best, median and worst IGD, HV and T (seconds) values of the three algorithms on the extra test problems

Metric	Prob.	SPEA2+SDE	NSGA-III	MOEA/D-TPN
IGD	POL	7.9373E-02	1.4470E-01	**5.6305E-02**
		(1.0496E-01)$^-$	(2.0711E-01)$^-$	**5.9582E-02**
		1.7027E-01	2.2150E-01	**6.1822E-02**
	mF4	1.4703E+00	1.4805E-01	**1.2396E-01**
		(1.4738E+00)$^-$	(1.5435E-01)$^-$	**1.2535E-01**
		1.4783E+00	1.6239E-01	**1.2754E-01**
	F5	1.8505E-01	2.5032E-02	**1.3646E-02**
		(1.8536E-01)$^-$	(2.7036E-02)$^-$	**1.4040E-02**
		1.8613E-01	2.9269E-02	**1.4545E-02**
	F6	3.3742E-02	4.4938E-01	**1.6892E-02**
		(4.8208E-02)$^-$	(5.1625E-01)$^-$	**1.9734E-02**
		6.6152E-02	5.4934E-01	**2.4442E-02**
HV	POL	5.3820E+02	5.3535E+02	**5.3832E+02**
		(5.3589E+02)$^-$	(5.3535E+02)$^-$	**5.3646E+02**
		5.3220E+02	5.3106E+02	**5.3256E+02**
	mF4	1.4285E+03	1.6610E+03	**1.6665E+03**
		(1.4160E+03)$^-$	(1.6533E+03)$^-$	**1.6599E+03**
		1.4030E+03	1.6440E+03	**1.6520E+03**
	F5	7.7864E+00	7.9840E+00	**7.9832E+00**
		(7.7364E+00)$^-$	(7.9760E+00)$^-$	**7.9770E+00**
		7.6664E+00	7.9704E+00	**7.9714E+00**
	F6	6.0536E+00	3.8400E+00	**6.0656E+00**
		(5.9720E+00)$^-$	(3.7672E+00)$^-$	**6.0012E+00**
		5.8832E+00	3.7000E+00	**5.9440E+00**
T	POL	2.0201E+02	3.9143E+02	**6.9348E+01**
		(2.0550E+02)$^-$	(4.2283E+02)$^-$	**7.3065E+01**
		2.2123E+02	6.2033E+02	**8.1042E+01**
	mF4	3.5728E+02	4.5907E+02	**7.5071E+01**
		(3.6065E+02)$^-$	(5.7852E+02)$^-$	**8.6718E+01**
		4.2019E+02	6.4615E+02	**9.4585E+01**
	F5	4.5921E+02	6.0510E+02	**7.2329E+01**
		(4.7347E+02)$^-$	(6.1632E+02)$^-$	**8.0427E+01**
		5.1030E+02	7.9555E+02	**9.1976E+01**
	F6	4.3947E+02	4.8475E+02	**7.7635E+01**
		(4.5637E+02)$^-$	(5.0677E+02)$^-$	**7.8930E+01**
		4.7875E+02	6.7858E+02	**9.9403E+01**

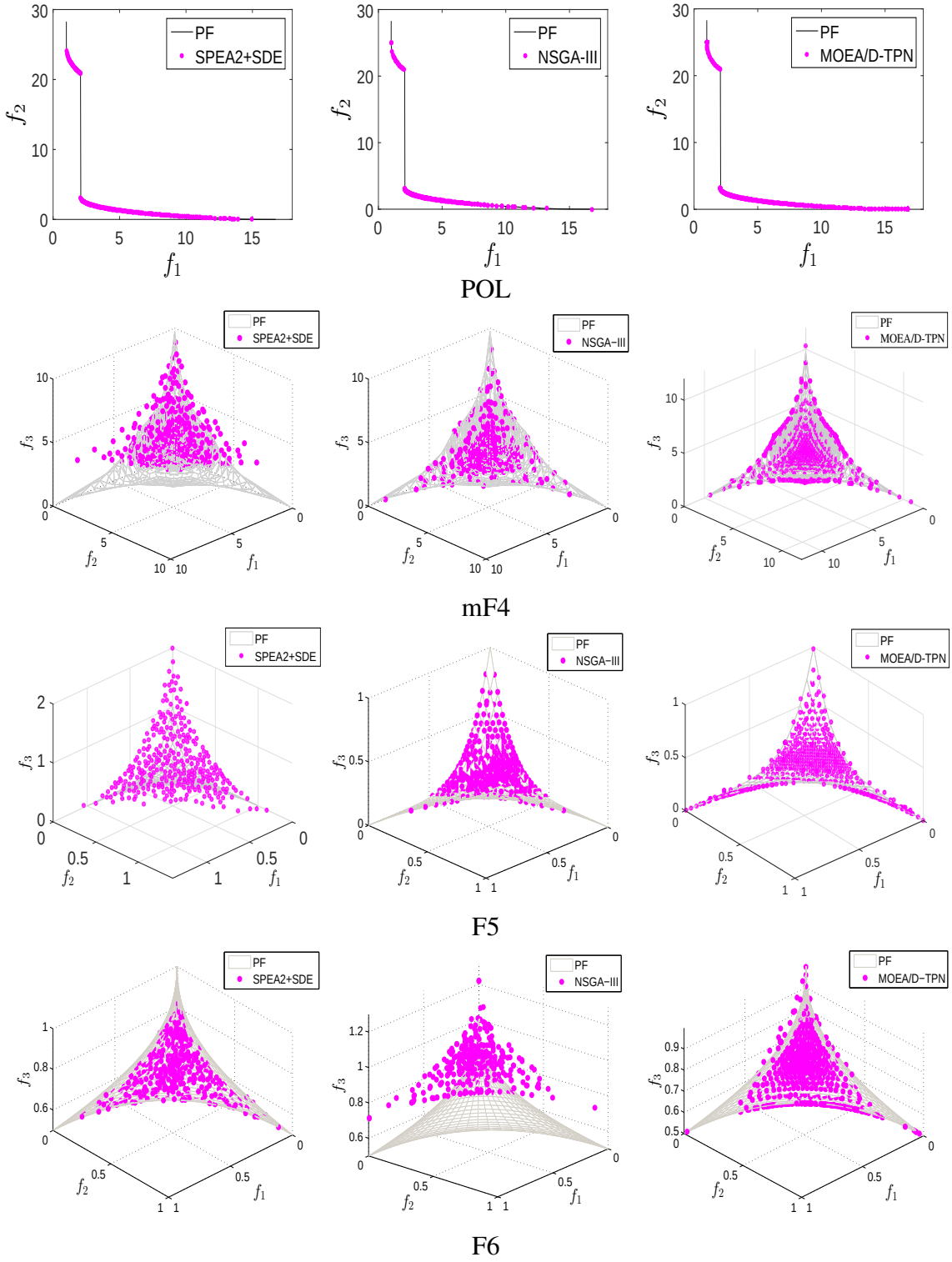


Fig. 3.5 PF approximations with lowest IGD values among 30 runs on POL, mF4, F5 and F6.

3.4 Sensitivity Analysis

3.4.1 Further Investigation of TP

As has been shown in the previous experimental study, the TP scheme evidently improves MOEA/D by providing a better set of evenly-distributed solutions for the tested problems.

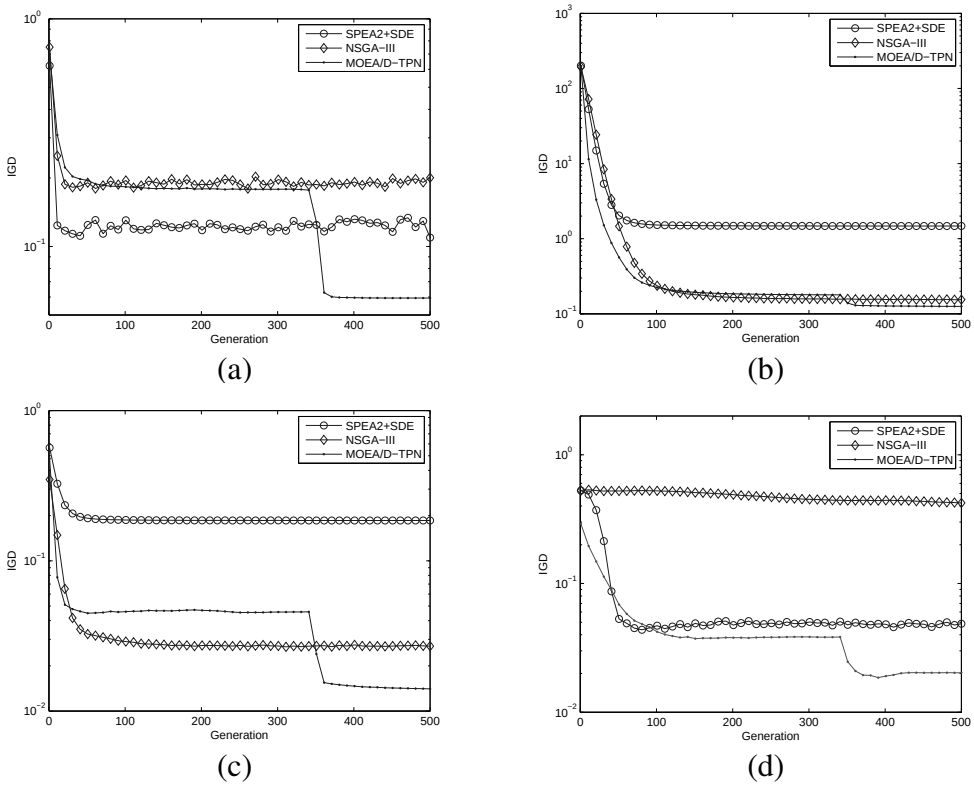
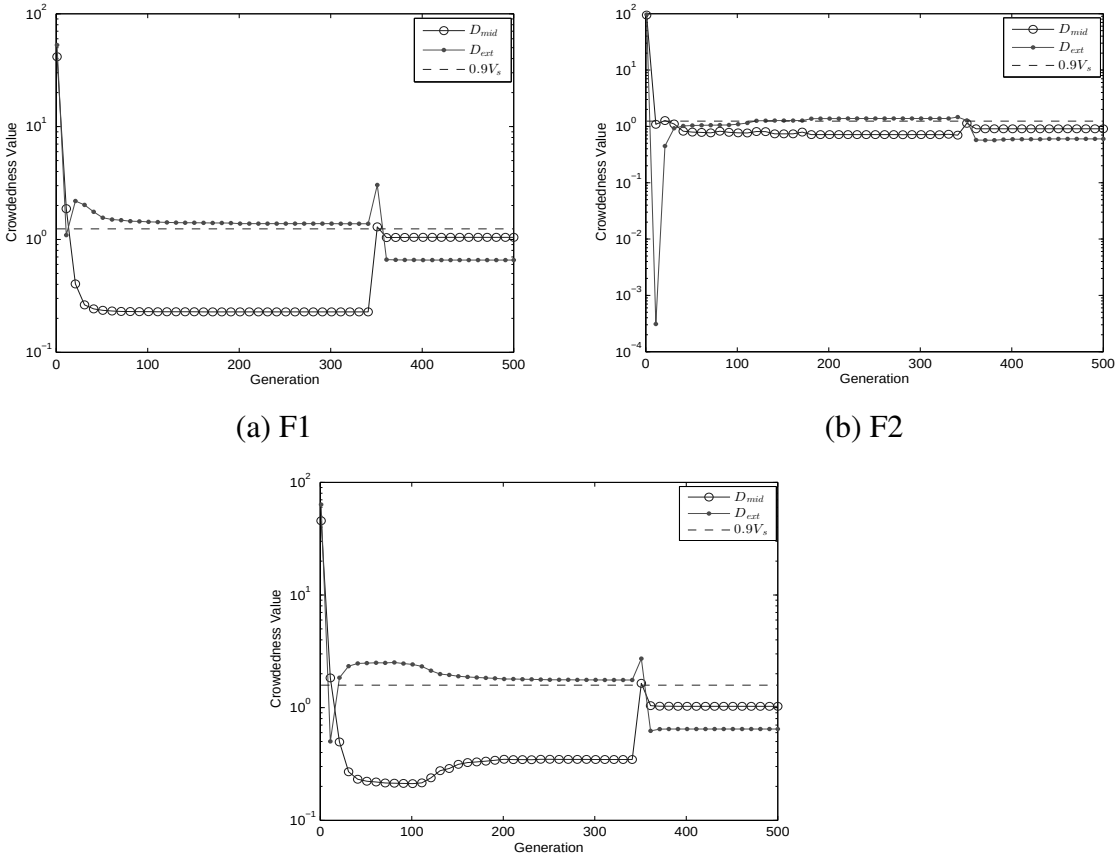


Fig. 3.6 Evolution of the mean IGD values of the test problems: (a) POL, (b) mF4, (c) F5, and (d) F6.

There are two basic issues involved in TP, however, which may influence the performance of the proposed algorithm. One is the definitions of the intermediate and extreme regions, since the definitions are based on the continuous PF, we would wonder what will happen to them for disconnected problems. The other one is whether TP has an influence on population diversity. Therefore, we carried out the following experiments.

3.4.1.1 Influence of Different Types of PF on D_{mid} and D_{ext}

To have a better understanding of the definition and function of D_{mid} and D_{ext}, we use F1 to F3 as the test problems, which represent two different types of problems. All the parameter settings of MOEA/D-TPN remain unchanged in the experiment. The crowdedness values of D_{mid} and D_{ext} against generation are plotted in Fig. 3.7, where V_s denotes a stable value obtained by D_{ext} in the first phase (350 generations). The figure clearly shows that, for both continuous and discontinuous problems, D_{mid} is smaller than D_{ext} when they reach their stable level, and the introduction of the second phase evolution helps find more extreme solutions, thus narrowing the gap between their crowdedness values. This indicates, for disconnected problems, the definitions of intermediate and extreme regions are also very meaningful and helpful. In the early stage (roughly 30 generations), especially



(a) F1

(b) F2

(c) F3

Fig. 3.7 Evolution curves of D_{mid} and D_{ext} on F1, F2, and F3.

shown for F2 in Fig. 3.7, D_{mid} seems to be higher than D_{ext}, and this is probably because the number of extreme solutions is too low, and many extreme individuals of W_e share the same solutions, leading to a small value of D_{ext}. It is worth noting that, the figure also confirms that parameter 0.9 is able to determine whether there is a significant difference between D_{mid} and D_{ext}.

3.4.1.2 Influence of TP on Population Diversity

As discussed in previous experiments, TP not only helps in finding extreme solutions, but seems to have the potential to increase population diversity for F2. To further investigate the diversity performance of TP, we tested MOEA/D+TP on F2. Fig. 3.8 shows the population at the 350th, 360th, 370th and 380th generations. We can observe that, at the end of the first phase (350th generation), the algorithm only finds solutions on the top left of the PF. When TP is triggered, the population gradually covers the disconnected subregions on the bottom right of the PF, which means TP helps diversify the population across the whole PF. One possible reason for this is that search directions have been changed in

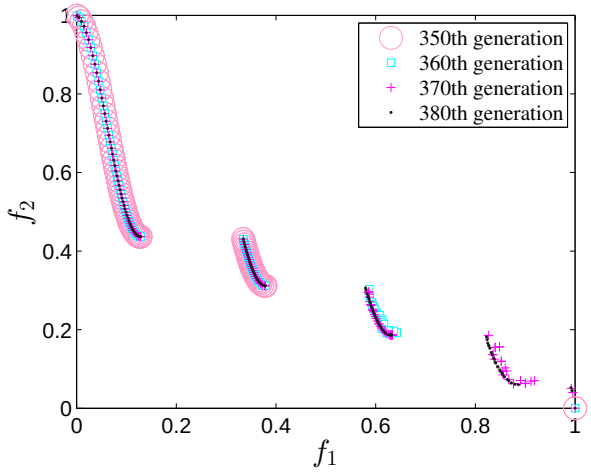


Fig. 3.8 Influence of TP on population diversity for F2.

Table 3.6 IGD values obtained by MOEA/D-TPN with different M_r settings for the instances F1 and F3

	F1			F3		
M_r	Best	Median	Worst	Best	Median	Worst
0.5	2.8338E-03	2.8487E-03	2.8652E-03	2.3177E-03	2.5467E-03	2.7673E-03
0.6	2.8328E-03	2.8476E-03	2.8664E-03	2.2376E-03	2.5282E-03	2.7168E-03
0.7	2.8256E-03	2.8455E-03	2.8614E-03	2.2342E-03	2.3835E-03	2.6090E-03
0.8	2.8272E-03	2.8506E-03	2.8662E-03	2.2667E-03	2.3880E-03	2.5654E-03
0.9	2.8367E-03	2.8564E-03	2.8911E-03	2.2795E-03	2.4364E-03	2.6435E-03
1	4.4105E-02	4.4278E-02	4.4482E-02	1.0936E-02	1.1245E-02	1.3015E-01

the second phase, which in return increases the chance for the population to explore new regions, thus providing a better distribution and coverage of solutions.

3.4.2 Effect of the Parameter M_r

The previous experimental results have shown that the TP scheme can improve the performance of MOEA/D. To further investigate the effect of the two-phase control parameter M_r on the performance of MOEA/D-TPN, six values of M_r (i.e., 0.5, 0.6, 0.7, 0.8, 0.9, 1) are tested on the instances F1 and F3. Note that, $M_r = 1.0$ means that the algorithm does not use the TP scheme. All the other parameters remain unchanged.

The IGD values are shown in Table 3.6. It can be clearly seen that $M_r = 0.7$ produces better results than the other settings for F1 and F3. Besides, the IGD values further demonstrate that the use of TP provides a notable improvement (different IGD values of $M_r = 1$ and $M_r < 1$) in approximating the PFs of F1 and F3.

Table 3.7 IGD values obtained by MOEA/D-N with different σ_{share} settings for the instances F2 and UF4

	F2			UF4		
σ_{share}	Best	Median	Worst	Best	Median	Worst
0.001	2.2851E-03	2.2935E-03	2.3380E-03	4.0583E-02	4.4081E-02	4.8602E-02
0.005	2.2832E-03	2.2891E-03	2.3370E-03	3.5806E-02	4.2000E-02	4.8282E-02
0.01	2.2857E-03	2.2931E-03	2.3401E-03	3.9485E-02	4.2244E-02	4.8336E-02
0.05	2.3556E-03	2.5442E-03	2.8170E-03	3.9932E-02	4.3017E-02	4.8391E-02
0.2	3.2096E-03	4.0473E-03	6.5844E-03	4.1503E-02	4.3274E-02	4.8961E-02
0.5	4.3035E-03	4.9096E-03	1.3758E-02	4.1904E-02	4.3827E-02	5.0084E-02

3.4.3 Effect of the Sharing Radius σ_{share}

In the proposed algorithm, the sharing radius σ_{share} in the sharing function is an important factor. A large σ_{share} will reduce the impact of the niche-guided scheme, while a small one may lead to too much exploration in the search space and may not ensure a fast convergence of the population. In the following, σ_{share} is set to 0.001, 0.005, 0.01, 0.05, 0.2, and 0.5, and MOEA/D-N is tested on F2 and UF4, both of which have shown to be sensitive to σ_{share} in the previous study. Note that, we do not use MOEA/D-TPN here because TP helps to solve the problem, which may shadow the importance of the niche-guided scheme. So, to avoid this situation, it is better to investigate the effect of the sharing radius separately.

Table 3.7 gives the effect of varying the sharing radius σ_{share}. Clearly, high values of σ_{share} aggravate IGD values because the algorithm is given more chance to produce a new individual by selecting neighbouring parent individuals, while a low value will make the algorithm concentrate on global exploration and cannot provide sufficient convergence performance. Notably, setting the value of σ_{share} to 0.005 offers better results on the tested instances F2 and UF4.

3.4.4 Effect of Population Size

The population size (N) is another key factor that greatly affects the performance of algorithms. To investigate its influence, we test three levels of population sizes for the algorithms SPEA2+SDE, NSGA-III and MOEA/D-TPN on the problems POL and mF4. Table 3.8 shows the population size settings for these algorithms. Since both NSGA-III and MOEA/D-TPN use the simplex-lattice design to set the number of reference points or sub-problems, their population sizes should satisfy the requirement $N = C^H_{M+H-1}$, where H is the number of divisions on each objective and M is the number of objectives. The population size for NSGA-III is set as the smallest multiple of four higher than N, which

Table 3.8 Population size settings for three algorithms

Level	POL			mF4		
	SPEA2 +SDE	NSGA -III	MOEA/D -TPN	SPEA2 +SDE	NSGA -III	MOEA/D -TPN
1	52	52	50	48	48	45
2	100	100	100	92	92	91
3	200	200	200	192	192	190

Table 3.9 Best, median and worst IGD values of three algorithms on POL and mF4

Level	POL			mF4		
	SPEA2 +SDE	NSGA -III	MOEA/D -TPN	SPEA2 +SDE	NSGA -III	MOEA/D -TPN
1	1.8971E-01	3.1802E-01	**1.8071E-01**	1.5419E+00	**3.5911E-01**	4.2461E-01
	2.6678E-01	5.3476E-01	**2.0400E-01**	1.5695E+00	**3.7089E-01**	4.4307E-01
	3.6169E-01	6.0683E-01	**2.6377E-01**	1.6279E+00	**3.8630E-01**	4.7349E-01
2	1.4821E-01	2.4915E-01	**9.4054E-02**	1.4931E+00	2.6242E-01	**2.5146E-01**
	1.8373E-01	3.2066E-01	**1.0687E-01**	1.5119E+00	2.7067E-01	**2.6229E-01**
	2.4422E-01	3.6438E-01	**1.2287E-01**	1.5232E+00	2.7626E-01	**2.6902E-01**
3	7.9373E-02	1.4470E-01	**5.6305E-02**	1.4775E+00	1.7747E-01	**1.6328E-01**
	1.0496E-01	2.0711E-01	**5.9582E-02**	1.4795E+00	1.8393E-01	**1.6476E-01**
	1.7027E-01	2.2150E-01	**6.1822E-02**	1.4842E+00	1.8544E-01	**1.6714E-01**

is referenced to the setting in [43]. SPEA2+SDE has the same population size setting as NSGA-III.

Table 3.9 gives the IGD results of the three algorithms at different population size levels. It can be observed that, for POL, MOEA/D-TPN outperforms the rest on each level of the population size by a clear margin. When the population size is too small, NSGA-III obtains better IGD values for mF4 than MOEA/D-TPN and SPEA2+SDE, but it is gradually exceeded by MOEA/D-TPN with an increase in the population size. This implies MOEA/D-TPN work well if a normal size of population is provided.

3.4.5 Limitations

The study has two main limitations. The first is that there are some parameters involved in MOEA/D-TPN, and it is not easy for users to set them in advance without any knowledge about a problem. For example, the setting of the control parameter M_r used in the two-phase scheme may vary with the difficulty of the problem considered. On one hand, for easy problems, $M_r = 0.5$ may be enough to drive most of individuals toward the true PF, but for difficult problems, $M_r = 0.5$ may not be enough to obtain an accurate estimation of population crowdedness and the nadir point. On the other hand, a large M_r is good for estimating population crowdedness, but may cause the algorithm to have few

computational resources for the second-phase evolution, thus unable to have a good approximation at the end of the evolution. For these reasons, M_r is suggested to be in [0.5, 0.8]. Note that, the two-phase scheme is really helpful for complex problems in question and a value of 0.7 for M_r generates good results for them. Another key parameter is σ_{share}, and its setting relates to the number of objectives and the population size. $\sigma_{share} = 0.005$ may not be the best choice for all the problems, however, the experimental study has seen an improvement in MOEA/D with the use of niche. The second limitation is the current work mainly focuses on bi- and three-objective complex problems, which may induce some bias. Although MOEA/D-TPN has achieved encouraging performance on the problems in question, that does not imply MOEA/D-TPN is always better than the other compared algorithms. Further investigations are required to have a better understanding of its performance.

3.5 Summary

An extensive review of the performance of MOEA/D has shown that the MOEA/D algorithms tend to find solutions located in the intermediate region of an irregular PF and have difficulty in solving problems with complex PF shapes, i.e., the PF with a sharp peak and long tail, and disconnected regions. This chapter has extended the MOEA/D framework to overcome this drawback by introducing a novel TP scheme and a niche-guided scheme. While TP alternatively divides the whole optimization procedure, according to the crowdedness of solutions on the PF, into two phases, and helps the population in finding points on the boundary of the PF in the second phase; the niche-guided scheme is used to maintain the population diversity in cases where the PF has several disconnected regions that may result in MOEA/D producing many duplicate solutions. In this way, the improved MOEA/D is expected to be capable of handling difficult landscapes and achieving desirable optimization results.

The proposed algorithm has been tested on some complex problems, including existing and newly designed test functions. Experimental results have shown that the improved MOEA/D achieves better performance than several compared MOEA/D variants, and outperforms another two algorithms, i.e., SPEA2+SDE and NSGA-III, on the test problems. We would like to summarize some key findings regarding the proposed MOEA/D-TPN as follows:

1. The niche-guided scheme provides a high level of population diversity for complex test problems (e.g., F2, UF4) during the evolution, especially when the problem being optimized is discontinuous.

2. When solving problems whose PF has a sharp peak or a long tail, MOEA/D with TP shows significantly better performance than that without TP. The improved

MOEA/D has been tested on the convex version of DTLZ2, mF4, and F5, proving that it is very efficient for finding points on the boundary of the PF. Besides, TP can adjust search directions and diversify individuals (e.g., F2), which may be helpful for disconnected problems.

3. The experimental results have also revealed that, although a good distribution of weight vectors (i.e., WS-transformation) for MOEA/D is available, it cannot necessarily guarantee the evenness of the resulting solutions on complex PF shapes. This may motivate researchers to seek other possible improvements, like advanced decomposition approaches, to enhance MOEA/D instead of pursuing only a better distribution of weight vectors.

Chapter 4

EAs Based on Scalarizing Functions for MOPs

Scalarizing functions, as the name suggests, are a kind of functions that can cast a vector to a scalar value. This way, two vectors can be indirectly distinguished by comparing the scalar values cast. It is well recognized that scalarizing functions can be incorporated with EAs to solve MOPs. With the aid of scalarizing functions involving possibly human decision makers' preference information, an MOP is transformed into a number of simple subproblems. After the scalarization phase, these simple subproblems can be solved by some existing theory and methods. So far, many scalarizing-based EAs have been proposed and their effectiveness has been widely verified.

Despite their popularity for MOPs, the properties of scalarizing functions have not been well understood. In this chapter, while identifying limitations of existing scalarizing functions, we propose new scalarizing functions and analyze their properties deeply. After that, we develop an EA framework based on the proposed scalarizing functions for solving hard MOPs.

This chapter is organized as follows. In Section 4.1, we introduce some related knowledge and the incentive of this research work. Section 4.2 presents the proposed scalarizing functions, followed by an EA framework on the basis of them in Section 4.4. Section 4.5 provides experimental studies with respect to the proposed EA framework. A deep analysis of parameter sensitivities is presented in Section 4.6. Section 4.7 concludes the work.

4.1 Introduction

Despite plenty of advances, decomposition-based EAs like MOEA/D still receive increasing research interests. A particular research direction is with regard to scalarizing functions, which have not been fully explored yet. In MOEA/D, the weighted sum (WS), the weighted Tchebycheff (TCH) and penalty-based boundary intersection (PBI) are the top

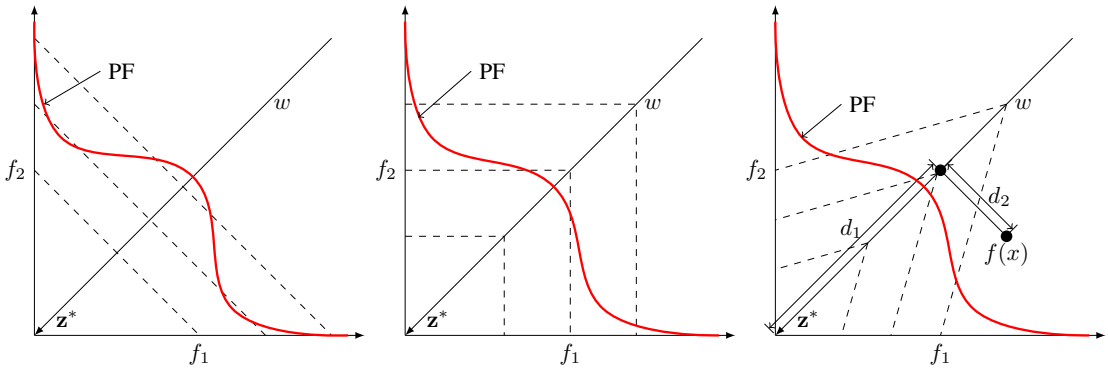
three commonly-used scalarizing functions. These scalarizing functions respectively have their own strengths and drawbacks [174, 184, 185]. In view of the advantages and disadvantages of each scalarizing function, Ishibuchi *et al.* [85, 86] proposed to use different scalarizing functions adaptively or simultaneously during the search. In [59], a generalized form of scalarizing functions was developed to cope with various PF geometries. Sato [147] proposed an inverted PBI method to overcome the poor spread performance of existing scalarizing functions in some problems. Modified or advanced scalarizing functions have also been developed to facilitate environmental selection in other algorithms [28, 94, 144, 186].

It is noteworthy that one important property of a scalarizing function is the shape or positioning of its contour lines [50, 184]. The contour lines are a set of equal scalarizing function values and play a crucial role in guiding the search in scalarizing search algorithms [50]. In [50], the authors argued that the dynamics of the search process is rather independent of the scalarizing function under consideration and instead mainly influenced by the induced contour lines. Another study [175] showed that constraining the contours of the scalarizing function can improve the search ability. Recently, Wang *et al.* [174] have systematically studied the search ability of a family of widely-used scalarzing functions with different contours, called the L_p scalarizing functions, and have argued that different contours should be used at different search stages.

Generally, desired contours in scalarizing search algorithms can be obtained by 1) using traditional scalarizing functions and modifying their contours by adding constraints [175] or specifying different parameters [174] or 2) designing new and effective scalarizing functions. The former method looks intuitive and easy but may not always generate the exact contours one wants. For example, the modified scalarizing functions (excluding PBI) in both [175] and [174] cannot produce contour lines that have opening angles [50] smaller than $\pi/2$, which may not balance diversity and convergence well during the search. In this chapter, the focus is on the latter.

The three scalarizing functions mentioned previously, i.e., WS, TCH, and PBI, have been widely used in decomposition-based MOEAs. Their contour lines are displayed in Fig. 4.1. Despite great success in solving a variety of MOPs, these scalarizing functions have their own limitations. For example, PBI is very sensitive to the search landscape of the objective space and may miss some PF points if the underlying penalty factor is not well-tuned [147, 184].

On the other hand, WS and TCH belong to the family of the L_p ($p \geq 1$) scalarizing functions [174], and they are two extreme cases of this family (WS and TCH correspond to L_1 and L_∞, respectively). As pointed out by Wang *et al.* [174], TCH is the best in the L_p family in terms of diversity maintenance because its contour lines have the smallest opening angle, i.e., $\pi/2$, as shown in Fig. 4.1(b). However, We argue here that a contour with $\pi/2$ opening angle is still insufficient to maintain diversity in some cases. Fig. 4.2



(a) WS (b) TCH (c) PBI

Fig. 4.1 Illustration of three scalarizing functions on weight vector w, where dashed lines are contour lines.

illustrates a situation where the L_∞ scalarizing function fails to maintain diversity. In the figure, three individuals (B, C and D) are of great importance to diversity, but they will be replaced by two boundary individuals (A and E) because their improvement regions [1] contain A and/or E, i.e., $\{A\} \subset \Phi(B)$, $\{A, E\} \subset \Phi(C)$, and $\{E\} \subset \Phi(D)$. This implies that L_∞ along with its family members lacks the property of maintaining/promoting diversity itself. It may be of little use and even fails if search environments are very complex and little information is available in advance.

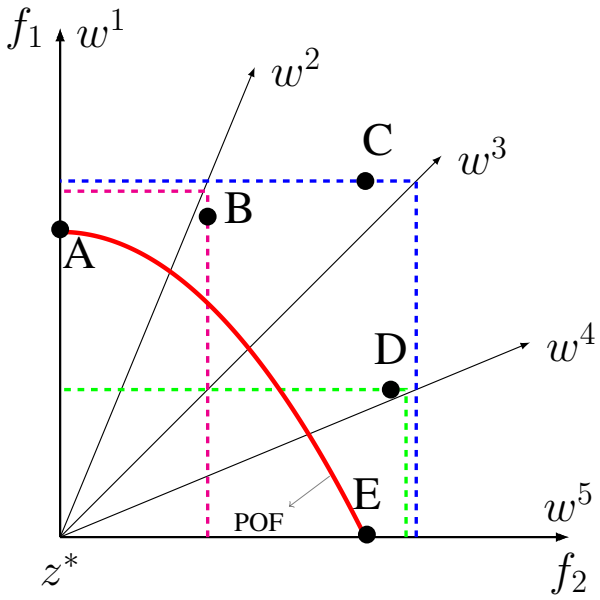


Fig. 4.2 Illustration of solution distribution in the bi-objective space. Dashed lines are contours of L_∞ scalarizing functions.

Generally, the above-mentioned drawback with regard to TCH can be alleviated by the following possible ways. First, constraints can be added to the contour line of TCH to reduce the opening angle or TCH works with appropriate replacement strategies to stop

[1] In scalarizing-based search, the improvement region (denoted $\Phi(x)$) of a solution x refers to the area where any other solution is better than this solution in terms of the scalarizing value.

abandoning diverse individuals. Second, a new and effective scalarizing function with adjustable contours or improvement regions to control diversity would help MOEA/D yield good performance. Obviously, the second approach is more straightforward and easier to implement because it does not require any modification of the basic framework of decomposition-based algorithms. For this reason, it is desirable to devise new scalarizing functions for evolutionary multiobjective optimization (EMO).

4.2 New Scalarizing Functions

In the following, two new scalarizing functions are proposed for EMO.

4.2.1 Multiplicative Scalarizing Function (MSF)

The MSF approach is written as:

$$g^{msf}(x|w,z^*) = \frac{\left[\max\limits_{1\le i\le M}\left(\frac{1}{w_i}|f_i(x)-z_i^*|\right)\right]^{1+\alpha}}{\left[\min\limits_{1\le i\le M}\left(\frac{1}{w_i}|f_i(x)-z_i^*|\right)\right]^{\alpha}}, \tag{4.1}$$

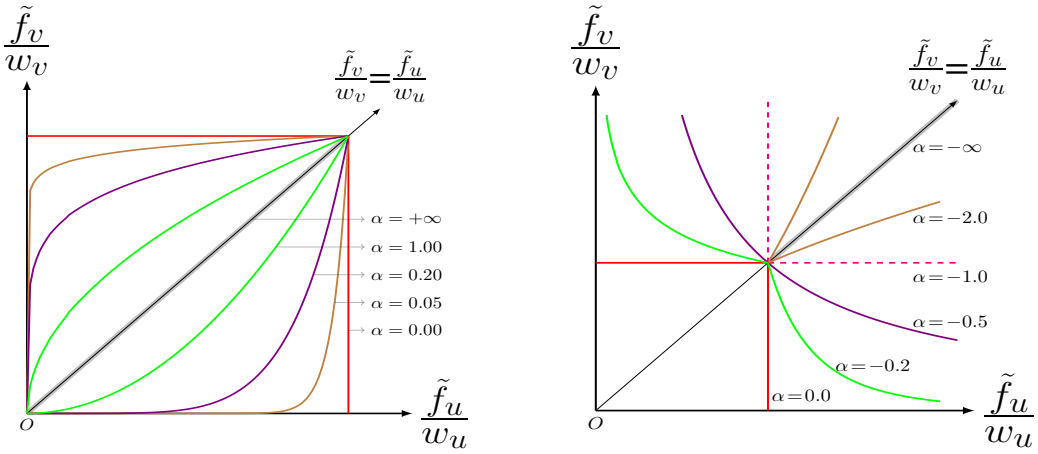
where α is a control parameter, and if the denominator is zero, a sufficiently small positive value (e.g., 10^{-4}) is added to keep division legal. When $\alpha = 0$, $g^{msf}(x|w,z^*)$ degenerates to Eq.(2.3) presented in Chapter 3. That is, the TCH method is a special case of MSF.

For denotion convenience, let $\tilde{f}_i = f_i - z_i^*$. We also assume u and v ($1 \le u,v \le M$) are the indices that maximize and minimize $\tilde{f}_i/w_i$ ($1 \le i \le M$), respectively. Fig. 4.3 presents contour lines of MSF with different α values on the $(\tilde{f}_u/w_u)(\tilde{f}_v/w_v)$-plane. As can be seen from the figure, the size of the improvement region decreases with the increase of α. Obviously, $\alpha \ge 0$ is more desired compared with $\alpha < 0$ because just like the WS method, MSF with $\alpha < 0$ is unable to approximate non-convex PFs and is very likely to lose diversity as mentioned earlier[2]. For this reason, we only consider $\alpha \ge 0$ in this work.

Theorem 1. *On the $\tilde{f}_u\tilde{f}_v$-plane, the maximum size of improvement region enclosed by $g^{msf}(x|w,z^*) = c$ ($c \ge 0$), denoted $\Delta(c)$, is equal to $w_u w_v c^2/(2\alpha+1)$.*

Proof. The improvement region is enclosed by the curve $g^{msf}(x|w,z^*) = c$, as shown in Fig. 4.4. It is easy to see that $g^{msf}(x|w,z^*) = c$ and $\tilde{f}_u/w_u = \tilde{f}_v/w_v$ have two common points of intersection, i.e., (0,0) and (cw_u, cw_v). Next, we calculate the two parts of $g^{msf}(x|w,z^*) = c$ below and above $\tilde{f}_u/w_u = \tilde{f}_v/w_v$, respectively.

[2]However, $\alpha \le 0$ may provide fast convergence in the early stage of the search and solve concave MOPs efficiently.



(a) $\alpha \geq 0$ (b) $\alpha \leq 0$

Fig. 4.3 Contour lines of MSF with different α values.

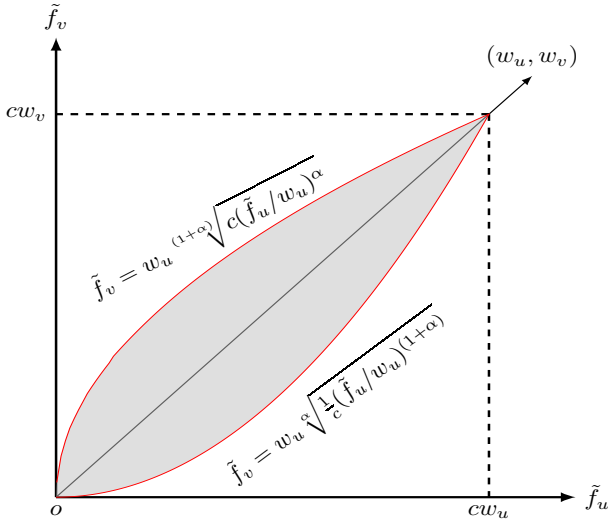


Fig. 4.4 The improvement region (shaded area) of MSF.

If $\tilde{f}_u/w_u > \tilde{f}_v/w_v$, then $g^{msf}(x|w,z^*) = (\tilde{f}_u/w_u)^{1+\alpha}/(\tilde{f}_v/w_v)^\alpha = c$. Thus $f_v = w_v \sqrt[(1+\alpha)]{c(\tilde{f}_u/w_u)^\alpha}$.

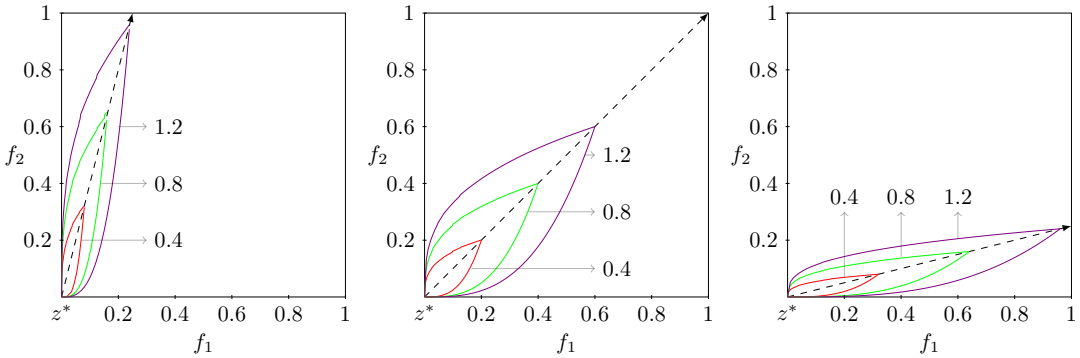
If $\tilde{f}_u/w_u < \tilde{f}_v/w_v$, then $g^{msf}(x|w,z^*) = (\tilde{f}_v/w_v)^{1+\alpha}/(\tilde{f}_u/w_u)^\alpha = c$. Thus $f_v = w_v \sqrt[\alpha]{\frac{1}{c}(\tilde{f}_u/w_u)^{(1+\alpha)}}$.

Therefore, $\Delta(c)$ is equal to the area bounded by the above two function curves. That is,

$$\Delta(c) = \int_0^{cw_u} \left(w_u \sqrt[\alpha]{\frac{1}{c}\left(\frac{\tilde{f}_u}{w_u}\right)^{(1+\alpha)}} - w_u \sqrt[(1+\alpha)]{c\left(\frac{\tilde{f}_u}{w_u}\right)^\alpha} \right) d\tilde{f}_u = \frac{w_u w_v c^2}{2\alpha+1}. \tag{4.2}$$

□

It is clear from the theorem that, the size of the improvement region (particularly in the bi-objective space) for a predefined weight vector is controlled by α. When α is fixed, the improvement region is affected by the weight vector values, as shown in Fig. 4.5. Generally, intermediate weight vectors tend to have larger $\Delta(c)$ values than



(a) $w = (0.2, 0.8)^T$ (b) $w = (0.5, 0.5)^T$ (c) $w = (0.8, 0.2)^T$

Fig. 4.5 Contour lines of MSF with $\alpha = 0.5$ for contour values of 0.4, 0.8, and 1.2.

boundary ones. This is because the product of the elements of an intermediate weight vector is larger than those of an boundary weight vector. This means that subproblems associated with intermediate weight vectors have more opportunities to be updated than those with boundary weight vectors. Thus, to be fair for all the subproblems, it is plausible to penalize intermediate subproblems or compensate boundary subproblems so that all subproblems can have relatively equal size of improvement regions.

According to Theorem 1, a straightforward way of balancing subproblems is to vary α for different subproblems. In this work, we adjust α as follows:

$$\alpha = \beta \left\{ M \min_{1 \le i \le M} (w_i) \right\}, \tag{4.3}$$

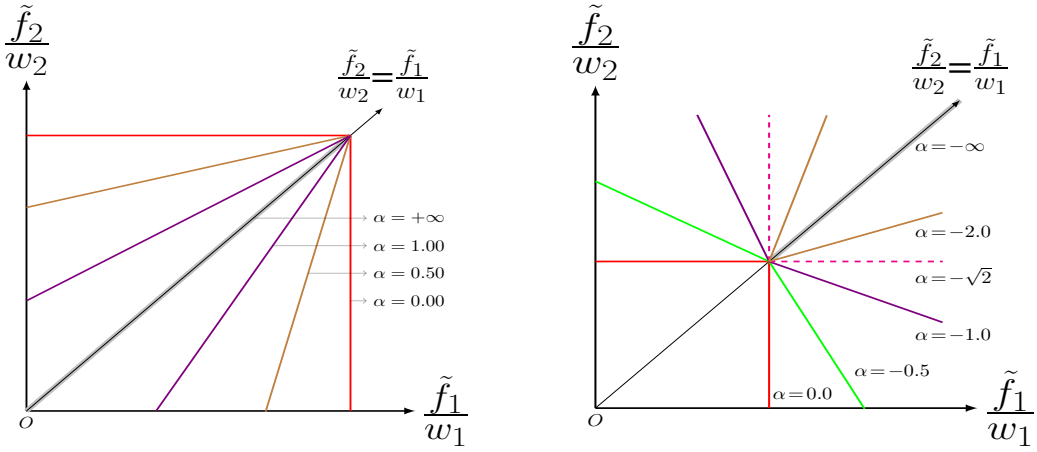
where w_i is the i-th element of weight vector w, and M is the number of objectives. $0 \le M \min_{1 \le i \le M} (w_i) \le 1$ always holds under the assumption of $\sum_{i=1}^{M} w_i = 1$. β is the underlying control parameter. It is worth mentioning that the improvement region in TCH is non-adjustable as it is constant (i.e., $\Delta(c) = w_u w_v c^2$) if both the contour line value c and the weight vector w are predetermined. However, for MSF, $\Delta(c)$ can control each subproblem's improvement region through the adjustment of α (or β in Eq.(4.3)).

4.2.2 Penalty-based Scalarizing Function (PSF)

Inspired by the idea of PBI that controls diversity by penalizing solutions far away from a weight vector, we modify the weighted Chebycheff function in the following way:

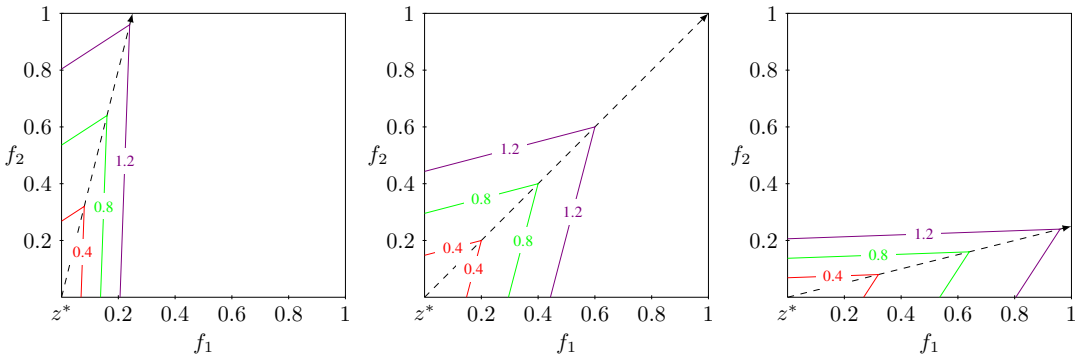
$$g^{psf}(x|w, z^*) = \max_{1 \le i \le M} \left(\frac{1}{w_i} |f_i(x) - z_i^*| \right) + \alpha d, \tag{4.4}$$

$$d = \frac{\sqrt{\|f(x) - z^*\|^2 \|w\|^2 - \|(f(x) - z^*)^T w\|^2}}{\|w\|}, \tag{4.5}$$



(a) $\alpha \geq 0$ (b) $\alpha \leq 0$

Fig. 4.6 Contour lines of PSF with different α values



(a) $w = (0.2, 0.8)^T$ (b) $w = (0.5, 0.5)^T$ (c) $w = (0.8, 0.2)^T$

Fig. 4.7 Contour lines of PSF with $\alpha = 1.0$ for contour values of 0.4, 0.8, and 1.2.

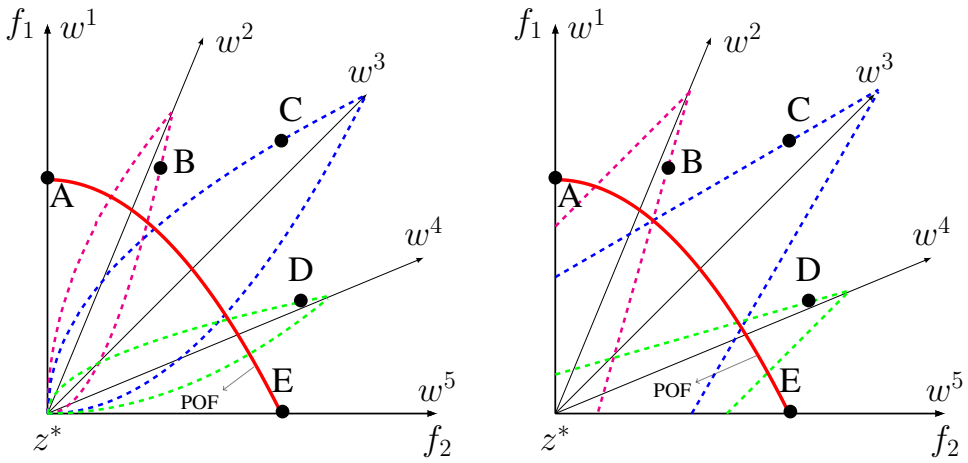
where d is the perpendicular distance of a solution to the weight vector w, i.e., d is the same as the perpendicular distance d_2 defined in PBI [188] (see Eq. (2.6) for the definition of d_2). α is a penalty value used to control diversity. Fig. 4.6 illustrates contour lines of PSF with different α values. Similar to MSF, the improvement region of PSF varies dramatically with α, and $\alpha \geq 0$ is preferable as diversity is the focus of this work. Fig. 4.7 presents the contour lines of PSF for different weight vectors.

Like MSF, the size of improvement region of PSF also depends largely on α and the considered weight vector w. Thus, PSF use the same adjustment strategy (see Eq.(4.3)) to balance different subproblems.

4.2.3 Similarities and Differences

4.2.3.1 MSF and PSF vs L_p

Let us revisit the problem previously illustrated in Fig. 4.2, where L_∞ and other L_p scalarizing functions cannot induce proper contours or improvement regions to avoid diversity loss. We wonder whether the proposed MSF and PSF can overcome this drawback.



(a) MSF (b) PSF

Fig. 4.8 Illustrations of MSF and PSF for maintaining diversity, where dashed lines are contours.

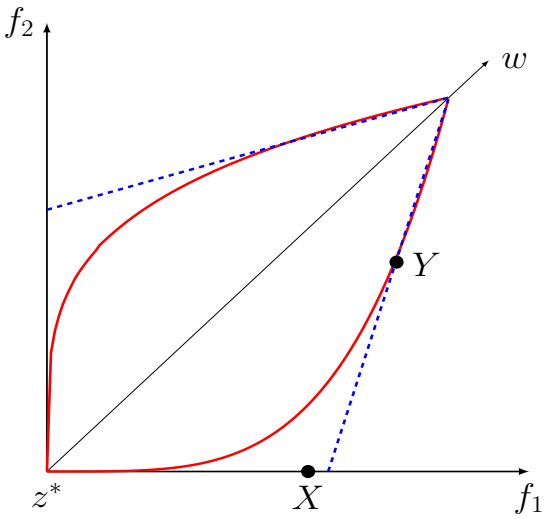


Fig. 4.9 An illustration where MSF (red solid) and PSF (blue dashed) induce different shapes of contours.

Fig. 4.8 presents the contours of MSF and PSF (both with $\alpha = 1$) passing through three important points (B, C and D). It is clear that when MSF or PSF is used, all the five points from A to E can survive during the replacement because no point resides in the improvement region of another point. Thus, MSF and PSF are effective and promising for diversity maintenance.

4.2.3.2 MSF vs PSF

The improvement region of both methods varies with α, and both degenerates to TCH when $\alpha = 0$. For $\alpha \geq 0$, however, the geometries of MSF and PSF are different. the contour lines of MSF are nonlinear whereas those of PSF are polytopes. Thus, boundary points beside the ideal point can be in an improvement region induced by PSF, but this is

not the case with MSF. Fig. 4.9 presents a situation where a boundary point X will replace an intermediate point Y associated with the search direction w if PSF with inappropriate opening angles is used. But this will not happen in MSF. For this reason, MSF may keep diversity better than PSF whereas PSF may have advantage in locating boundary solutions.

4.2.3.3 PSF vs PBI

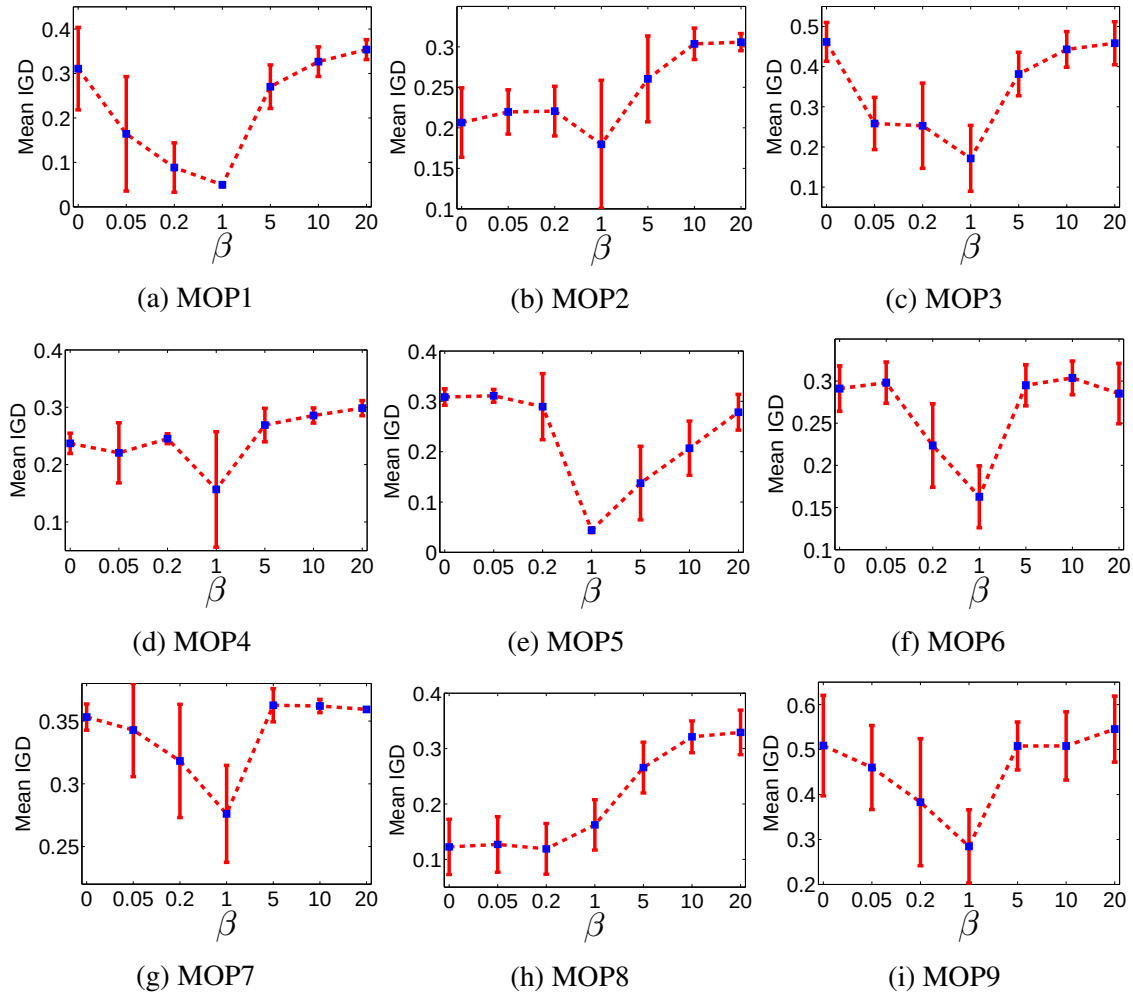
Since PSF borrows the idea of diversity maintenance from PBI, PSF and PBI have similar contour lines, i.e., their contour lines are polytopes. However, they differs much in convergence promotion. Specifically, PBI measures convergence via d_1 values [188] (see Eq. (2.6) for the definition of d_1). In complex problems with irregular PF shapes [90], d_1 values vary significantly. In order to balance diversity and convergence, PBI needs to carefully select the penalty factor. Otherwise, PBI is likely to obtain an incomplete approximation of the PF [147]. Unlike PBI, PSF measures convergence via TCH, which can approximate both convex and nonconvex PF geometries. Therefore, PSF may be less sensitive to different PF scenarios compared with PBI.

4.3 Parameter Sensitivity in MSF and PSF

As our focus is mainly on the diversity aspect of scalarizing functions, test problems used for experimental validation should be able to challenge MOEAs' diversity performance. For this reason, the MOP [127] test suite is selected. This test suite has seven instances, each of which has local attractors on boundary regions of the PF. Thus, MOEAs are very easily trapped into these attractors if their diversity is not well maintained. A detailed description of the MOP test suite can be found in Appendix A, where two more tri-objective instances, i.e., MOP8 and MOP9, are proposed by considering new characteristics. MOP8 places local attractors in the intermediate regions of its linear PF, whereas MOP9 increases optimization difficulties by placing local attractors only on corner regions (i.e. the intersection of the PF and coordinate axes). These added features are expected to further understanding of MOEAs' search behaviour.

The proposed MSF and PSF are integrated into the MOEA/D-DE [120] framework, whose recombination operator is replaced by the adaptive operator [123] due to its reported success on MOP problems [127].

Parameters in MOEA/D-DE were set as follows. The population size N was 100 and 105 for bi- and tri-objective problems, respectively. The neighbourhood size T was $T = 0.1N$ and the probability δ used to select mating neighbourhood was $\delta = 0.9$. The maximal allowable number n_r of solutions to be replaced by a child solution was $n_r = 2$. Due to the difficulty of the MOP test suite, the maximum number $MaxGen$ of generations



(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

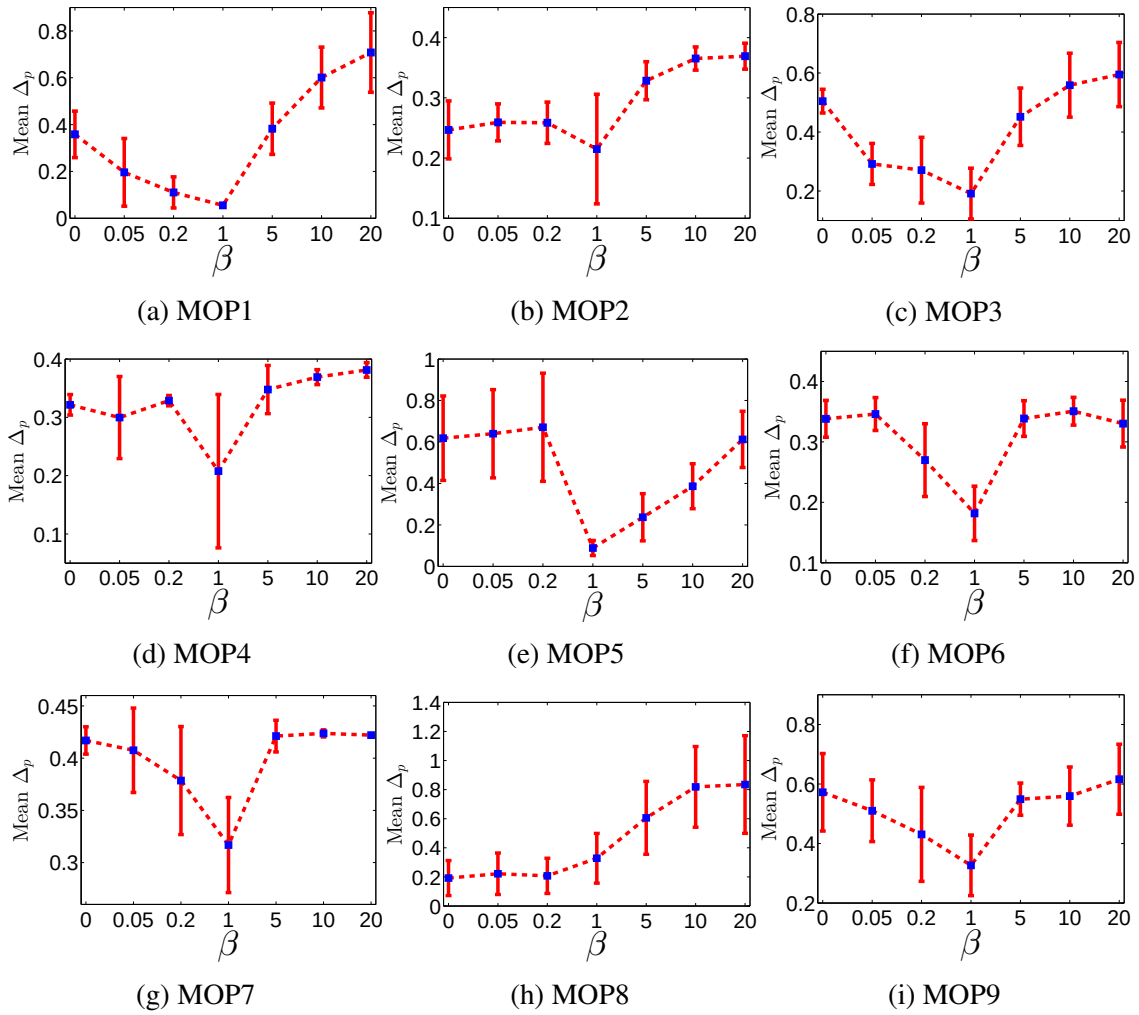
(g) MOP7 (h) MOP8 (i) MOP9

Fig. 4.10 Mean IGD values obtained by MSF with different β settings.

was set to $MaxGen = 5000$. The total number of independent runs was 31. Performance metrics chosen for this experiment are IGD [192], Δ_p [151] and HVD [197].

Since both MSF and PSF use Eq. (4.3) to assign α values, we just need to test the influence of different β values in Eq. (4.3). In the experimental study, β was chosen from $\{0, 0.05, 0.2, 1, 5, 10, 20\}$ for MSF and $\{0, 1, 5, 10, 20, 100\}$ for PSF. $\beta = 0$ is a special case where both MSF and PSF degenerate to TCH.

Figs. 4.10–4.12 plot the mean values of three metrics obtained by MSF with different β settings, where standard deviation is shown around the mean values. Two observations can be obtained from the figures. First, all the three metrics are roughly consistent when they are used for performance assessment. The only exception occurs on MOP7, where both IGD and Δ_p show the performance improves at first and then degrades as β increases from 0 to 20. HVD, however, shows a conflicting performance trend on MOP7. This may be because HVD prefers boundary solutions but does not necessarily favour well-diversified distribution on this particular instance. Second, for the majority of the problems, the performance is likely to be maximized when β approximately equals one.



(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

(g) MOP7 (h) MOP8 (i) MOP9

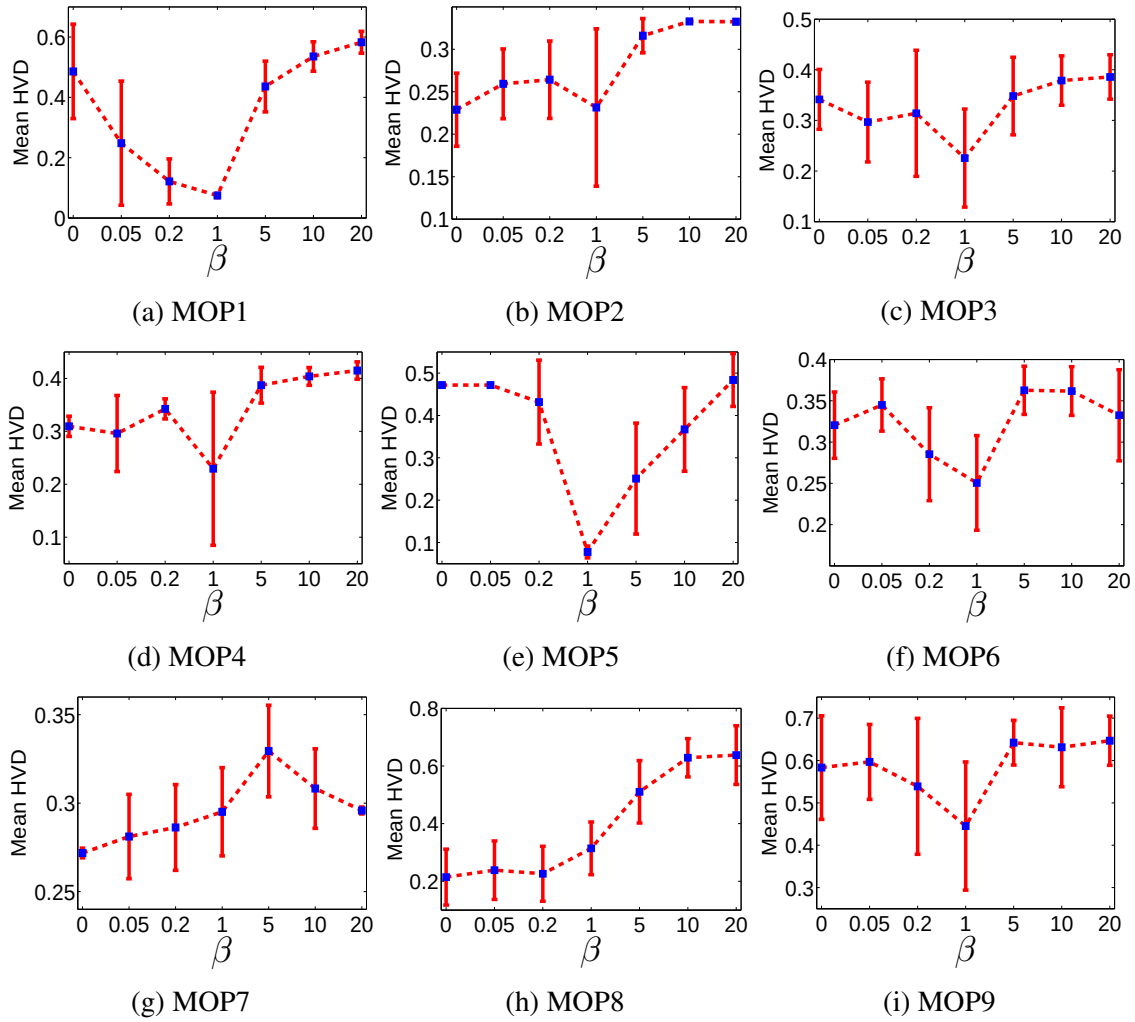
Fig. 4.11 Mean Δ_p values obtained by MSF with different β settings.

Meanwhile, it seems that a smaller β value is suitable for MOP8. This implies that, when local attractors reside in the intermediate regions of the PF, restrictions on the diversity aspect of MSF can be relaxed and MSF with a large improvement region is helpful in this situation.

On the other hand, the mean values of the three metrics obtained by PSF with different β settings are displayed in Figs. 4.13–4.15. Similar observations can be obtained from these figures, and PSF works best on most of the problems when β is around ten.

The above results clearly show that both MSF and PSF can help improve decomposition-based MOEAs if the corresponding control parameter is well configured. The experiment indirectly reflects that the popular TCH method (corresponding to the case of $\beta = 0$ in MSF and PSF) are not always the best choice, and enhancing diversity can leads to a clear performance improvement.

Besides, the mean HVD evolution curves obtained by MSF and PSF versus the number of generations are shown in Figs. 4.16 and 4.17. We can observe that different β values



(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

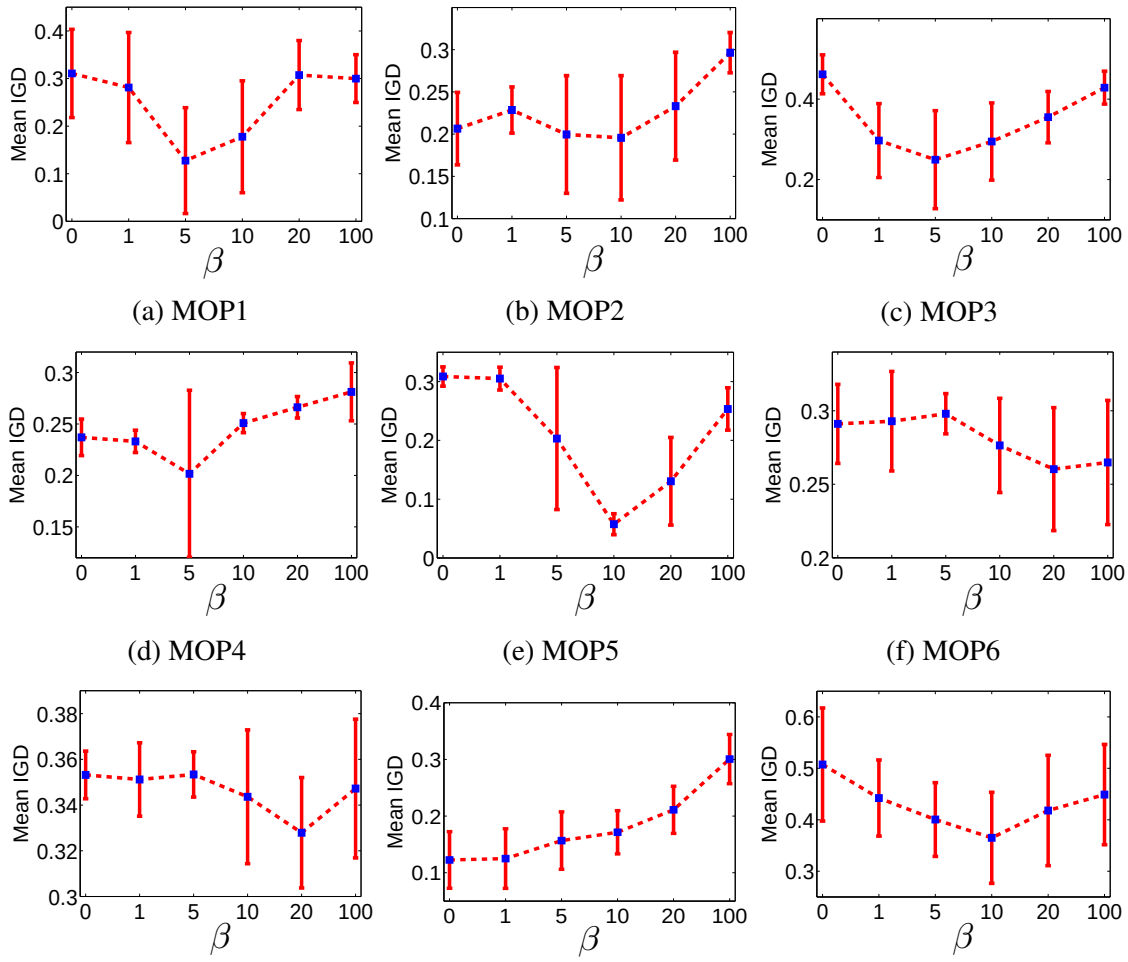
(g) MOP7 (h) MOP8 (i) MOP9

Fig. 4.12 Mean HVD values obtained by MSF with different β settings.

result in distinct performances. $\beta = 0$ is not the best setting for MSF and PSF on the majority of the test problems in terms of final HVD values. However, it seems that smaller β values are likely to converge faster, owing to relatively larger improvement regions. These observations suggest decomposition-based MOEAs may need different β (or the resulting α) values at different stages of the search. Therefore, it is plausible to adaptively adjust the value of α during the search.

4.4 Proposed EA Framework

The proposed MOEA/D variant (eMOEA/D) remains almost the same as its predecessors [120, 188] except a few modifications in scalarizing methods, population reproduction and solution replacement. The framework of the algorithm is depicted in Algorithm 4.1. First, a set of uniformly-distributed weight vectors is created, and each weight vector is assigned a neighbourhood containing T closest weight vectors. Meanwhile, the ideal and nadir



(a) MOP1 (b) MOP2 (c) MOP3

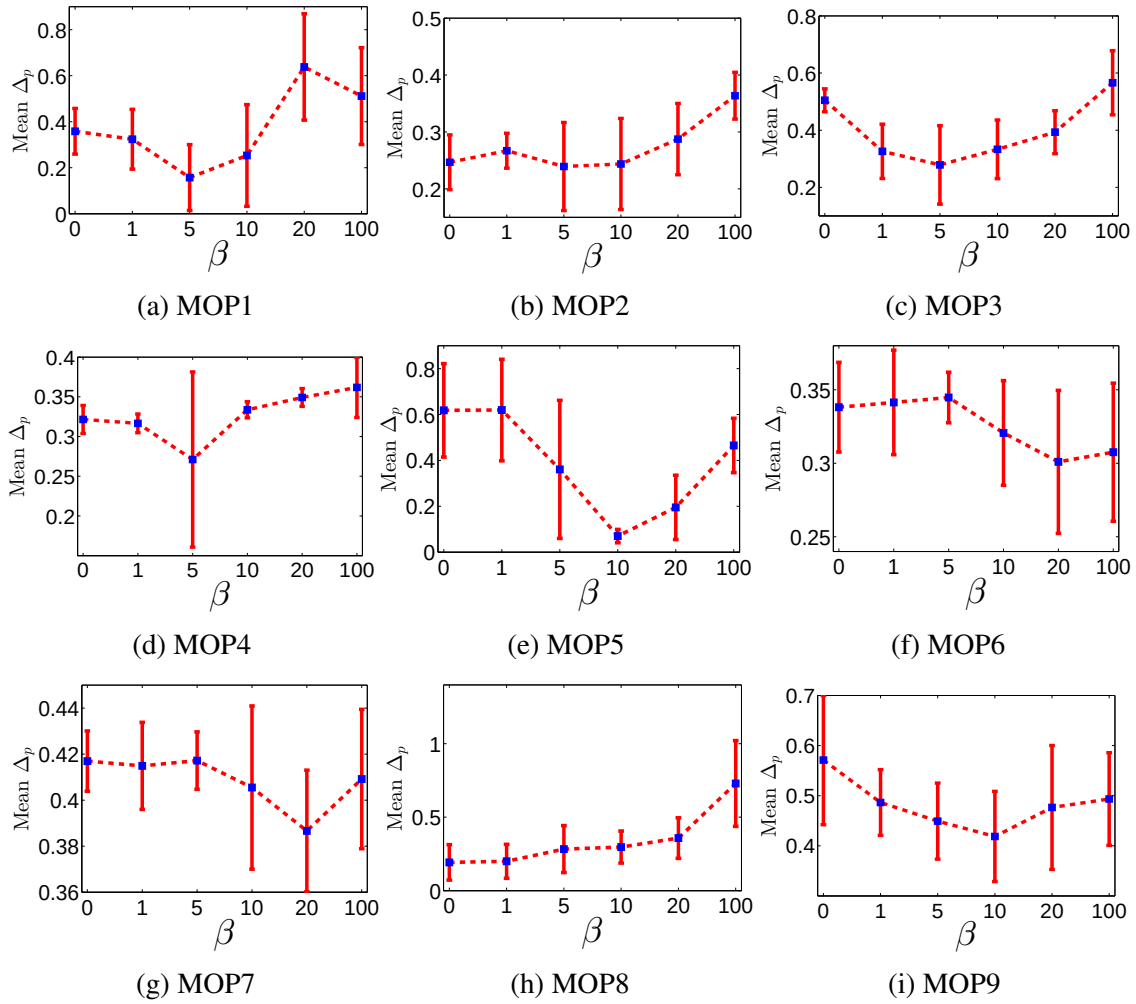
(d) MOP4 (e) MOP5 (f) MOP6

(g) MOP7 (h) MOP8 (i) MOP9

Fig. 4.13 Mean IGD values obtained by PSF with different β settings.

points are estimated by the best and worst objective values, respectively, in the population. Then, either MSF or PSF is chosen as a scalarizing function beforehand. The α value of the scalarizing function SF is generationally updated in line 7. For each subproblem i, mating parents are selected only from $B(i)$ (see line 9), which is the same as the original MOEA/D [188] but different from another popular variant MOEA/D-DE [120]. In line 10, genetic operators are applied on the selected parents to produce offspring. The offspring is evaluated in terms of the objective vector and is then used to update the ideal point. From line 12 to line 21, a new solution replacement strategy is introduced, and similar to MOEA/D-DE, we also place a restriction on the number of replacements (see lines 18-20). At the end of every generation, the approximated nadir point is updated by the whole population.

In the following subsections, the main modifications, i.e., the adaptive scalarizing strategy, reproduction operation and solution replacement, are explained in detail.



(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

(g) MOP7 (h) MOP8 (i) MOP9

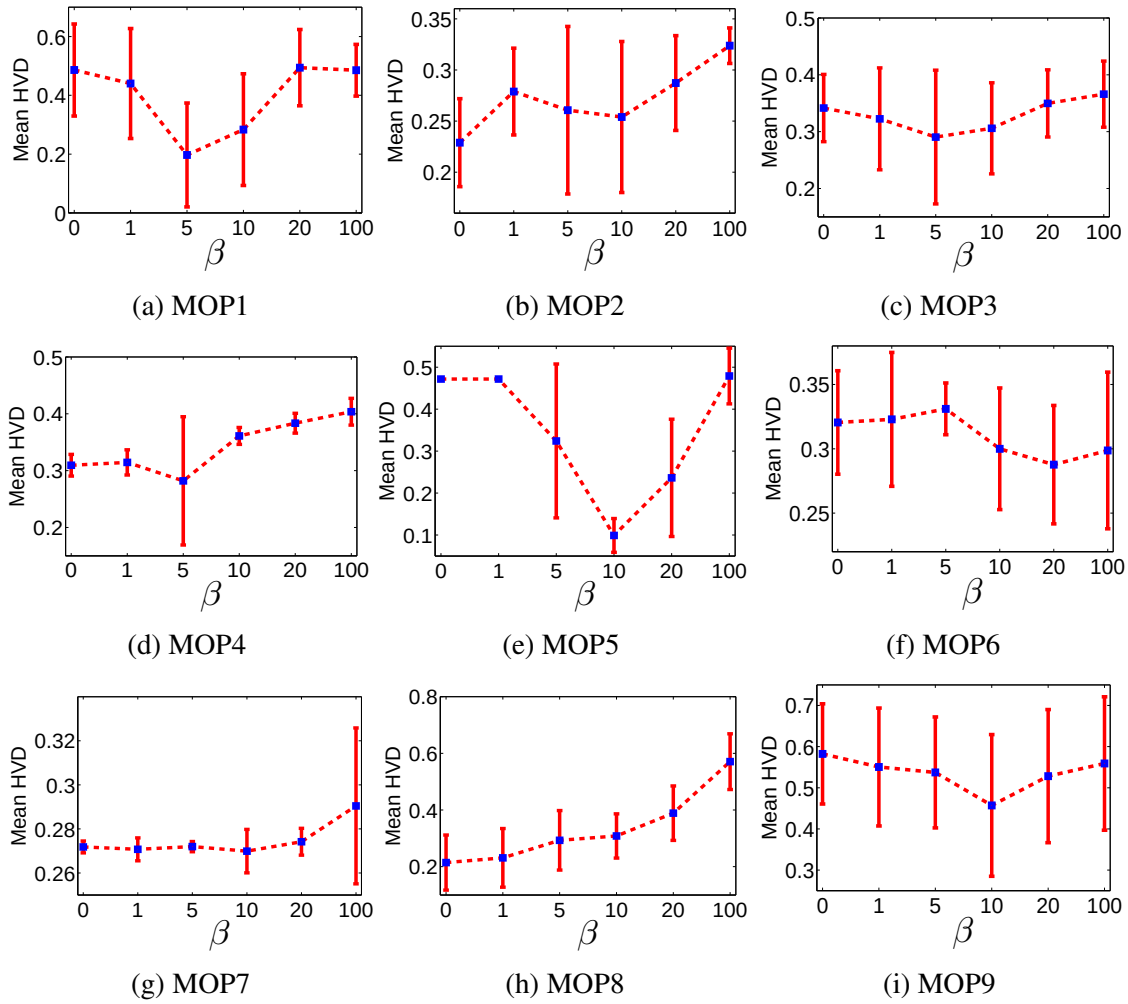
Fig. 4.14 Mean Δ_p values obtained by PSF with different β settings.

4.4.1 Adaptive Scalarizing Strategy

While the proposed scalarizing functions are helpful for maintaining population diversity, it may decrease the convergence performance. A small α value in both MSF and PSF are beneficial to convergence, but it is very likely to cause loss of diversity. This is just the case with TCH, which struggles to recover from the loss of diversity for hard problems. Without any information about problem properties *a priori*, it is plausible to emphasize diversity at the early stage of search and then gradually emphasize convergence at the late stage. To this end, we propose an adaptive strategy to adjust the value of α (line 7 of Algorithm 4.1). As a result, Eq.(4.3) is rewritten as follows:

$$\alpha = \beta(1 - \frac{gen}{MaxGen})\Big\{M \min_{1\leq i\leq M}(w_i)\Big\}, \tag{4.6}$$

where *gen* is the current generation number, and *MaxGen* is the maximum number of generations. It is clear that, for each subproblem, α is decreased linearly as the evolution

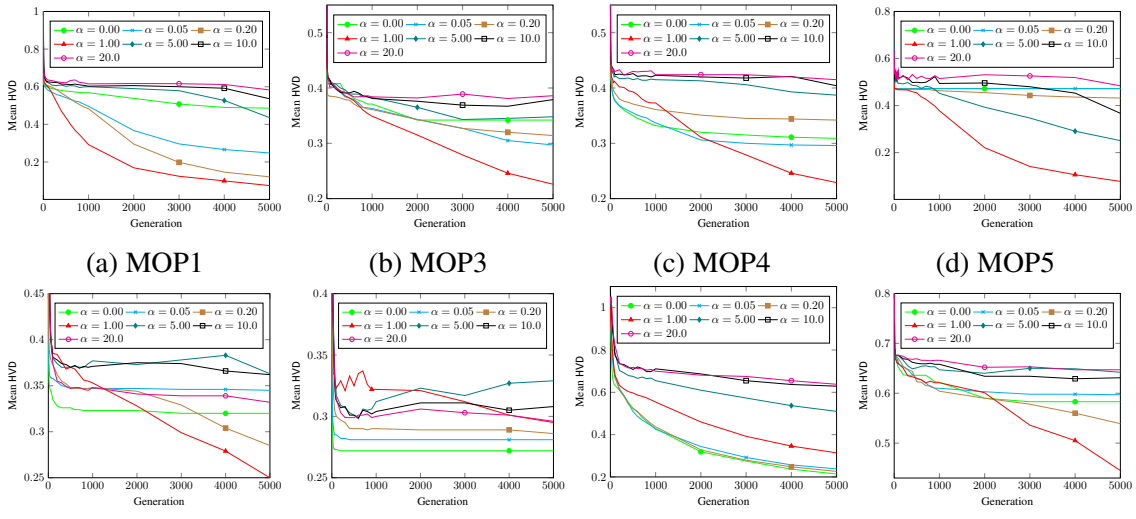


(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

(g) MOP7 (h) MOP8 (i) MOP9

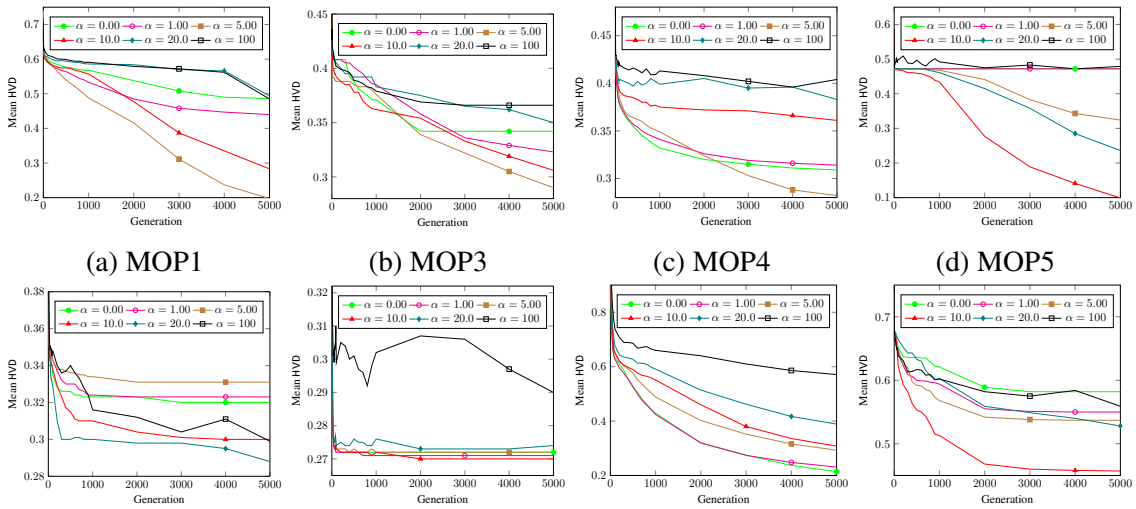
Fig. 4.15 Mean HVD values obtained by PSF with different β settings.



(a) MOP1 (b) MOP3 (c) MOP4 (d) MOP5

(e) MOP6 (f) MOP7 (g) MOP8 (h) MOP9

Fig. 4.16 Evolution curve of the mean HVD metric obtained by MSF with different β settings.



(a) MOP1 (b) MOP3 (c) MOP4 (d) MOP5

(e) MOP6 (f) MOP7 (g) MOP8 (h) MOP9

Fig. 4.17 Evolution curve of the mean HVD metric obtained by PSF with different β settings.

proceeds and becomes zero at the end of search. This means, both MSF and PSF gradually degenerate to TCH, and in this process the improvement region for each subproblem is gingerly increased, resulting in steady-state de-emphasis of diversity and speed-up of convergence simultaneously. Note that, any adaptive decreasing strategy (whatever is linear or nonlinear) can be used as long as it can reduce α gradually. The linear strategy is adopted here because it is very simple and meets the requirement of reducing α well.

4.4.2 Reproduction Operation

Reproduction operation (lines 9–10 of Algorithm 4.1) includes mating pool selection and genetic recombination. In many MOEA/D variants [120, 176], a probability parameter δ is adopted to select a mating pool from either the neighbourhood of solutions or the whole population. The main purpose for this is to increase population diversity. However, this induces the difficulty in tuning of such an extra parameter. Since we have introduced advanced scalarizing functions that can keep diversity well, we discourage the use of the probability parameter and simply set the mating pool as the neighbourhood of solutions.

When mating parents are randomly selected from the mating pool, the next step is to perform genetic operators on the mating parents to generate offspring. In this work, we use the adaptive recombination operator [123] as our genetic operator. The adaptive operator was also used in [127] and showed good performance for hard problems.

4.4.3 Replacement Operation

Replacement operation (lines 12–21 of Algorithm 4.1) is a key step in many MOEA/D variants. It is related to what and how subproblems can be updated. If a new solution is

Algorithm 4.1: The eMOEA/D Framework

Input: stopping criterion (*MaxGen*), population size (N), neighbourhood size (T), replacement size (n_r);
Output: approximated Pareto-optimal set P;

Generate a uniform spread of N weight vectors: $\{w^1, w^2, \ldots, w^N\}$ and then compute the T closest weight vectors to each weight vector by the Euclidean distance. For each w^i, set $B(i) = \{i_1, \ldots, i_T\}$ where $w^{i_1}, \ldots, w^{i_T}$ are the T closest weight vectors to w^i;
Generate an initial population $P = \{x^1, \ldots, x^N\}$ by uniformly randomly sampling from the decision space;
Initialize *ideal* and *nadir* points, i.e., z^* and z^{nad};
Choose a scalarizing function SF for MOEA/D;
$gen \leftarrow 1$;
while $gen \leq MaxGen$ **do**
    Update α for the selected SF according to Eq. (4.6);
    for $i \leftarrow 1$ **to** N **do**
        Randomly select indexes r_1 and r_2 from $B(i)$;
        Apply genetic operators on individuals x^{r_1}, x^{r_2} to produce a new solution y;
        Evaluate the objective vector of y, and update z^*;
        Find the T most suitable subproblems for y: $S = \{s_1, s_2, \ldots, s_T\}$;
        $c \leftarrow 0$;
        for $j \leftarrow 0$ **to** T **do**
            if $g^{SF}(y|w^{s_j}, z^*) < g^{SF}(x^{s_j}|w^{s_j}, z^*)$ **then**
                $x^{s_j} \leftarrow y$ and $c \leftarrow c + 1$;
            end
            if $c \geq n_r$ **then**
                break;
            end
        end
    end
    Update z^{nad} using P, $gen \leftarrow gen + 1$;
end

not suitable for subproblems that are chosen to be updated, then both population diversity and convergence can be negatively affected. To this end, various replacement strategies have been proposed [176, 198]. The main idea behind these strategies is to find the most suitable subproblem for a newly-generated individual y and then conduct replacement within the neighbourhood of this subproblem.

However, these strategies fail to consider that the individual y may not be good for the neighbouring subproblems of the most suitable subproblem. If the individual does not improve any solution of the neighbourhood of the most suitable subproblem but does improve other subproblems outside the neighbourhood, it should enter the population. In other words, the replacement range should be gingerly elaborated. In this work, the

replacement range is composed of the most T suitable subproblems. It is calculated as follows. First, $g^{SF}(y|w^i,z^*)$ is computed for each $1 \leq i \leq N$. Second, all the $g^{SF}(y|w^i,z^*)$ values are sorted in ascending order. Then, the subproblems corresponding to the first T smallest values are regarded as the replacement range $S = \{s_1, s_2, \ldots, s_T\}$, with s_1 being the first most suitable and s_T being the T-th most suitable (line 12 of Algorithm 4.1).

The replacement procedure (lines 14–21 of Algorithm 4.1) is executed on the ordered replacement range $S = \{s_1, s_2, \ldots, s_T\}$ one by one. Like its predecessor [120], the proposed eMOEA/D framework allows at most n_r solutions to be replaced by a newly generated solution.

4.5 Experimental Studies

The experiment in this section is designed for two purposes. One is to verify the proposed eMOEA/D. The other purpose is to deeply analyze the performance of other existing decomposition-based MOEAs and discuss the applicability of some recently-developed many-objective MOEAs in multiobjective optimization.

4.5.1 Compared Algorithms and Parameter Settings

Algorithms for comparison consist of decomposition-based and dominance-based MOEAs. The decomposition-based MOEAs are MOEA/D with TCH [120], ACD [175], AGR [176], DU [185], STM [122], and M2M [127] schemes. The dominance-based MOEAs are PICEA-g [172] and NSGA-III [43]. Note that, NSGA-III can be seen as a mixture of decomposition-based and dominance-based methods. For notational convenience, MSF* and PSF* denote the proposed eMOEA/D with MSF and PSF, respectively.

Key parameters in each compared algorithm remain the same as in the referenced papers. The population size, stopping criterion, and other important parameters are kept the same as in Section 4.3. All the algorithms use the adaptive operator [123] as the recombination operator. The key factor β in MSF* and PSF* is set to 1 and 10, respectively, based on the previous experimental study.

4.5.2 Experimental Results and Analysis

In this subsection, the Δ_p and HVD results are reported as IGD and Δ_p have shown consistent performance assessment in the previous experiment. Tables 4.1 and 4.2 show the best, median, and worst values of Δ_p and HVD over 30 independent runs, respectively. The best values obtained by one of the ten algorithms are highlighted in bold face. The differences between the approximations are assessed by the Wilcoxon rank-sum test [179] at the 0.05 significance level, with the standard Bonferroni correction [1] to deal with the

Table 4.1 Best, median, and worst Δ_p values obtained by different algorithms

Prob.	MSF*	PSF*	TCH	ACD	AGR	DU	STM	M2M	PICEA-g	NSGA-III
	9.14E-03	9.38E-03	7.37E-02	9.50E-03	1.86E-02	1.01E-02	**8.65E-03**	1.46E-02	4.10E-01	1.12E-02
MOP1	**9.46E-03**	9.68E-03	3.97E-01‡	9.77E-03	1.99E-02‡	1.04E-02‡	9.98E-02‡	1.60E-02‡	4.17E-01‡	1.16E-02‡
	9.95E-03	**9.92E-03**	4.11E-01	9.95E-03	2.63E-02	1.17E-02	3.96E-01	1.75E-02	4.21E-01	4.75E-02
	4.60E-03	4.63E-03	1.89E-01	4.62E-03	7.65E-03	5.47E-03	4.72E-03	1.18E-02	3.60E-01	6.01E-03
MOP2	**4.67E-03**	4.67E-03	2.35E-01‡	4.67E-03	2.26E-02‡	5.74E-03‡	7.47E-03‡	1.36E-02‡	3.73E-01‡	1.59E-02‡
	4.86E-03	5.58E-03	4.12E-01	4.87E-03	2.02E-01	7.60E-03	2.76E-01	1.22E-01	4.12E-01	1.44E-01
	5.65E-03	5.24E-03	4.63E-01	**5.11E-03**	1.05E-02	6.52E-03	4.63E-01	1.28E-02	4.63E-01	6.92E-03
MOP3	6.02E-03	6.00E-03	4.93E-01‡	**5.98E-03**	3.49E-02‡	8.37E-03‡	4.63E-01‡	1.55E-02‡	4.63E-01‡	4.11E-01‡
	3.40E-02	4.04E-02	6.36E-01	**8.77E-03**	1.41E-01	3.76E-02	4.76E-01	2.87E-02	4.76E-01	4.63E-01
	6.65E-03	8.43E-03	3.04E-01	8.28E-03	7.08E-03	6.72E-03	6.60E-03	1.28E-02	3.38E-01	6.76E-03
MOP4	**1.31E-02**	1.31E-02	3.19E-01‡	1.33E-02	1.45E-02‡	1.32E-02	2.81E-01‡	1.55E-02‡	3.64E-01‡	2.89E-01‡
	2.52E-02	2.49E-02	3.74E-01	2.04E-02	9.64E-02	1.88E-02	2.90E-01	2.87E-02	3.87E-01	2.97E-01
	8.58E-03	9.11E-03	2.95E-01	1.07E-02	1.80E-02	1.54E-02	2.64E-02	1.81E-02	2.25E-01	1.88E-02
MOP5	3.11E-02	**1.26E-02**	6.28E-01‡	1.27E-02◇	2.32E-02‡	1.69E-02◇	1.92E-01‡	2.65E-02‡	3.48E-01‡	2.84E-02‡
	1.21E-01	4.14E-02	1.08E+00	1.53E-01	4.61E-02	**3.52E-02**	1.92E-01	2.12E-01	5.78E-01	2.23E-01
	4.30E-02	4.34E-02	2.83E-01	1.76E-01	8.15E-02	5.15E-02	5.69E-02	1.29E-01	2.83E-01	5.05E-02
MOP6	**4.35E-02**	4.40E-02	3.59E-01‡	2.65E-01‡	1.62E-01‡	6.10E-02‡	2.49E-01‡	1.76E-01‡	3.59E-01‡	2.06E-01‡
	4.69E-02	4.90E-02	3.59E-01	5.37E-01	2.66E-01	1.49E-01	2.83E-01	1.08E+00	3.59E-01	2.50E-01
	7.78E-02	**6.78E-02**	3.69E-01	3.25E-01	2.83E-01	8.70E-02	3.24E-01	1.89E-01	4.21E-01	3.25E-01
MOP7	8.30E-02	**7.21E-02**	4.21E-01‡	3.59E-01‡	3.59E-01‡	1.01E-01†	3.59E-01‡	2.46E-01‡	4.21E-01‡	3.25E-01‡
	1.34E-01	2.55E-01	4.21E-01	4.21E-01	4.21E-01	1.84E-01	4.21E-01	3.80E+00	4.21E-01	4.22E-01
	4.13E-02	4.10E-02	6.22E-02	1.73E-01	6.21E-02	7.01E-02	4.54E-02	2.29E-01	4.73E-01	4.43E-02
MOP8	4.71E-02	4.58E-02	1.65E-01‡	3.00E-01‡	2.10E-01‡	7.78E-02‡	5.40E-02‡	6.43E-01‡	1.36E+00‡	**4.56E-02**
	7.82E-02	8.74E-02	4.61E-01	4.14E-01	4.26E-01	9.30E-02	4.04E-01	1.13E+00	1.67E+00	**4.75E-02**
	7.01E-02	8.23E-02	3.24E-01	1.69E-01	1.64E-01	8.79E-02	3.24E-01	2.02E-01	3.26E-01	3.25E-01
MOP9	**7.78E-02**	9.17E-02	5.11E-01‡	3.67E-01‡	2.24E-01‡	9.25E-02†	3.25E-01‡	3.36E-01‡	5.09E-01‡	3.25E-01‡
	9.30E-02	2.40E-01	7.47E-01	9.78E-01	3.78E-01	2.13E-01	4.09E-01	1.06E+00	6.75E-01	3.25E-01

† and ◇ indicate MSF* and PSF* significantly outperform the corresponding algorithm, respectively.

‡ indicates both MSF* and PSF* significantly outperform the corresponding algorithm.

Table 4.2 Best, median, and worst HVD values obtained by different algorithms

Prob.	MSF*	PSF*	TCH	ACD	AGR	DU	STM	M2M	PICEA-g	NSGA-III
	1.39E-02	1.43E-02	7.66E-02	1.59E-02	2.73E-02	1.53E-02	**1.32E-02**	2.48E-02	5.81E-01	1.64E-02
MOP1	**1.43E-02**	1.46E-02	5.46E-01‡	1.61E-02‡	2.91E-02‡	1.58E-02‡	6.80E-02‡	2.70E-02‡	5.97E-01‡	1.71E-02‡
	1.50E-02	**1.49E-02**	5.82E-01	1.64E-02	3.56E-02	1.69E-02	5.39E-01	2.88E-02	6.08E-01	3.95E-02
	6.30E-03	6.48E-03	1.48E-01	6.90E-03	1.02E-02	8.39E-03	6.42E-03	1.76E-02	2.85E-01	8.97E-03
MOP2	**6.39E-03**	6.55E-03	2.22E-01‡	8.29E-03‡	1.82E-02‡	8.64E-03‡	6.78E-03	1.91E-02‡	3.12E-01‡	9.79E-03‡
	6.53E-03	8.36E-03	3.33E-01	9.51E-03	2.64E-01	1.20E-02	1.84E-01	1.36E-01	3.33E-01	9.81E-02
	6.08E-03	6.29E-03	2.15E-01	6.13E-03	1.03E-02	8.12E-03	2.15E-01	1.26E-02	2.15E-01	9.38E-03
MOP3	**6.76E-03**	6.80E-03	3.48E-01‡	7.16E-03†	1.24E-02‡	8.69E-03‡	2.15E-01‡	1.53E-02‡	2.15E-01	2.10E-01‡
	7.94E-03	1.45E-02	4.15E-01	1.03E-02	1.55E-01	3.19E-02	2.82E-01	1.88E-02	2.82E-01	2.15E-01
	3.45E-03	6.48E-03	2.78E-01	7.21E-03	4.35E-03	4.26E-03	**3.06E-03**	8.72E-03	3.57E-01	5.60E-03
MOP4	9.45E-03	**9.57E-03**	3.12E-01‡	1.33E-02‡	1.31E-02	1.02E-02	2.29E-01	1.48E-02‡	3.77E-01◇	2.58E-01‡
	2.10E-02	**1.54E-02**	3.48E-01	2.82E-02	1.25E-01	1.33E-02	2.45E-01	2.47E-02	3.96E-01	2.87E-01
	1.38E-02	1.45E-02	4.72E-01	1.70E-02	2.73E-02	2.33E-02	2.94E-02	2.92E-02	3.13E-01	2.50E-02
MOP5	1.63E-02	**1.60E-02**	4.72E-01‡	1.77E-02	3.12E-02‡	2.46E-02‡	3.13E-01‡	3.55E-02‡	4.72E-01‡	3.14E-02‡
	2.09E-02	**1.87E-02**	4.72E-01	1.88E-02	4.63E-02	3.95E-02	3.13E-01	5.45E-02	4.72E-01	3.34E-01
	5.15E-02	5.25E-02	2.46E-01	1.20E-01	9.28E-02	6.23E-02	5.77E-02	1.53E-01	2.40E-01	6.56E-02
MOP6	**5.25E-02**	5.31E-02	3.47E-01‡	2.17E-01‡	2.00E-01‡	7.66E-02‡	1.83E-01‡	1.91E-01‡	3.38E-01‡	1.74E-01‡
	6.32E-02	6.50E-02	3.47E-01	3.47E-01	2.91E-01	1.20E-01	2.46E-01	2.96E-01	3.39E-01	2.00E-01
	9.58E-02	1.08E-01	2.60E-01	1.70E-01	1.58E-01	9.67E-02	1.58E-01	1.63E-01	2.66E-01	1.77E-01
MOP7	**1.03E-01**	1.11E-01	2.72E-01‡	2.62E-01‡	2.65E-01‡	1.13E-01	2.55E-01‡	2.14E-01‡	2.66E-01‡	1.81E-01‡
	1.60E-01	2.10E-01	2.72E-01	2.98E-01	2.72E-01	1.88E-01	2.72E-01	1.20E+00	2.67E-01	2.87E-01
	4.44E-02	**4.38E-02**	8.65E-02	6.59E-02	6.94E-02	9.25E-02	5.62E-02	2.63E-01	4.75E-01	5.31E-02
MOP8	5.81E-02	**5.46E-02**	2.02E-01‡	1.16E-01‡	1.01E-01‡	1.01E-01‡	6.88E-02‡	4.29E-01‡	6.11E-01‡	5.74E-02
	9.58E-02	1.09E-01	3.81E-01	1.54E-01	1.52E-01	1.71E-01	3.15E-01	7.30E-01	7.28E-01	**6.24E-02**
	9.25E-02	**8.05E-02**	1.59E-01	1.32E-01	1.04E-01	9.71E-02	1.58E-01	2.05E-01	1.70E-01	1.76E-01
MOP9	**1.01E-01**	1.03E-01	6.15E-01‡	3.96E-01‡	1.72E-01‡	1.03E-01‡	1.59E-01‡	2.48E-01‡	6.08E-01‡	1.79E-01‡
	1.71E-01	3.25E-01	6.76E-01	6.20E-01	3.58E-01	2.23E-01	4.78E-01	4.63E-01	6.76E-01	1.81E-01

† and ◇ indicate MSF* and PSF* significantly outperform the corresponding algorithm, respectively.
‡ indicates both MSF* and PSF* significantly outperform the corresponding algorithm.

problem of the higher probability of Type I errors in multiple comparisons. Signs of †, ⋄ and ‡ in superscript form on median values indicate the significance of the proposed methods.

From the tables, we can obtain the following observations:

1. Compared with the predecessor TCH, all the other MOEA/D variants show improvements on the MOP problems in some sense. The improvements are obvious on the five bi-objective problems, as indicated by the Δ_p and HVD values. However, on the four tri-objective problems, ACD, AGR, STM, and M2M do not have any advantage over TCH, as their Δ_p and HVD values are very similar to those of TCH.

2. The two dominance-based MOEAs, i.e, PICEA-g and NSGA-III, are generally no better than the decomposition-based ones on these problems, and NSGA-III performs better than PICEA-g in terms of the metrics.

3. MSF* and PSF* significantly outperform the other algorithms on the majority of cases. On the bi-objective cases, ACD also shows comparable performance in terms of Δ_p. However, ACD degrades dramatically and performs worse than most of the algorithms on the tri-objective cases.

4. DU, PICEA-g, and NSGA-III, who were originally devised for many-objective optimization, show little advantage for solving the test problems. Specifically, both DU and NSGA-III work on the bi-objective cases but deteriorate significantly on the tri-objective cases. PICEA-g, however, performs poorly on all the test problems.

5. Apart from dimensionality, other characteristics of the test problems also affect the compared algorithms' performance. Taking MOP6 and MOP8 for example, they have the same PF shape except that the former has local attractors in boundary regions of the PF whereas the latter has those in intermediate regions. Judging by HVD, MOPs with boundary attractors are easier than those with intermediate ones for MSF*, PSF*, DU, M2M, and PICEA-g, but seem more difficult for STM and NSGA-III. Additionally, if we compare the algorithms' performance on MOP7 and MOP9, we can see that most of the algorithms degrade when the number of local attractors in boundary regions decrease. This is understandable because the decrease of the number of local attractors reduces the chance of finding boundary solutions on the PF. However, it seems that such features do not influence too much the performance of the proposed methods, as the obtained HVD results vary little.

For the inspection of the real performance of these algorithms, we also plot their PF approximations on several selected test problems in Figs. 4.18 to 4.21. From the plots, we can observe that some algorithms (e.g., ACD, AGR, DU and M2M) converge slowly and some (e.g., TCH, PICEA-g, and NSGA-III) cannot maintain diversity well. Also, most

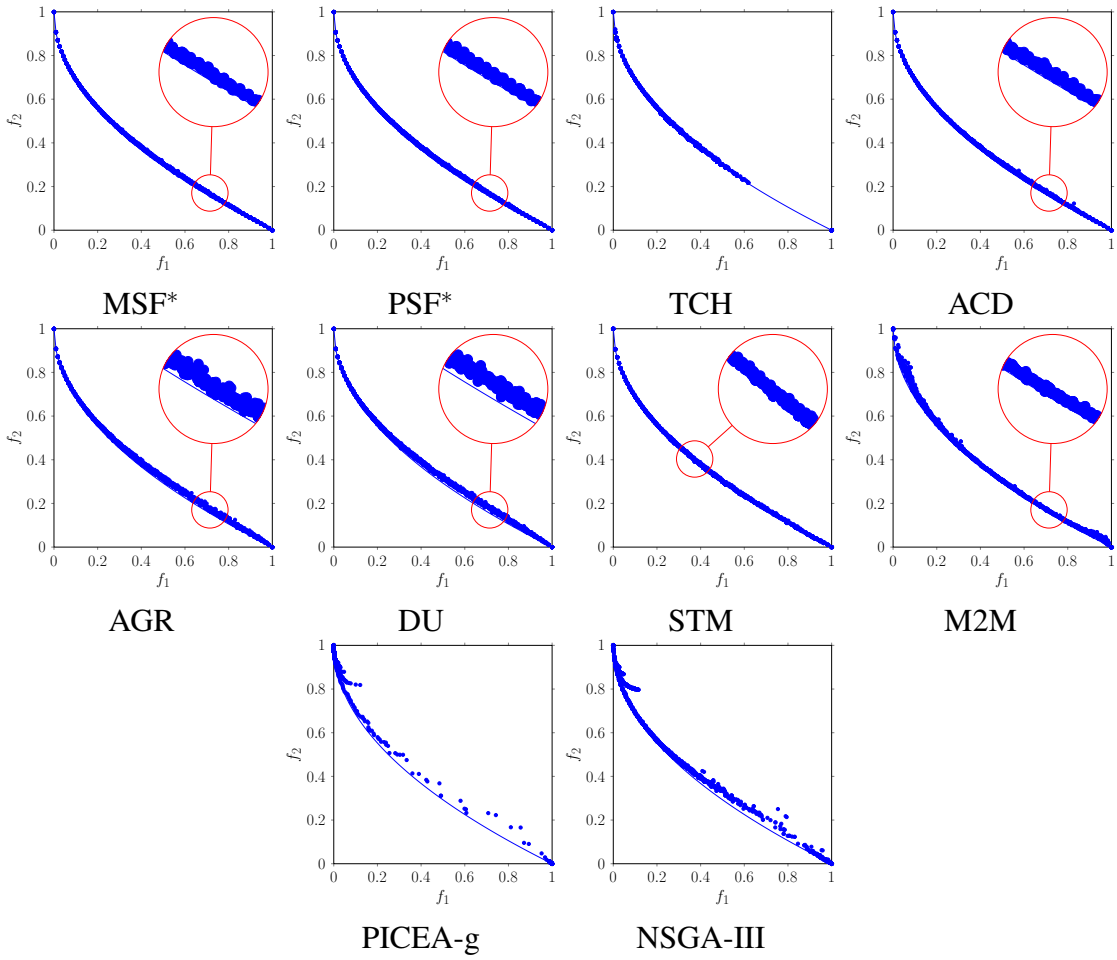


Fig. 4.18 PF approximations obtained by ten algorithms for MOP1.

of them cannot work well on tri-objective problems, particularly on MOP9. Nevertheless, both MSF* and PSF* show better performance compared with the other algorithms.

4.6 Further Investigations

4.6.1 Influence of Mating Selection

Many MOEA/D variants are developed based on the predecessor [120], and thus inevitably inherit a parameter δ that is the probability of choosing mating parents from the neighbourhood of subproblems rather than the whole population. δ is of undisputed importance in the predecessor because it helps much to enhance diversity. However, most MOEA/D variants take for granted that δ is always beneficial.

Here, δ from 0 to 1, with an increment of 0.2, is tested in the framework of MSF*. Fig. 4.22 shows the influence of δ on the obtained HVD values. It is clear that a large value of δ is roughly good for all the problems. This indicates that the higher probability of choosing subproblems' neighbourhood as mating range, the better the resulting perfor-

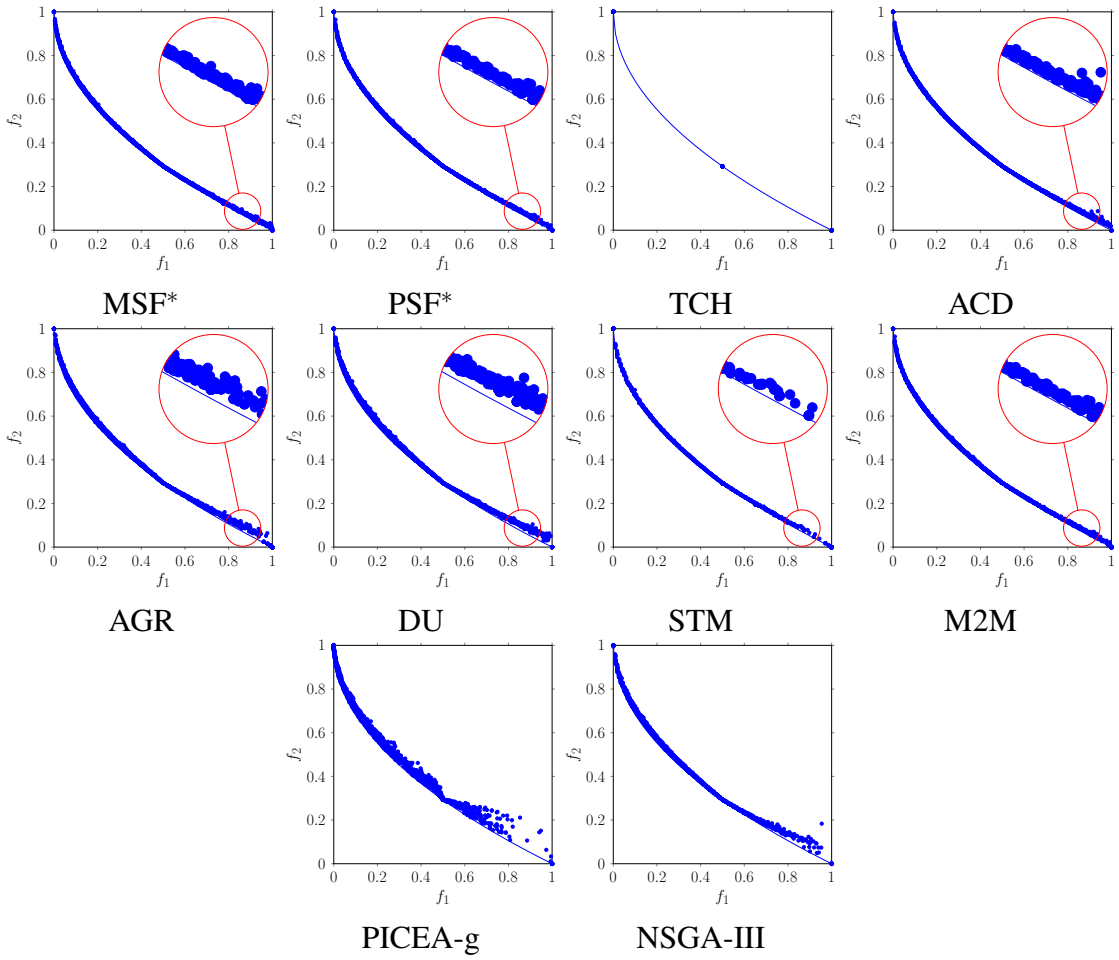


Fig. 4.19 PF approximations obtained by ten algorithms for MOP5.

mance. This is probably because our methods have already soundly considered diversity within scalarizing functions, and in this situation using as much neighbourhood mating as possible to enhance local search helps the convergence of population. Thus, in our eMOEA/D framework we discourage the use of δ and simply select only the neighbourhood as the mating range.

4.6.2 Influence of Replacement Strategies

It has been shown that the performance of decomposition-based MOEAs can be significantly affected by replacement strategies [122, 176]. To achieve efficient population replacements, MOEA/D needs to find an appropriate replacement range. Most often, the replacement range is the neighbourhood of the best matched subproblem [122]. In AGR [176], the best matcher is the one having the minimal scalarizing function value, whereas in other MOEA/D variants [198], the best matcher is considered the one that can be improved most among all the subproblems. However, in our MOEA/D framework, the replacement range consists of the top T best matchers. In this subsection, we investigate

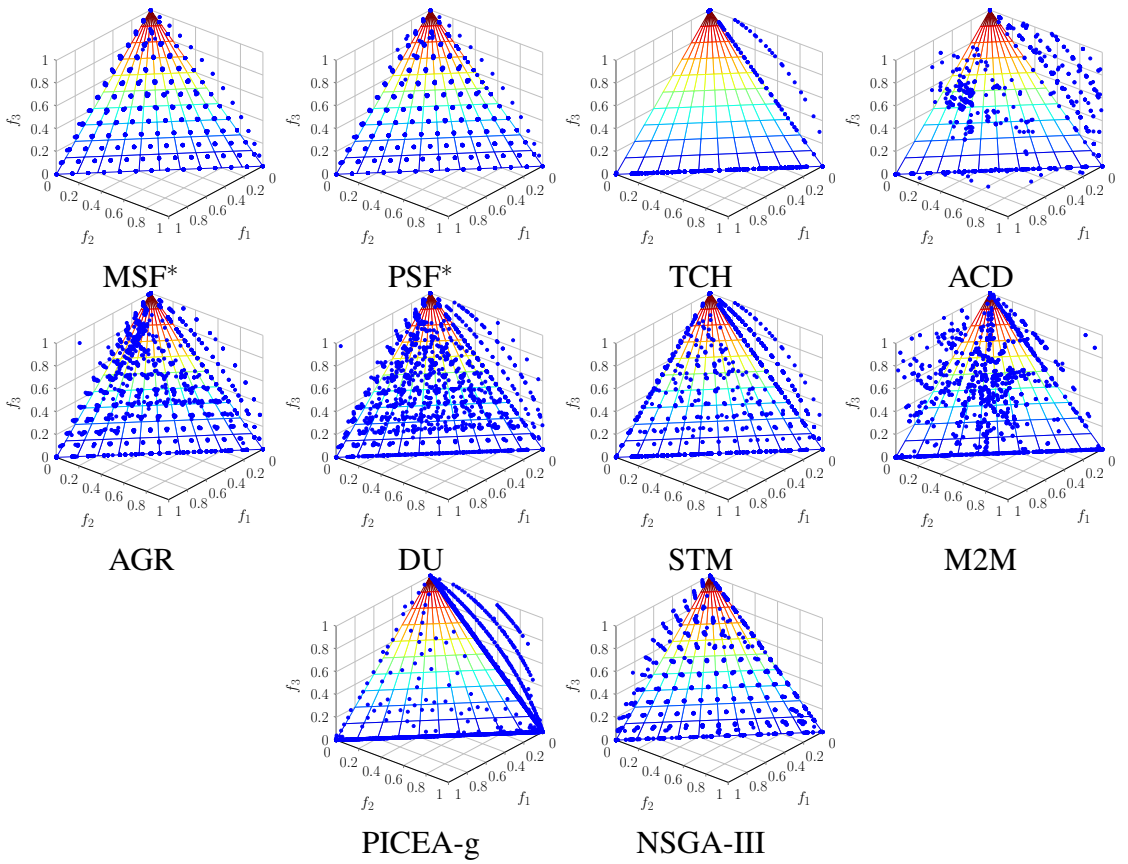


Fig. 4.20 PF approximations obtained by ten algorithms for MOP6.

the influence of different replacement strategies. We compare our replacement strategy (called TMS) with that of AGR (called NMS). To study the influence of the definition of matchers, we also include a replacement strategy whose replacement range is composed of the top T most improved subproblems, and this strategy is called TIS. To assess the efficiency of replacement strategies, we define the replacement rate (RR) of the population in every generation as

$$RR = \frac{N_T}{Nn_r}, \tag{4.7}$$

where N_T is the total number of replacements that occur in the considered generation, and as has been stated before, N and n_r are the population size and the maximal allowable number of replacements, respectively. The larger the value of RR, the better the replacement efficiency.

The three above-mentioned strategies have been tested in MSF* on four selected problems. Fig. 4.23 plots the evolution curves of the mean RR value of 100 independent runs. It can be observed that replacements occur mainly at the early stage of search and the occurrence drops to near zero as the population is near the PF. Another observation is that NMS performs worse than TMS and TIS in terms of the RR value, and TMS is better than TIS on the two bi-objective problems but is similar to TIS on the two tri-objective

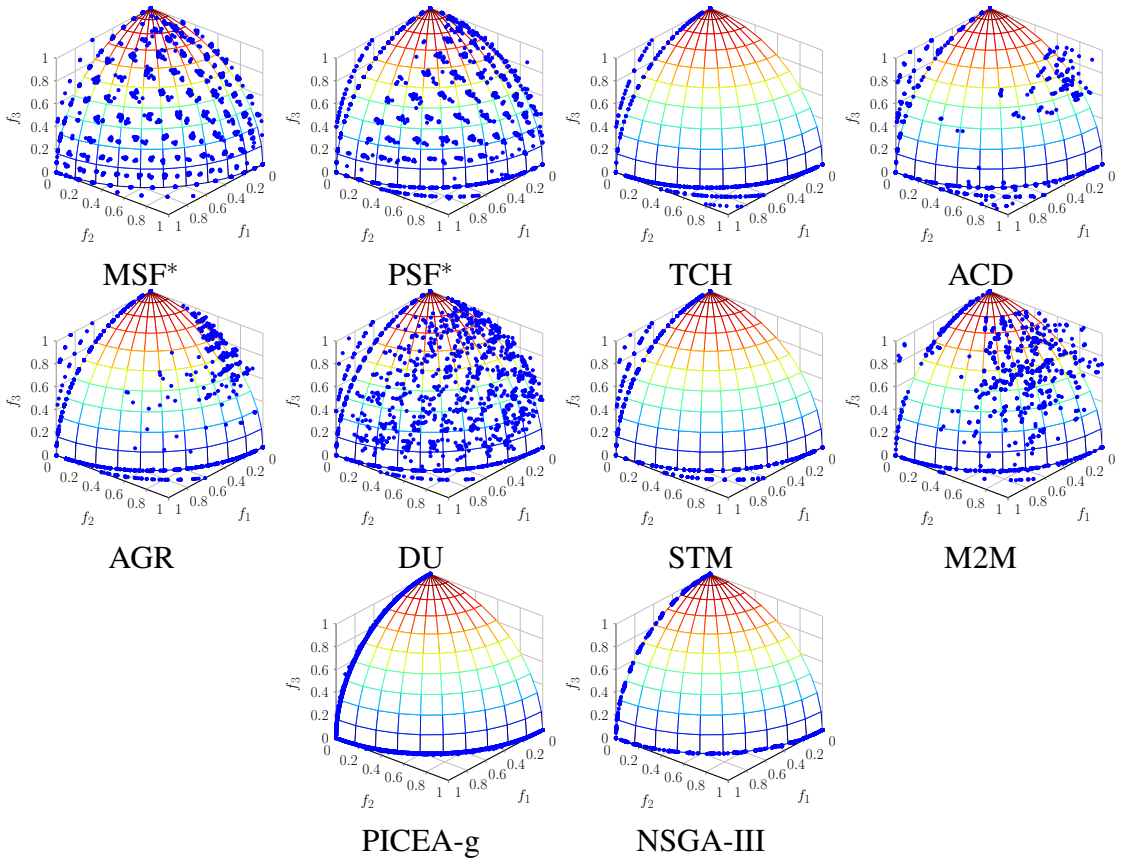


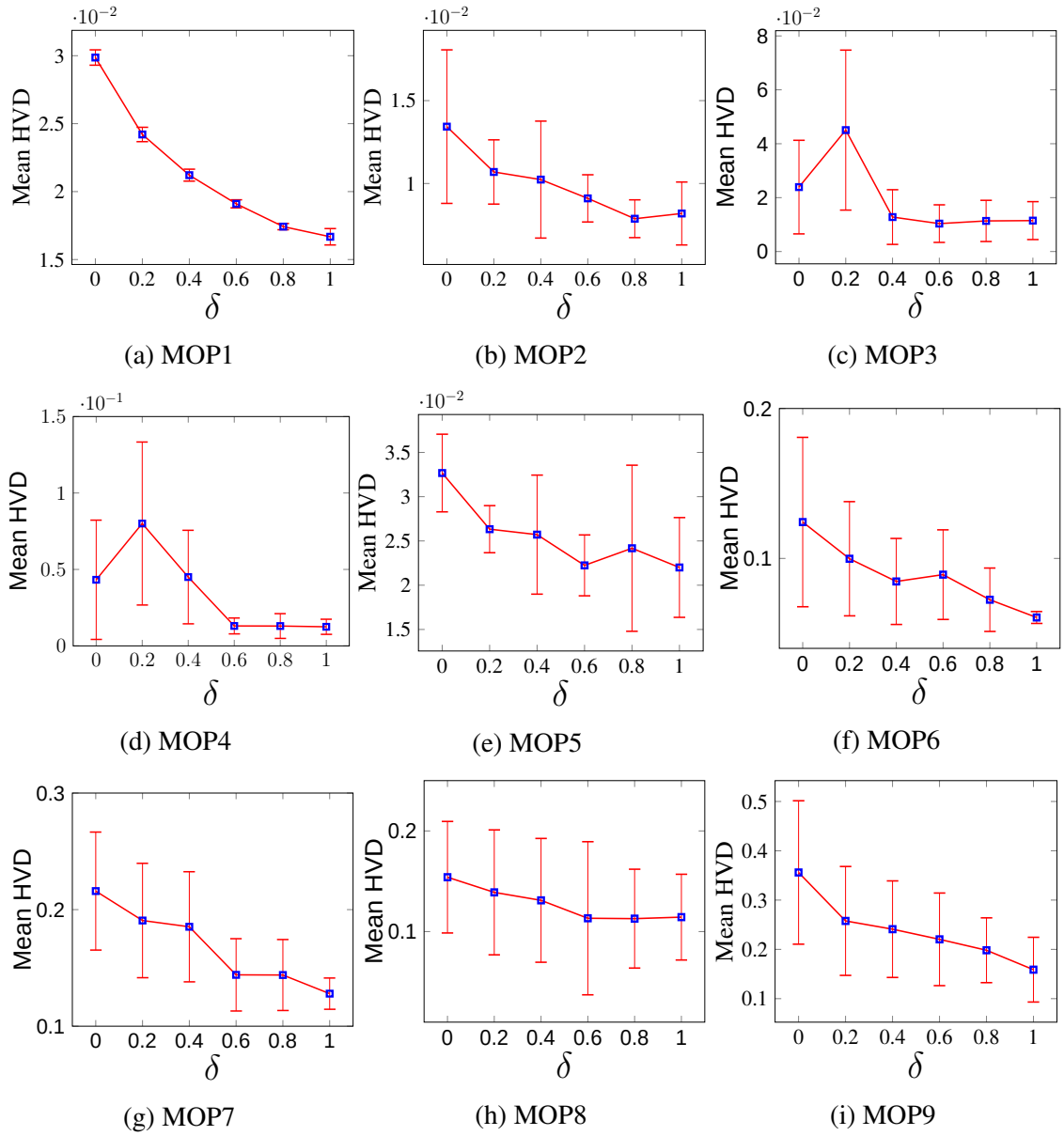
Fig. 4.21 PF approximations obtained by ten algorithms for MOP9.

problems. The high replacement rate of TMS helps the population evolve fast. This observation can be also used to partly explain why MSF* and PSF* perform better than AGR in the previous experiments.

4.6.3 Comparison of PSF and PBI

Since PSF and PBI have similar contour lines, it is interesting to make a comparison between them. Both PSF and PBI use our eMOEA/D framework, and accordingly the comparison objects are actually PSF* and the proposed eMOEA/D with PBI. They are investigated in two convex problems, i.e., F1 and convex DTLZ2 (CDTLZ2), mentioned in [90]. The convex problems are chosen here because it has been increasingly recognized that irregularly-shaped problems (particularly convex ones) influence much the performance of scalarizing functions [90, 139, 147, 174]. The penalty factor of PBI is set to 5, according to [188]. Other parameter settings remain the same as in Section 4.5.2 except the maximal number of generations is changed to 500.

The whole approximations of 31 independent runs are plotted in Fig. 4.24, and the corresponding Δ_p and HVD values are shown in Table 4.3. Both the considered indicators and the graphical plots clearly illustrate that PSF helps yield better performance than

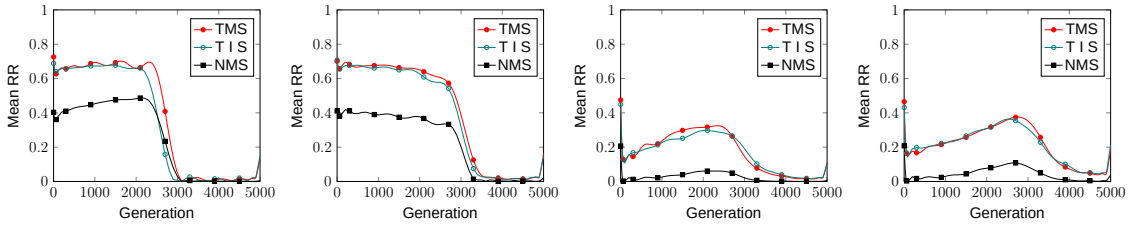


(a) MOP1 (b) MOP2 (c) MOP3

(d) MOP4 (e) MOP5 (f) MOP6

(g) MOP7 (h) MOP8 (i) MOP9

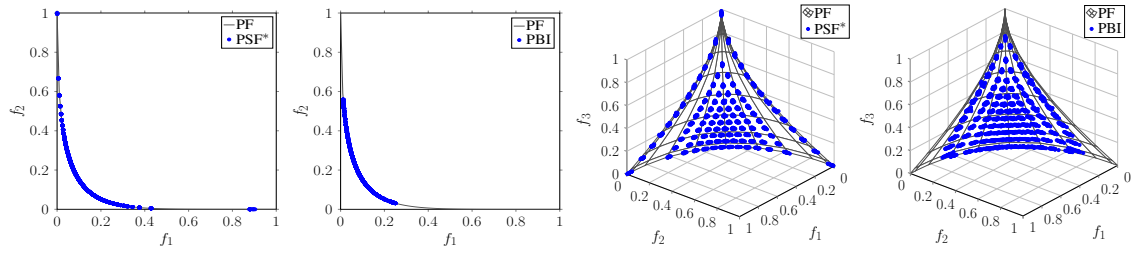
Fig. 4.22 Mean HVD values obtained by MSF* with different δ settings.



(a) MOP1 (b) MOP2 (c) MOP6 (d) MOP7

Fig. 4.23 Evolution curves of the mean RR obtained by different replacement strategies.

PBI. By inspecting closely the approximations in the figure, we can see that PBI favours intermediate regions of the PF and is very likely to miss boundary solutions in convex problems. This is due to the fact that PBI favours solutions with small distances to the



(a) F1 (PSF*) (b) F1 (PBI) (c) CDTLZ2 (PSF*) (d) CDTLZ2 (PBI)

Fig. 4.24 PF approximations of PSF* and PBI over 30 runs on two convex problems.

Table 4.3 Best, median and worst values of Δ_p and HVD obtained by PSF* and PBI

	Δ_p		HVD	
Prob.	PSF*	PBI	PSF*	PBI
F1	**7.53E-02**	2.93E-01	**4.51E-03**	3.34E-02
	7.57E-02	2.93E-01	**4.53E-03**	3.35E-02
	7.87E-02	2.93E-01	**4.63E-03**	3.35E-02
CDTLZ	**4.53E-02**	8.56E-02	**2.12E-02**	2.63E-02
	4.61E-02	9.03E-02	**2.17E-02**	2.90E-02
	4.86E-02	9.52E-02	**2.28E-02**	3.11E-02

ideal point found so far if the penalty factor is not properly set or population diversity receives little emphasis. On the other hand, PSF* has a better coverage than PBI and has the potential to maintain extreme solutions and boundary solutions.

4.6.4 Influence of Recombination Operators

This section investigates the influence of different recombination operators on the performance of our methods. We have compared Li and Liu's operator (named LLX) [123] with simulated binary crossover (SBX) [42] and differential evolution (DE) [138] on the problems MOP1 and MOP2. The investigation is conducted on the MSF* framework, and parameter settings remain the same (parameters in DE are the same as those of MOEA/D-DE).

Table 4.4 provides the HVD results obtained by different recombination operators. It is clear to see that the performance depends largely on the recombination operator chosen for population reproduction. LLX performs the best in terms of the HVD metric, followed by DE, and SBX ranks the last. For visual inspection, we also plot the whole PF approximations over 30 independent runs obtained by each recombination operator, which are shown in Fig. 4.25 and 4.26. It is clear again that LLX shows much better performance than the other two operators. A possible explanation for the high performance of LLX is that LLX can focus on generating diversified individuals at the early stage of the search and then place more emphasis on population convergence as the evolution proceeds. This kind of adaptive strategy helps yield appealing approximations.

Table 4.4 Best, median and worst HVD values obtained by MSF with different recombination operators

Problem	SBX	DE	LLX
MOP1	4.98E-02	3.26E-02	**1.39E-02**
	5.93E-02	3.75E-02	**1.43E-02**
	8.46E-02	4.25E-02	**1.50E-02**
MOP2	1.18E-01	1.02E-02	**6.30E-03**
	2.07E-01	7.35E-02	**6.39E-03**
	3.27E-01	3.27E-01	**6.53E-03**

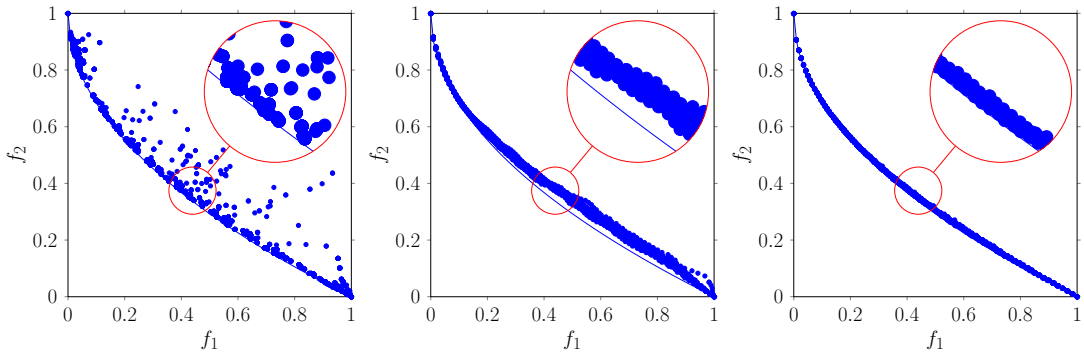


Fig. 4.25 PF approximations obtained by different recombination operators for MOP1.

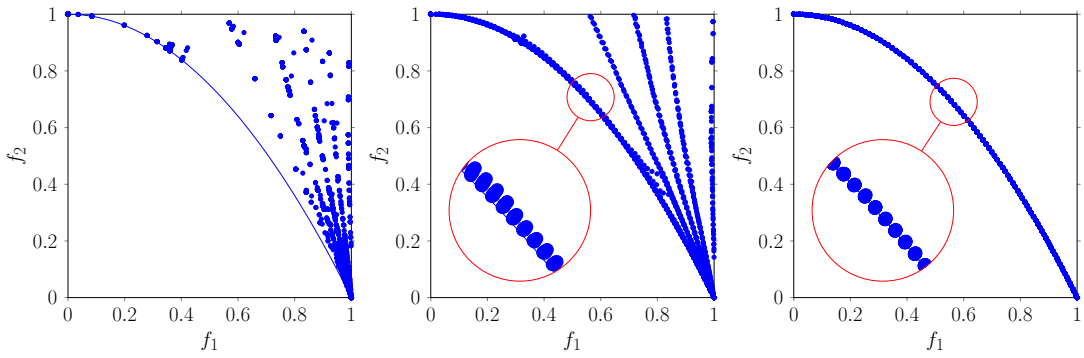


Fig. 4.26 PF approximations obtained by different recombination operators for MOP2.

4.7 Summary

Decomposition-based MOEAs are an important class of methods for multiobjective optimization, and have been frequently shown to work well when proper scalarizing functions are provided. In this chapter, we have proposed two new scalarizing functions which can induce controllable contours.

By adjusting the size of induced improvement regions, the new scalarizing functions can easily manage population diversity. We have studied the influence of the new

scalarizing functions and have demonstrated that the proposed scalarizing functions with proper control of diversity can significantly boost the performance of decomposition-based MOEAs. Additionally, we have proposed an efficient MOEA/D (i.e., eMOEA/D) framework based on new strategies. We have studied the effect of the new strategies. The experimental results have clearly verified the effectiveness of the efficient MOEA/D framework.

In the current work, the proposed eMOEA/D uses a very simple adaptive strategy (i.e., linearly decreasing α in MSF and PSF) to adjust the balance between diversity and convergence at different stages of the search. Despite appealing performance, the adaptive strategy may not be the best choice because different search stages have different (not necessarily linearly-decreasing) convergence or diversity requirements. Further investigations in this direction are beneficial. In our future research, it will be also interesting to investigate the performance of the proposed scalarizing functions in many-objective optimization.

Chapter 5

EAs for Many-objective Optimization Problems

In many real-world applications, optimization problems often involve four or more objectives [87]. Recent studies have suggested that conventional MOEAs are subjected to the scalability challenge, i.e., the performance of these MOEAs degrades dramatically with the increase in the number of objectives. This fact gives rise to a new term, known as many-objective optimization problems (MaOPs), to better refer to those MOPs that have four or more objectives. Note that some communities, such as the multi-criterion decision making (MCDM) [4], do not differentiate between multiobjective and many-objective optimization.

In many-objective optimization, Pareto-based EAs become of limited use as a large portion of population becomes nondominated with regard to each other. In other words, these kinds of algorithms cannot induce sufficient selection pressure toward a set of trade-off solutions. To overcome this difficulty, a straightforward way is to stress solutions' contribution to diversity so that selection pressure can be improved. Following this line, in this chapter we propose a new method to deal with many-objective optimization.

The rest of this chapter is organized as follows. Section 5.1 presents related work and the incentive of this research. In Section 5.2, a new population sorting method is proposed. Section 5.3 describes the framework of the proposed algorithm, together with a detailed description of its components. Section 5.4 presents experimental studies on multiobjective and many-objective optimization. Extensive investigations and discussions regarding the proposed algorithm are provided in Section 5.5. Section 5.6 concludes this work.

5.1 Introduction

In the design of EAs for multiobjective optimization, two goals should be considered: 1) minimizing the gap between candidate solutions and the true PF (convergence) and 2)

maximizing the distribution of candidate solutions (diversity). However, these two goals are generally assumed to be conflicting [201]. In practice, most existing MOEAs achieve convergence by prior Pareto-based sorting of the evolving population and diversity by the additional calculation of individuals' density information. The well-known nondominated sorting genetic algorithm II (NSGA-II) [41] and strength pareto evolutionary algorithm 2 (SPEA2) [202] are representative examples of this method. Taking nondominated sorting [41] for example, it sorts a combined population (R) as follows. First, each individual is compared with all other individuals in R, and all nondominated solutions of R are identified and assigned to front L_1. Then, individuals in L_1 are removed from R, and the remaining individuals in R are compared with each other to determine the nondominated set, which are assigned to front L_2. The procedure is repeated until no individuals are left in R, i.e., all individuals have been assigned to a front.

Fig. 5.1 gives a graphical illustration of the nondominated sorting. The main idea behind the nondominated sorting is to classify the entire combined population into different nondominated fronts according to individuals' convergence. After the nondominated sorting, the new population can be constructed by selecting solutions of different nondominated fronts, one at a time. The selection starts with individuals of the first front L_1 and continues with those of the second front L_2, followed by the rest of the fronts and so on. Since only limited slots are allowable in the new population, not all fronts can be considered. When the last allowed front (e.g., L_l) is being considered, there may exist more individuals in L_l than the remaining slots in the new population. In this situation, niche-preservation strategies, such as crowding distance [41], the-farthest-the-first method [27], and nearest neighbour technique [202], are desirable for selecting the remaining number of individuals from front L_l in order to maintain diversity.

Methods like nondominated sorting actually conduct environmental selection in a convergence-first-and-diversity-second manner [127]. The advantage is that it can provides a fast convergence speed. However, the disadvantage is that this kind of methods may undermine population diversity if well-converged individuals are not diversified. Taking Fig. 5.1 again as an example, diversity loss occurs if only six nondominated individuals are allowed to be preserved. The loss of diversity in the example could further incur evolutionary stagnation where overcrowded boundary regions are overexploited and intermediate regions are left unexplored. Furthermore, convergence-first based methods have encountered great difficulties in many-objective optimization where problems have four or more objectives [43]. In many-objective optimization, the convergence-first-and-diversity-second strategy can be of limited use because the proportion of nondominated solutions is very high and diversity preservation is very likely to be carried out only on nondominated solutions. The population is at the risk of losing diversity and preserved solutions may be far from each other if nondominated solutions are not well distributed. Correspondingly, reproduction operators struggle to generate promising solutions for un-

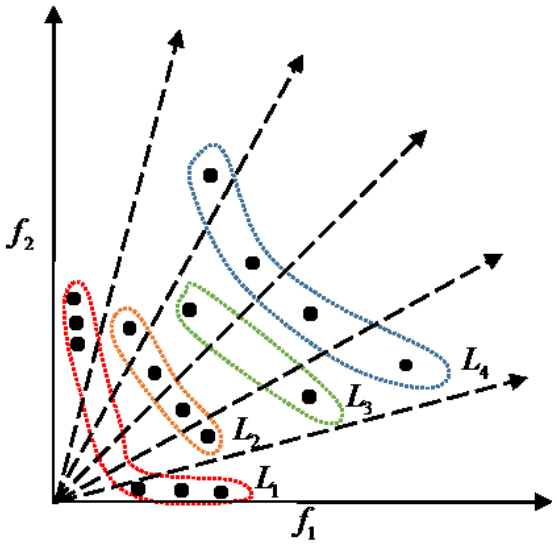


Fig. 5.1 Nondominated Sorting.

explored regions as distant parents are not very effective to generate good offspring solutions in many-objective optimization [43, 106]. In fact, some dominated but promising solutions can contribute to population diversity, and proper use of them can increase the selection pressure in many-objective optimization. In this sense, diversity outweighs convergence and should be emphasized in many-objective optimization.

Bearing this in mind, this chapter proposes a new sorting approach with the aid of a set of diverse reference directions. The approach sorts the population into different fronts, each front representing a level of diversity and convergence. On the basis of this new sorting approach, together with reference directions, a new version of strength Pareto evolutionary algorithm (SPEA), denoted SPEA/R, is proposed for both multiobjective and many-objective optimization. SPEA/R is a substantial extension of early-developed prominent SPEA methods [202, 203].

5.2 Diversity-based Sorting

The proposed sorting method (diversity first and convergence second or DFCS) works as follows. First, the objective space is partitioned into a number of subspaces with the aid of a reference direction set W. Reference directions in W are required to be uniformly distributed. Then, each individual (whose objective values need to be normalized beforehand) in the combined population is associated with a subspace. This can be done by identifying the nearest reference direction to the considered individual. In each subspace, individuals are assigned a fitness value that can reflect its convergence level. Potential fitness assignment approaches for this purpose can be scalarizing functions used in MOEA/D [188], strength fitness in SPEA2 [202], or nondominated ranks in NSGA-II

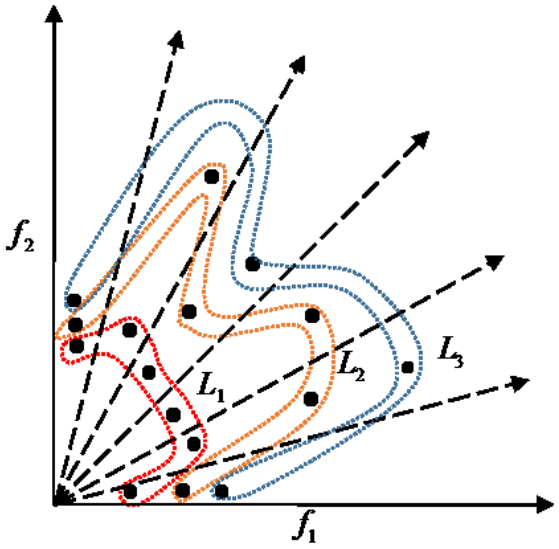


Fig. 5.2 DFCS Sorting.

[41], whichever is the easiest for users to implement. An individual with the best fitness from each subspace is assigned to front L_1. After that, the individuals in L_1 are removed from the subspaces, and another one with the best fitness from each subspace is assigned to front L_2. If multiple solutions have the same fitness, a random one is considered. This procedure continues until each individual in each subspace has been assigned to a front. Note that, in case that a subspace is empty, this subspace is skipped.

Fig. 5.2 illustrates the outcome of DFCS sorting, where population distribution is identical to that of Fig. 5.1. After sorting, the new population can be constructed by selecting solutions of different fronts, one at a time. Similar to the nondominated sorting, not all fronts can be considered due to the limited number of slots in the new population. If the last allowed front (e.g., L_l) has more individuals than the remaining slots, random selection on L_l can be performed to fill up the new population. Note that, it is advisable to use techniques that are helpful for convergence to select individuals from L_l. For example, fitness assignment can be performed on L_l, and individuals with relatively good convergence are priorly selected.

The advantages and disadvantages of the proposed DFCS sorting are summarized as follows.

Advantages As can be seen from Fig. 5.2, the DFCS sorting enhances local diversity in each subspace. As a result, population diversity can be well maintained during the evolution. Besides, population convergence is also properly considered in the course of sorting. The sorting method can provide a good coverage and spread of approximation.

Disadvantages Since the DFCS sorting employs a reference direction set for population partition, the resulting population distribution depends largely on the uniformity of

Algorithm 5.1: Framework of SPEA/R

Input: N (population size)
Output: : approximated Pareto-optimal front

```
1  Generate a diverse reference direction set W: W := Reference_Generation();
2  Create an initial parent population P;
3  while stopping criterion not met do
4      Apply genetic operators on P to generate offspring population P̄;
5      Q := P ∪ P̄;
6      Normalize objectives of members in Q: Q̄ := Objective_Normalization(Q);
7      for each reference direction i ∈ W do
8          Identify members of Q̄ close to i: E(i) := Associate(Q̄, W, i);
9          Calculate fitness values of members in E(i): Fitness_Assignment(E(i));
10     end
11     P := DFCS_Environment_Selection(Q̄, W);
12 end
```

the reference direction set. Also, the convergence speed of algorithms based on DFCS might be affected, due to the emphasis on diversity.

5.3 DFCS-based Strength Pareto EA

The basic framework of the proposed algorithm, called SPEA/R, is based on DFCS sorting and presented in Algorithm 5.1. SPEA/R starts with an initial population and the construction of a predefined set of reference directions, which splits the objective space into a number of independent subregions, helping guide the search toward the whole PF with a good guarantee of population diversity in the objective space. For each generational cycle, on the basis of the preserved parent population, SPEA/R applies genetic operator to reproduce an offspring population, followed by a union of the parent and offspring populations. Then, to make it capable of handling problems with disparately scaled objectives, SPEA/R introduces an objective normalization strategy after the merging of the two populations. Afterwards, each member in the combined population is associated with a reference direction (or a subregion). This way, the combined population members are distributed to different subregions. A novel fitness assignment technique is applied on individuals residing in each subregion. Thereafter, DFCS-based environmental selection is adopted to construct a new parent population for the next generation. In the following subsections, the implementation of each component of SPEA/R will be detailed step by step.

5.3.1 Generation of the Reference Direction Set

Any reference-direction-based MOEA cannot ignore the importance of the setting of reference directions (or weight vectors in [188]). Early MOEA/D algorithms employ a systematic approach, developed by Das and Dennis [39], to generate $H = \binom{p+M-1}{M-1}$ reference directions on a unit simplex for M objectives, where p is the number of divisions considered along each objective coordinate. The systematic approach works well for a small number of objectives, especially for bi-objective problems, where the number of reference directions can be arbitrarily designated. For MaOPs, however, this approach will generate a large number of reference directions if intermediate reference directions (which require $p \geq M$) within the simplex are pursued [43]. This inevitably pushes up the computational burden of MOEAs. To avoid such a situation, a two-layer (boundary and inside layers) approach for objectives over seven was proposed in [43, 112], which uses the systematic approach to generate two reference direction sets: one set on the boundary layer and the other on the inside layer. Despite that the two-layer approach improves the generation of reference directions, it still produces a large number of reference directions for MaOPs, which will be illustrated later.

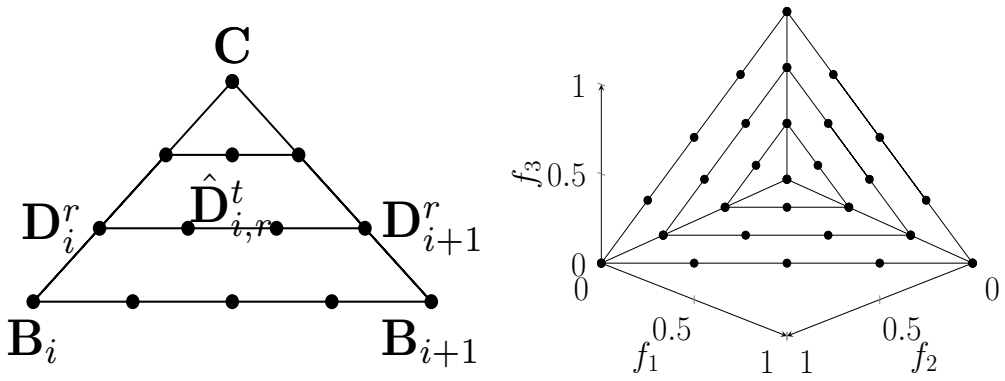
To reduce these drawbacks, we present a k-layer reference direction generation approach. Since any reference direction should be sampled from a unit simplex, we can partition the unit simplex into a number of subsimplexes and then generate a set of diverse reference directions on each subsimplex. First, we denote the central reference direction as $\mathbf{C} = (1/M, \cdots, 1/M)$, and the ith extreme reference direction (the intercept on the ith axis) as $\mathbf{B}_i = (b_1, \cdots, b_M)$ where $b_i = 1$ and $b_j = 0$ for all $j \neq i$, $1 \leq j \leq M$, $1 \leq i \leq M$. Thus, the unit simplex can be partitioned into M subsimplexes, each of which (denoted as $Simp(i)$) is bounded by points $\mathbf{C}$, $\mathbf{B}_i$ and $\mathbf{B}_{i+1}$. In the following, we explain how to use our proposed k-layer approach to generate reference directions for the subsimplex $Simp(i)$, and reference directions on other subsimplexes can be constructed in the same way.

For the subsimplex $Simp(i)$, we first generate points on sides $\mathbf{CB}_i$ and $\mathbf{CB}_{i+1}$. As illustrated in Fig. 5.3(a), the rth reference direction (denoted as $\mathbf{D}_i^r$) within the line $\mathbf{CB}_i$ can be calculated as follows:

$$\mathbf{D}_i^r = \mathbf{C} + \frac{r}{k}(\mathbf{B}_i - \mathbf{C}), \tag{5.1}$$

where $r \in \{1, \cdots, k\}$. This generates k reference directions (actually k layers from vertex $\mathbf{C}$ to the base $\mathbf{B}_i\mathbf{B}_{i+1}$) from the central point to the ith extreme point. After that, we focus on calculating reference directions within the rth layer. Likewise, the tth reference direction within the line $\mathbf{D}_i^r\mathbf{D}_{i+1}^r$ on the rth layer is computed by

$$\hat{\mathbf{D}}_{i,r}^t = \mathbf{D}_i^t + \frac{t}{r+1}(\mathbf{D}_{i+1}^r - \mathbf{D}_i^t), \tag{5.2}$$



(a) (b)

Fig. 5.3 Intersections of reference directions and a unit simplex: (a) reference directions on the subsimplex $Simp(i)$; (b) reference directions (with 28 directions generated by 3 layers) in three-dimensional space.

where $t \in \{1, \cdots, r\}$. This generates r reference directions for the rth layer of $Simp(i)$. Similarly, diverse reference directions on the rest of $M-1$ subsimplexes can be produced by the above method. At last, the constructed reference direction set W comprises the central reference direction $\mathbf{C}$, and reference directions of each layer in each subsimplex.

It is easy to see that, for k layers, the total number (H_M^k) of reference directions for an M-objective problem is given by

$$H_M^k = \sum_{i=1}^{M}\{\sum_{r=1}^{k} r+k\}+1 = \frac{Mk(k+3)}{2}+1. \tag{5.3}$$

For example, for $M = 3$ and $k = 1$, the reference directions are created on a triangle with vertices at (1,0,0), (0,1,0), and (0,0,1), including three midpoints of the sides of the triangle and an intermediate point at (1/3,1/3,1/3). Fig. 5.3(b) presents a simple example of reference direction set generated by three layers. In this work, for bi-objective problems, we use Das and Dennis's systematic approach to predefine a set of uniform reference directions, while for $M > 2$, the k-layer approach is used. The generation of a predefined reference direction set is described in Algorithm 5.2.

5.3.2 Offspring Reproduction and Objective Normalization

Reproduction (line 4 in Algorithm 5.1) is a step to create a new offspring population to update the parent population $\overline{P}$ (which is actually regarded as the archive). Here, mating selection plays a important role in reproduction. Each parent individual $P_1 \in \overline{P}$ needs a mate $P_2 \in \overline{P}$ to do reproduction. SPEA/R employs a restricted mating scheme to select the mate P_2 for P_1. Specifically, K candidates different from P_1 are randomly chosen from the parent population. Then, the candidate minimizing the Euclidean distance (in objective space) to P_1 can be screened as P_2. $K = 20$ is recommended in this work based on some

Algorithm 5.2: `Reference_Generation()`

Input: K (number of layers), M (number of objectives), $\overline{N}$ (archive size)
Output: W (reference direction set)

```
if M < 3 then
    Use Das and Dennis's method [39] to generate W;
end
else
    Generate extreme points B_i for i = 1, ..., M, and the central point C;
    for i := 1 to M do
        for r := 1 to K do
            Calculate all points on the r-th layer by Eq. (5.2);
        end
    end
end
```

Algorithm 5.3: `Objective_Normalization(`Q`)`

Input: Q (combined population)
Output: $\overline{Q}$ (normalized population)

```
for i := 1 to M do
    Compute the ideal point z^i_min := min_{q∈Q} f_i(q);
    Compute the worst point z^i_max := max_{q∈Q} f_i(q);
end
for each member q ∈ Q do
    Computed the normalized objective vector by Eq. (5.4);
    Save the normalized q to Q̄;
end
```

preliminary experiments. The restricted mating scheme may help alleviate recombination issues in many-objective optimization, where recombining two distant or very different parents is too disruptive and not likely to generate good children [43, 106].

After the production of the new offspring population P, SPEA/R then combines it and the parent population to form a population Q (line 5 in Algorithm 5.1), which is used later to normalize the objectives of individuals (line 6 in Algorithm 5.1). The normalization procedure is described in Algorithm 5.3. First, the ideal point $z_{min} = (z^1_{min}, \cdots, z^M_{min})$ and the worst point $z_{max} = (z^1_{max}, \cdots, z^M_{max})$ are constructed from the nondominated set of the combined $\overline{P}$ and P, where $z^i_{min} = min(f_i(q))$ and $z^i_{max} = max(f_i(q))$, $q \in Q$, $i = 1, \cdots, M$. Then, the objectives of member q are translated as follows:

$$\hat{f}_i(q) = \frac{f_i(q) - z^i_{min}}{z^i_{max} - z^i_{min}}, \tag{5.4}$$

where $i \in \{1, \cdots, M\}$ and $\hat{f}_i(q)$ denotes the ith normalized objective of member q.

Algorithm 5.4: `Associate`$(\overline{Q}, W, i)$

Input: $\overline{Q}$ (combined population), W (reference direction set)
Output: $E(i)$ (individuals in the ith subregion)

```
1 for each q ∈ Q̄ do
2 |   for each w ∈ W do
3 |   |   Compute the acute angle ⟨F̂(q), w⟩;
4 |   end
5 |   Assign ŵ = w : argmin_{w∈W} ⟨F̂(q), w⟩;
6 |   Assign θ_q = ⟨F̂(q), ŵ⟩;
7 |   Save q in E(ŵ);
8 end
```

5.3.3 Member Association and Fitness Assignment

After mapping the objectives of members of Q into a unit hypercube, next we need to associate each member in the normalized population $\overline{Q}$ with a reference direction (line 8 in Algorithm 5.1). The member association procedure is presented in Algorithm 5.4. For each reference direction $w^i \in W$, $i \in \{1, \cdots, H_M^k\}$, we define a subregion, denoted as Ψ^i, in the objective space, as follows:

$$\Psi^i = \{\hat{F}(x) \in \Omega_f | \langle \hat{F}(x), w^i \rangle \leq \langle \hat{F}(x), w^j \rangle\}, \tag{5.5}$$

where $j \in \{1, \cdots, H_M^k\}$, $x \in \Omega_x$, $\hat{F}(x)$ is the normalized objective vector of x, and $\langle \hat{F}(x), w^j \rangle$ is the acute angle between vectors $\hat{F}(x)$ and w^j. Using this definition can easily identify a number of members residing in Ψ^i, denoted as $E(i)$, from the normalized population $\overline{Q}$.

The idea of decomposing the objective space has also been employed in [21, 112, 127]. In both [21] and [127], the objective space decomposition provides a way to approximate a small segment of the PF, while in [112], it is used for local density estimation and diversity maintenance.

The decomposition of the objective space can facilitate fitness assignment, as shown in Algorithm 5.5. In detail, each member a in $E(i)$ is assigned a "local" [1] strength value $S(a)$, representing the number of solutions it dominates in $E(i)$:

$$S(a) = C(\{b \in E(i) | a \preceq b, a \in E(i)\}), \tag{5.6}$$

[1] The term "local" used here is to clarify the difference between our fitness assignment and that in SPEA2 [202].

Algorithm 5.5: `Fitness_Assignment`$(E(i))$

Input: $E(i)$ (individuals in the ith subregion), $\overline{Q}$ (combined population), W (reference direction set)
Output: FV (fitness values of members in $E(i)$)

```
for each a ∈ E(i) do
    Compute the "local" raw fitness R(a) using Eq. (5.7);
    Estimate the density value D(a) using Eq. (5.8);
    Compute the "local" fitness value FV_l(a) := R(a) + D(a);
    Assign the final fitness value FV(a) using Eq. (5.10);
end
```

where $C(\cdot)$ denotes the cardinality of a set. The "local" strength value is then used to calculate the "local" raw fitness $R(a)$ of a member a in $E(i)$, as follows:

$$R(a) = \sum_{b \in E(i), b \preceq a} S(b), \tag{5.7}$$

where the "local" raw fitness depends on the strengths of its dominators in the same subregion. Note that, similar to SPEA2, the fitness is to be minimized here.

In the case where individuals in $E(i)$ do not dominate each other, their raw fitness values will be zero and the above fitness assignment will make no sense. Fig. 5.4 presents such a situation, where both a and b are in the same subregion and they are nondominated individuals. Intuitively, a is better than b because a is closer to the associated reference direction (y-axis). Thus, individuals' other information should be considered. We adopt an angle-based density estimation technique to discriminate between individuals having identical raw fitness values. Each individual $a \in E(i)$ has a unique angle value $\theta_a = \langle \hat{F}(a), w^i \rangle$, which is actually the acute angle between $\hat{F}(a)$ and the associated reference direction w^i. Then, the density $D(a)$ of individual a is estimated by

$$D(a) = \frac{\theta_a}{\theta_a + \theta_m}, \tag{5.8}$$

where $\theta_m = \max_{1 \le i \le H_M^k} \min_{j \neq i} (w^i, w^j)$, i.e., the largest acute angle between two neighbouring reference directions, is added to ensure that $D(a)$ is smaller than one. The "local" fitness value of individual a, denoted as $FV_l(a)$, is composed of its raw fitness and density value, combined in the following form:

$$FV_l(a) = R(a) + D(a). \tag{5.9}$$

This way, individuals with better local diversity and convergence will have higher final fitness. Thus, a is better than b in the case illustrated in Fig. 5.4.

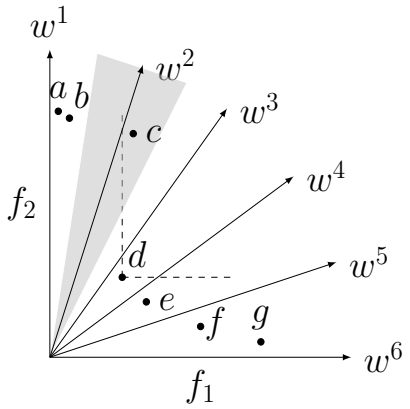


Fig. 5.4 Influence of decomposed subregions on environmental selection. The grey area represents the subregion occupied by w^2, i.e., Ψ^2, and the dashed lines are used to indicate d dominates c.

Despite great benefit for subregion diversity and local convergence, the "local" fitness assignment may impair global convergence if all individuals in Ψ^i are dominated by individuals in other subregions. To avoid this situation, individual a is also assigned a "global" fitness value, denoted as $FV_g(a)$, which is actually the raw fitness in SPEA2 (see [202]). Besides, if a is the only member in Ψ^i and dominated by individuals in other subregions, it should be given a chance to survive to the next generation. Thus, the final fitness of a, or $FV(a)$, is calculated as:

$$FV(a) = \begin{cases} FV_l(a) & \text{if } |\Psi^i| = 1; \\ FV_l(a) + FV_g(a) & \text{otherwise .} \end{cases} \tag{5.10}$$

Considering again the example in Fig. 5.4, individual c is the only member in the associated subregion Ψ^2, but, it is dominated by d in another subregion. This means Ψ^2 might be an underexploited area in the objective space and the search in this area should be enhanced. Conventional Pareto-dominance based techniques, e.g., NASG-II and SPEA2, however, are likely to ignore or even simply abandon important individuals like c in this area. In contrast, the proposed fitness assignment rewards the isolated c at an attempt to attract other individuals toward the underexploited area. This way, the fitness assignment hopefully provides a good approximation to each region of the PF.

5.3.4 Environmental Selection

In the environmental selection (line 11 in Algorithm 5.1), the best N individuals that can balance diversity and convergence should be preserved. Here, we present a new environmental selection strategy, which is shown in Algorithm 5.6. The strategy conducts DFCS-based sorting where individuals' convergence degree is measured by their fitness to classify the combined population $\overline{Q}$ into different fronts $\{F_1, F_2, \dots\}$. Solution selec-

Algorithm 5.6: DFCS$_{\text{E}}$nvironment_Selection$(\overline{Q}, W)$

Input: N (population size), $\overline{Q}$ (combined population), W (reference direction set)
Output: P (new parent population).

1 Set $P = \emptyset$;
2 $\{F_1, F_2, \ldots\} =$ DFCS_Sort(Q); // DFCS sorting based on fitness;
3 $l \leftarrow 1$;
4 **while** $C(P \cup F_l) \leq N$ **do**
5 $\quad P \leftarrow P \cup F_l$;
6 $\quad l \leftarrow l + 1$;
7 **end**
8 Fill up P with the best $N - C(P)$ individuals in terms of fitness from F_l.

tion starts from F_1 and ends until the last front F_l to be considered has more individuals than the remaining slots. Then, F_l is sorted according individuals' fitness and then the best $N - C(P)$ individuals are selected to fill up the N-sized P.

It should be noted that SPEA/R encourages convergence on the last front to be considered (line 8 in Algorithm 5.6). This is desirable because previous selection on the higher fronts has stressed diversity and later selection on F_l should prefer convergence to strike the balance between these two goals.

5.3.5 Computational Complexity of SPEA/R

The objective normalization (line 6 in Algorithm 5.1) requires $O(MN)$ computations. In line 8 of Algorithm 5.1, associating a combined population of $2N$ individuals to H_M^k reference directions takes $O(MNH_M^k)$. Suppose that $L_i = C(E(i))$, the number of individuals in the subregion Ψ^i, then $\sum_{i=1}^{H_M^k} L_i = N$. Thus, fitness assignment for $E(i)$ (line 9 in Algorithm 5.1) requires $O(ML_i^2)$ operations. For environmental selection, computational resources are mainly consumed by DFCS sorting. In Algorithm 5.6, line 2 requires $O(N \log N)$ comparisons to perform DFCS sorting. In this work, the population size N depends on H_M^k, as $N \approx H_M^k$. On average, the number of individuals in the ith subregion will be $L_i = 2N/H_M^k \approx 2$. Thus, the average complexity of one generational cycle of SPEA/R is $O(MN^2)$. In the worst case, that is, all the $2N$ individuals get trapped into one subregion and other subregions do not contain any member, the computational complexity reaches $O(MN^2)$, which is the same as the average complexity.

5.4 Experimental Studies

5.4.1 Experiments on Multiobjective Optimization

As a starting point, SPEA/R is studied on multiobjective problems. The test problems used here are the MOP [127] test suite, which is a modification of ZDT [200] and DTLZ [48] but more difficult than these predecessors. Since SPEA/R uses a framework similar to MOEA/D-M2M [127], it is interesting to make a comparison between them. Additionally, we also compared SPEA/R with a subproblem-constrained MOEA/D, i.e., MOEA/D-ACD [175], which is a recently-developed algorithm and has shown great promise for the MOP test problems. These three algorithms[2] employ the recombination operator [123], as suggested in MOEA/D-M2M [127]. For fairness, MOEA/D-ACD uses our reference direction initialization method. The population size was set to 100 (by the systematic approach [39]) and 313 (by our k-layer approach with $k = 13$) for bi- and three-objective problems, respectively. The maximum number of generation was set to 5000 for all the problems, and each algorithm was executed 30 independent runs for each problem. A detailed description of the MOP [127] test suite is provided in Appendix A.

5.4.1.1 Experimental Results and Analysis

Table 5.1 presents the results of SPEA/R, MOEA/D-M2M, and MOEA/D-ACD, where the mean and standard deviation values of IGD and HV are reported and the best value for each problem is marked in boldface. The differences between the approximations are assessed by the Wilcoxon rank-sum test [179] at the 0.05 significance level, with the standard Bonferroni correction [1] to deal with the problem of the higher probability of Type I errors in multiple comparisons.

The MOP [127] test suite contains seven hard-to-converge problems. In this suite, MOP4 is the only disconnected problem, and MOP6 and MOP7 are two three-objective problems. Besides, MOP4 to MOP7 are also diversity-resistant, which may be a big challenge to approximating well-distributed PFs if population diversity is not well maintained. Table 5.1 shows that SPEA/R performs significantly better than MOEA/D-M2M on most of the test problems, in terms of IGD and HV. SPEA/R competes well with MOEA/D-ACD in terms of HV on these problems. Generally, SPEA/R mainly loses on the three-objective MOP6. On another three-objective MOP7, however, SPEA/R wins the comparison by a clear margin.

To have a better understanding of these algorithms' performance, PF approximations over 30 runs for the seven MOP problems are displayed in Fig. 5.5. As can be seen from the figure, SPEA/R, MOEA/D-M2M, and MOEA/D-ACD are all able to approximate

[2]The source codes of MOEA/D-M2M and MOEA/D-ACD are from http://www.cs.cityu.edu.hk/∼qzhang/publications.html.

Table 5.1 Mean and stand deviation IGD and HV values on MOP problems

Metric	Prob.	SPEA/R	MOEA/D-M2M	MOEA/D-ACD
IGD	MOP1	**8.7805E-3(1.9373E-4)**	9.4133E-3(8.4998E-4)‡	9.0088E-3(1.6739E-4)†
	MOP2	**4.2374E-3(3.8551E-5)**	8.2719E-3(1.6819E-2)‡	4.4633E-3(5.8251E-5)†
	MOP3	**4.8235E-3(1.4936E-4)**	1.0236E-2(1.9945E-2)‡	4.9031E-3(1.5359E-4)†
	MOP4	**5.8664E-3(1.5107E-3)**	6.5855E-3(1.6268E-3)‡	7.7672E-3(1.5285E-3)‡
	MOP5	1.2053E-2(7.0952E-4)	9.3834E-3(5.1483E-4)	**8.6467E-3(2.2862E-4)**
	MOP6	4.0020E-2(2.6624E-3)	3.8164E-2(1.6047E-3)†	**2.5999E-2(3.5385E-4)**
	MOP7	**5.7604E-2(2.3640E-3)**	8.7838E-2(2.9091E-2)‡	1.0901E-1(3.9980E-3)‡
HV	MOP1	**3.6522E+0(2.8854E-4)**	3.6514E+0(1.0269E-3)‡	3.6520E+0(2.4682E-4)†
	MOP2	**3.3264E+0(1.0877E-4)**	3.3226E+0(1.8367E-2)‡	3.3207E+0(3.8048E-4)‡
	MOP3	**3.2101E+0(1.1819E-4)**	3.1825E+0(1.1641E-1)‡	3.2084E+0(1.1626E-3)†
	MOP4	**3.5128E+0(2.3030E-3)**	3.5109E+0(2.9883E-3)†	3.5071E+0(3.5365E-3)‡
	MOP5	3.6457E+0(1.1503E-3)	**3.6502E+0(1.4470E-3)**	3.6414E+0(5.7291E-3)†
	MOP6	7.7687E+0(3.9912E-3)	7.7356E+0(1.4549E-2)‡	**7.7956E+0(1.6302E-3)**
	MOP7	**7.3919E+0(3.2270E-3)**	7.3659E+0(3.0804E-2)‡	7.3730E+0(1.6802E-2)‡

‡ and † indicate SPEA/R performs significantly better than and equivalently to the corresponding algorithm, respectively.

the PF for the seven problems, but they perform differently in terms of convergence and diversity. Specifically, SPEA/R converges better than the other two algorithms on the first four bi-objective problems. On the two three-objective problems, i.e., MOP6 and MOP7, MOEA/D-M2M cannot achieve uniformly-distributed approximations and misses some boundary regions of the PF. This means that MOEA/D-M2M may not be able to cover the whole PF in higher-dimensional objective space. MOEA/D-ACD performs poorly in terms of diversity for MOP7, implying that adding constraints to subproblems is not sufficient to deal with hard-to-converge and diversity-resistant problems like MOP7. In contrast, SPEA/R maintains diversity well on both MOP6 and MOP7, although it does not fully converge to the PF in some runs.

Fig. 5.6 shows the three algorithms' IGD curves against the number of generations, where MOP1 is excluded as it has the similar IGD curve to MOP3. Clearly, SPEA/R is more capable of reducing the IGD value than the others and wins in most cases at late stage. This means that proper diversity maintenance helps promote the performance of SPEA/R.

The experiment on the MOP test suite shows that, SPEA/R and MOEA/D-M2M perform distinctly although both share some similar properties, e.g., decomposition of the objective space. The high performance of SPEA/R may be attributed to its good balance between diversity and convergence, which is achieved by our new proposed fitness assignment and environmental selection.

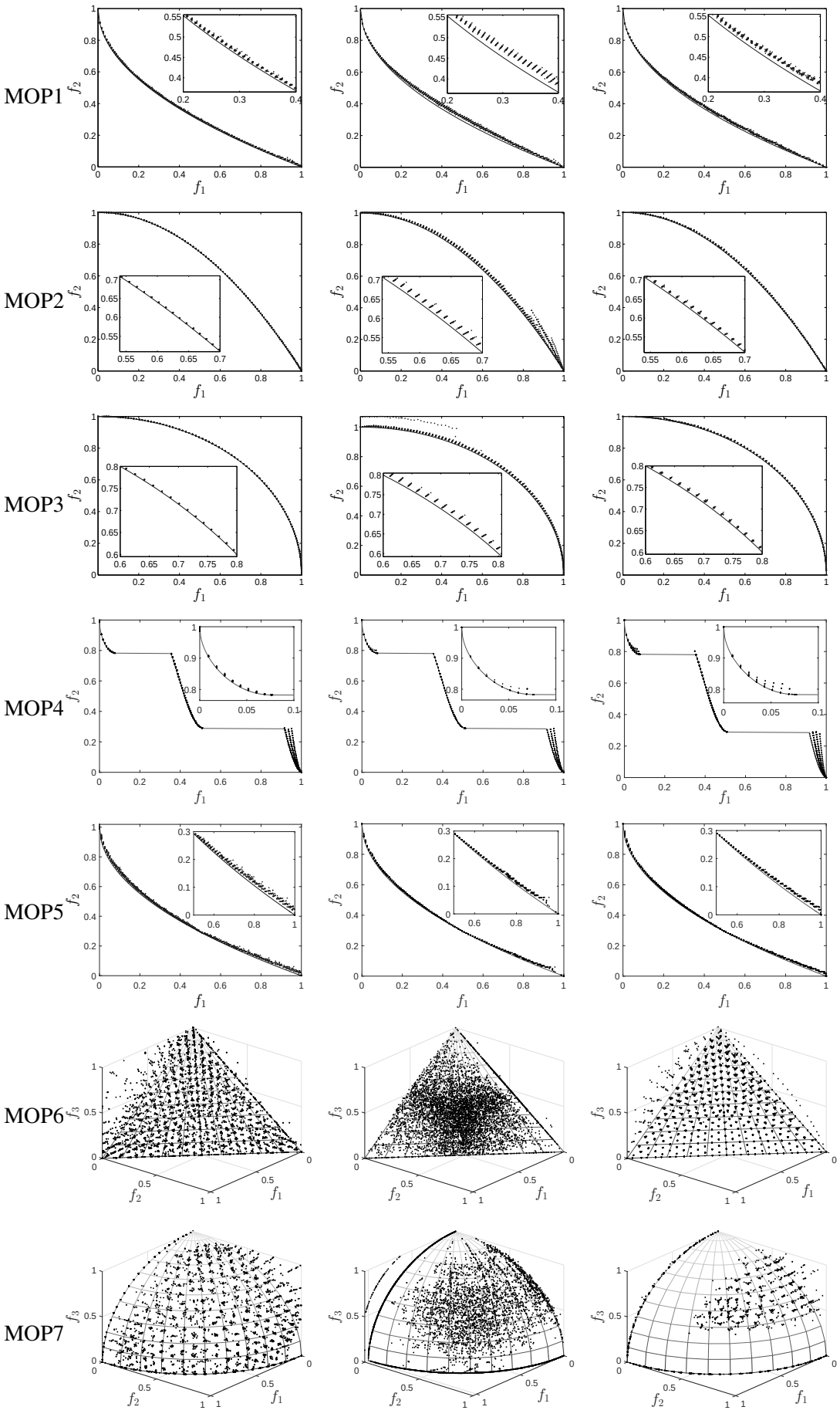
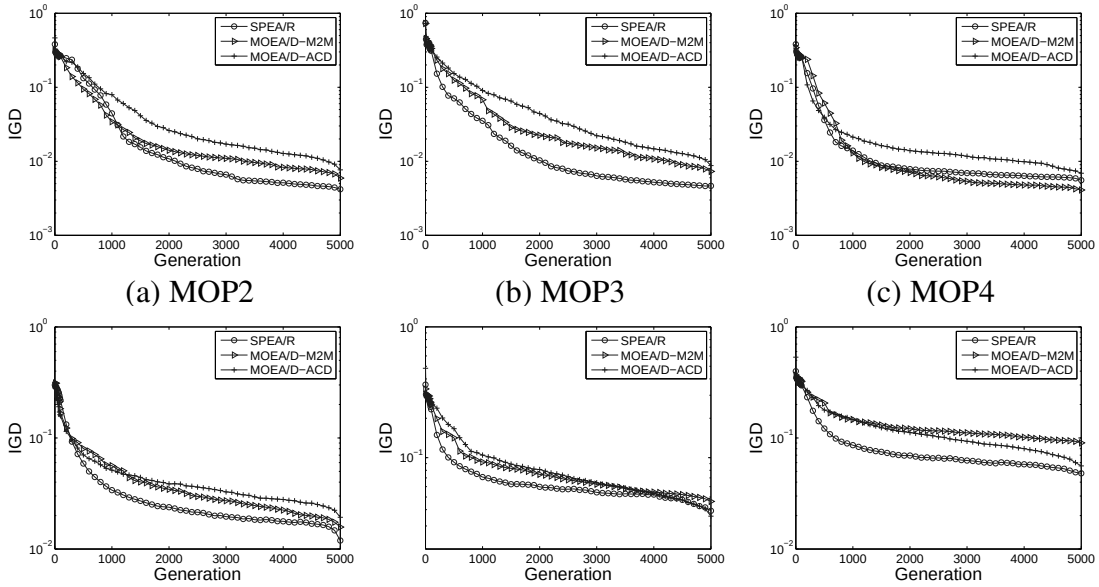


Fig. 5.5 PF approximations for MOP test problems over 30 runs.



(a) MOP2 (b) MOP3 (c) MOP4

(d) MOP5 (e) MOP6 (f) MOP7

Fig. 5.6 IGD curves of three algorithms for six MOP problems.

5.4.1.2 Comparison of Evolution Behaviour with MOEA/D-M2M

Experimental results in the previous subsection have validated the performance of SPEA/R, but it is still not clear why SPEA/R performs better than MOEA/D-M2M on the MOP test suite despite their similar framework. To answer this question, we further compare the evolution behaviour of these two algorithms on MOP2 and MOP3. To be more specific, the PF approximations obtained from three stages, i.e., the 50th (early stage), 500th (middle stage), and 1000th generation (late stage), are recorded, which are plotted in Fig. 5.7. It is clear to see from the figure that SPEA/R maintains good population diversity all the time, whereas MOEA/D-M2M tends to partition population into several subpopulations far away from each other before the late stage, which means diversity between neighbouring subpopulations is poorly controlled. As a consequence, MOEA/D-M2M takes more effort than SPEA/R to search unexplored regions before converging toward the PF and providing a good distribution of population, as illustrated by the 1000th-generation approximation for MOP2. This reason can be also used to explain the poor distribution of MOEA/D-M2M on the three-objective MOP6 and MOP7 in the previous experiment. The figure also indicates that the use of DFCS-based selection strategy can help SPEA/R to manage diversity and convergence well during the search.

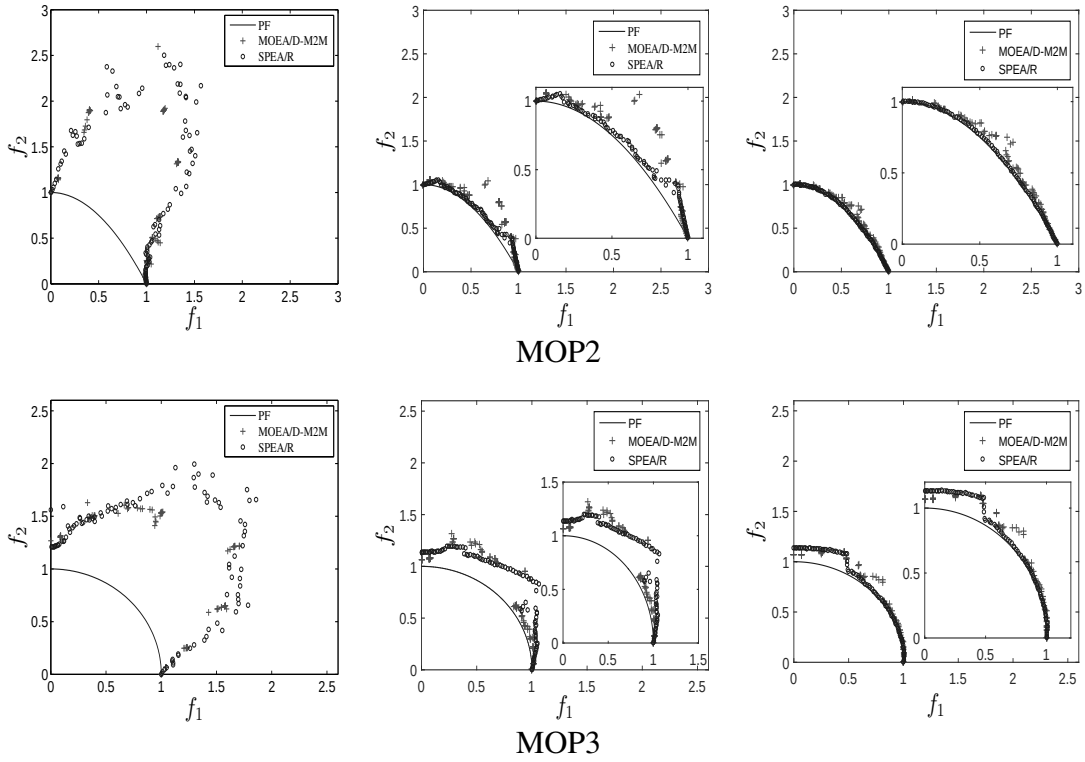


Fig. 5.7 Evolution behaviour comparison between SPEA/R and MOEA/D-M2M for three stages on MOP2 and MOP3. Left: 50th generation; middle: 500th generation; right: 1000th generation.

5.4.2 Experiments on Many-objective Optimization

Having had a good start on multi-objective optimization, SPEA/R is now examined on many-objective optimization. The following contributes to making a comparison of SPEA/R with state-of-the-art algorithms on many-objective problems.

5.4.2.1 Test Problems

The test problems used for algorithm comparison come from the WFG toolkit [82]. These problems contain a number of challenging characteristics, i.e., nonseparability, deception, multimodality, biased attributes, and various PF geometries. For each WFG test problem, the number of objectives varies from two to twelve, which considers both multiobjective and many-objective optimization. As recommended by the developers [82], the number of decision variables of all test instances is $n = k + l$, where k and l are the number of position-related variables and distance-related variables, respectively. $k = 2 \times (M - 1)$ and $l = 10$ are used in this work.

5.4.2.2 Compared Algorithms

Five popular or newly-developed MOEAs are used for comparison in our experimental studies. They are MOEA/D [188], HypE [6], SPEA2+SDE [117], PICEA-g [172], and NSGA-III [43], which represent different classes of metaheuristics. A brief description of each compared algorithm is given below.

1. MOEA/D[3] [188]: it is a representative of decomposition-based algorithms. In this work, PBI is adopted as the aggregation function for MOEA/D because it is empirically proved to be more effective than other decomposition methods for many-objective optimization in a recent study [43], and normalization [188] is used for scaled problems.

2. HypE[4] [6]: it is a representative of indicator-based MOEAs, which employs the hypervolume metric as an indicator in the environmental selection. In HypE, the fitness value of a solution is determined by not only its own hypervolume contribution but also the hypervolume contribution shared with others. Additionally, for the sake of computational complexity, HypE uses Monte Carlo simulation to approximate the exact hypervolume values.

3. SPEA2+SDE[5] [117]: this method introduces a density estimator that considers both the distribution and convergence information of individuals to increase the selection pressure in many-objective optimization. SPEA2+SDE has shown to be very promising for MaOPs [117].

4. PICEA-g[6] [172]: it introduces a new concept of preference-based coevolutionary algorithm (PICEA), which coevolves a family of decision-maker preferences together with a population of candidate solutions, for many-objective optimization. PICEA-g is an implementation of such a concept, where preferences gain higher fitness if it is satisfied by fewer solutions, and solutions gain fitness by meeting as many preferences as possible.

5. NSGA-III[7] [43]: it is an upgraded version of the most popular dominance-based NSGA-II algorithm, where a number of supplied reference points are used as a guideline for handling MaOPs. The basic framework of NSGA-III remains similar to NSGA-II, except that it maintains population diversity by niche preservation.

[3]The code of MOEA/D is from http://dces.essex.ac.uk/staff/qzhang/.

[4]The code of HypE is from http://www.tik.ee.ethz.ch/pisa/.

[5]The source code of SPEA2+SDE can be downloaded from http://www.tech.dmu.ac.uk/ syang/publications.html.

[6]The source code of the PICEA-g algorithm can be downloaded from http://www.sheffield.ac.uk/acse/staff/rstu/ruiwang/index.

[7]The source code of NSGA-III (version 1.1) can be downloaded from http://web.ntnu.edu.tw/∼tcchiang/publications/nsga3cpp/nsga3cpp.htm.

Table 5.2 Population size for different algorithms using the k-layer approach

M	k	H_M^k	MOEA/D	NSGA-III	SPEA/R
2	-	100	100	100	100
3	7	105	105	108	108
5	7	175	175	176	176
8	6	217	217	220	220
12	5	241	241	244	244

5.4.2.3 Parameter Settings

The parameters of the six MOEAs considered in the experiments are referenced from their original papers. Some key parameters in these algorithms were set as follows:

1. Reproduction parameters: All the algorithms used the simulated binary crossover (SBX) and polynomial mutation [43] as their genetic operators. The crossover probability was $p_c = 1.0$ and its distribution index was $\eta_c = 20$. The mutation probability was $p_m = 1/n$ and its distribution $\eta_m = 20$.

2. Population size: The population sizes (N) of different algorithms are presented in Table 5.2. The population sizes of all the algorithms except MOEA/D were set as $4\lceil H_M^k/4\rceil$, which is the smallest multiple of four not smaller than H_M^k, according to the suggestion in [43]. In other words, HypE, PICEA-g, and SPEA2+SDE use the same population size settings as NSGA-III and SPEA/R.

3. Stopping criterion and the number of executions: Each algorithm was terminated after a pre-specified number of generations. To be specific, for WFG problems, each algorithm stops after 300, 600, 1000, 1500, and 2000 generations for 2-, 3-, 5-, 8-, 12-objective cases, respectively. Additionally, each algorithm was executed 30 independent times on each test instance.

5.4.2.4 Experimental Results and Analysis

The performance measures for quantifying the performance of the compared algorithms in this section are IGD [43] and HV [203]. Note that, the PF points used for computing IGD here are a set of target points on the PF associated with reference directions, as suggested in [43]. For HV computation, the ith objective of the reference point used is $2i+2$ for all the WFG problems, and the HV values presented in this work are all normalized to $[0,1]$ by dividing $\prod_{i=1}^{M}(2i+2)$. The IGD and HV values of six algorithms on nine WFG test problems are presented in Tables 5.3 and 5.4, respectively.

The WFG1 problem mainly examines whether an MOEA can handle bias and mixed PF shapes. Both IGD and HV metrics indicate that PICEA-g is more suitable for this kind of problem than the other compared algorithms. SPEA/R competes well and even

Table 5.3 Mean and standard deviation IGD values obtained by six algorithms for WFG problems

Prob.	M	HypE	PICEA-g	MOEA/D	NSGA-III	SPEA2+SDE	SPEA/R
WFG1	2	8.8317E-1(3.0615E-2)†	8.4573E-1(7.7703E-2)†	1.0781E+0(4.1187E-2)‡	1.0260E+0(5.5762E-2)‡	**8.2124E-1(3.9450E-2)**	9.0085E-1(2.0778E-2)
	3	1.2748E+0(4.3651E-2)‡	**7.8534E-1(8.9205E-2)**	1.2041E+0(3.1711E-2)†	1.3784E+0(4.0202E-2)‡	1.2648E+0(2.3129E-2)‡	1.1794E+0(2.4394E-2)
	5	1.8676E+0(6.2653E-2)‡	**5.3311E-1(1.3060E-1)**	1.4877E+0(3.0123E-2)†	2.0084E+0(1.0479E-1)‡	1.8980E+0(1.6956E-2)‡	1.4780E+0(3.5911E-2)
	8	2.5670E+0(6.2118E-2)†	**1.4327E+0(1.4821E-1)**†	2.7002E+0(3.2555E-1)‡	2.7334E+0(9.6093E-2)‡	2.7327E+0(6.3839E-2)‡	1.5879E+0(6.9790E-2)
	12	3.6640E+0(4.1488E-1)	**3.2660E+0(6.1898E-1)**	4.9319E+0(7.5027E-1)†	4.8169E+0(2.8691E-1)†	3.6292E+0(8.4763E-2)	4.7653E+0(7.7595E-1)
WFG2	2	1.3134E-1(5.5902E-2)‡	8.6982E-2(6.4363E-2)‡	6.5756E-1(2.4753E-1)‡	9.8399E-2(7.0368E-2)‡	8.6488E-2(6.8054E-2)‡	**5.3339E-2(5.9669E-2)**
	3	3.9965E-1(9.4628E-2)‡	3.0184E-1(1.3370E-1)‡	1.6004E+0(5.1826E-1)‡	2.7108E-1(1.9464E-1)‡	2.9113E-1(1.3529E-1)‡	**1.9273E-1(1.4337E-1)**
	5	1.0282E+0(3.1081E-1)‡	6.9710E-1(3.7222E-1)‡	3.4876E+0(9.7999E-1)‡	6.1589E-1(5.8567E-1)‡	5.2034E-1(1.8118E-1)‡	**4.1324E-1(1.9367E-1)**
	8	1.6356E+0(2.0123E-1)‡	1.0825E+0(3.2538E-1)‡	6.9688E+0(3.6616E+0)‡	2.5990E+0(5.5451E-1)‡	1.0398E+0(8.0346E-2)†	**9.2918E-1(1.5704E-2)**
	12	3.6569E+0(1.0009E+0)‡	1.9909E+0(1.6166E-1)†	1.4420E+1(7.0122E+0)‡	5.3146E+0(2.1231E+0)‡	2.0416E+0(2.9444E-1)†	**1.9406E+0(4.8101E-1)**
WFG3	2	6.1055E-2(2.1969E-2)‡	1.5970E-2(1.0060E-3)‡	2.1120E-2(1.0954E-2)‡	1.7672E-2(3.9260E-3)‡	1.4922E-2(1.3055E-3)‡	**9.5981E-3(1.4972E-3)**
	3	3.6647E-1(7.0170E-2)‡	1.2092E-1(8.5003E-3)†	**8.4610E-2(2.0841E-2)**	1.1033E-1(1.9565E-2)‡	1.0415E-1(9.6999E-3)	1.2706E-1(1.9111E-2)
	5	8.7070E-1(3.4817E-1)‡	4.2029E-1(2.6257E-2)†	**2.1410E-1(9.2308E-3)**	3.9046E-1(5.6144E-2)	6.8446E-1(7.4234E-2)‡	4.7455E-1(5.0952E-2)
	8	1.2895E+0(3.8251E-1)	**1.0091E+0(1.1303E-1)**	8.6567E+0(2.5278E-2)‡	2.4852E+0(1.3923E+0)‡	2.3216E+0(2.1709E-1)‡	1.8096E+0(7.9589E-1)
	12	2.1678E+0(6.4471E-1)†	**1.5164E+0(4.9334E-1)**	1.3202E+1(5.0194E-2)‡	6.1449E+0(1.0145E+0)‡	4.3313E+0(2.2150E-1)‡	2.2071E+0(8.9759E-1)
WFG4	2	3.1877E-2(6.5145E-3)‡	1.6304E-2(1.2351E-3)‡	2.7940E-2(6.4054E-3)‡	1.2596E-2(3.0353E-3)‡	3.0674E-2(6.3801E-3)‡	**4.0642E-3(6.3104E-4)**
	3	5.6412E-1(8.5813E-2)‡	2.0043E-1(7.9213E-3)‡	6.9098E-2(1.0745E-2)‡	6.3007E-2(5.4682E-3)‡	2.9956E-1(1.5741E-2)‡	**2.8864E-2(2.1139E-3)**
	5	2.0444E+0(2.2951E-1)‡	1.1045E+0(4.7317E-1)‡	1.4182E-1(1.4965E-2)‡	3.6864E-1(4.4117E-1)‡	1.2366E+0(7.4328E-2)‡	**1.0932E-1(7.3118E-3)**
	8	6.0523E+0(1.5295E+0)‡	6.6026E+0(7.9339E-1)‡	1.4683E+1(1.1304E+0)‡	2.3447E+0(8.3237E-1)‡	3.7387E+0(2.1384E-1)‡	**3.0963E-1(4.4228E-2)**
	12	1.1022E+1(1.4125E+0)‡	1.4136E+1(9.6910E-1)‡	2.4085E+1(3.8119E-7)‡	9.3126E+0(9.3937E-1)‡	7.8294E+0(2.7645E-1)‡	**5.545E-1(5.8877E-2)**
WFG5	2	1.4442E-1(2.9026E-2)‡	6.9351E-2(1.8443E-3)†	7.2942E-2(1.6355E-3)†	6.9201E-2(1.9862E-3)‡	8.1770E-2(4.0625E-3)‡	**6.8656E-2(6.0334E-4)**
	3	7.9362E-1(1.4324E-1)‡	2.1530E-1(6.1536E-3)‡	1.0567E-1(4.0651E-3)†	2.1164E-1(1.2089E-2)‡	2.8447E-1(1.1878E-2)‡	**1.0007E-1(2.7774E-3)**
	5	2.3647E+0(5.0181E-1)‡	9.3128E-1(1.7072E-2)‡	1.7300E-1(2.0329E-2)‡	3.6011E-1(2.7441E-2)‡	1.1152E+0(5.7730E-2)‡	**1.5221E-1(3.3155E-3)**
	8	5.0708E+0(9.3870E-1)‡	3.6695E+0(6.6358E-1)‡	1.4665E+1(1.6616E-1)‡	1.0424E+0(1.2969E+0)‡	3.0894E+0(1.6408E-1)‡	**2.9294E-1(7.8991E-3)**
	12	1.1590E+1(3.7303E+0)‡	1.1189E+1(6.7757E-1)‡	2.3809E+1(3.6910E-2)‡	1.1152E+1(1.7458E+0)‡	6.8783E+0(3.5983E-1)‡	**5.9182E-1(5.6574E-2)**
WFG6	2	9.5505E-2(2.7439E-2)‡	8.7771E-2(1.6443E-2)†	1.2231E-1(2.9457E-2)‡	**6.2542E-2(8.0865E-3)**	8.3074E-2(2.2805E-2)†	8.2235E-2(1.7636E-2)
	3	5.0657E-1(5.9845E-2)‡	2.2653E-1(1.1940E-2)‡	1.6990E-1(4.3933E-2)‡	1.3889E-1(1.6059E-2)†	3.0489E-1(2.5190E-2)‡	**1.2494E-1(1.8689E-2)**
	5	1.7161E+0(1.7421E-1)‡	9.3252E-1(2.8199E-2)‡	2.4944E-1(5.9006E-2)‡	2.5876E-1(1.8224E-2)‡	1.1083E+0(4.5895E-2)‡	**1.9652E-1(2.1632E-2)**
	8	3.5742E+0(2.2169E-1)‡	2.4527E+0(1.0243E-1)‡	1.3441E+1(3.4359E+0)‡	3.5306E-1(3.9311E-2)‡	2.8712E+0(1.5114E-1)‡	**3.1442E-1(4.5861E-2)**
	12	8.5555E+0(1.2719E+0)‡	8.3868E+0(1.5120E+0)‡	2.4086E+1(1.0499E-3)‡	1.2026E+0(1.4127E+0)‡	6.5637E+0(3.4155E-1)‡	**5.6185E-1(6.4777E-2)**
WFG7	2	8.2165E-2(2.8926E-2)‡	1.5910E-2(6.6738E-4)‡	2.3036E-2(5.5223E-3)‡	6.1770E-3(1.2478E-3)‡	2.9225E-2(4.7514E-3)‡	**3.0428E-3(6.4407E-4)**
	3	6.6418E-1(9.6773E-2)‡	1.9958E-1(6.0806E-3)‡	1.0757E-1(6.5050E-2)‡	4.5169E-2(5.3488E-3)‡	2.6732E-1(1.8452E-2)‡	**1.7752E-2(2.2178E-3)**
	5	2.1347E+0(2.2676E-1)‡	9.3521E-1(2.3560E-2)‡	1.3512E-1(2.0759E-2)‡	1.9524E-1(3.9223E-2)‡	1.2314E+0(7.5941E-2)‡	**7.6146E-2(6.0145E-3)**
	8	5.7576E+0(1.2137E+0)‡	4.7858E+0(1.3194E+0)‡	4.1326E+0(4.8119E+0)‡	1.9059E+0(5.2245E-1)‡	3.3239E+0(2.3293E-1)‡	**4.4555E-2(1.4906E-3)**
	12	1.3634E+1(2.8675E+0)‡	1.1684E+1(1.1888E+0)‡	1.8854E+1(7.2573E+0)‡	9.0098E+0(1.1672E+0)‡	7.2319E+0(2.0432E-1)‡	**1.2239E+0(1.7913E-1)**
WFG8	2	1.1761E-1(1.3492E-2)‡	1.7830E-1(8.4903E-3)‡	1.9949E-1(7.7770E-2)‡	1.0440E-1(4.4972E-3)‡	9.6227E-2(7.0488E-3)‡	**6.5348E-2(8.3518E-3)**
	3	6.4260E-1(9.6987E-2)‡	3.6302E-1(6.6066E-3)‡	2.9721E-1(1.8361E-2)‡	2.6280E-1(1.1173E-2)‡	3.9859E-1(1.2363E-2)‡	**1.8791E-1(1.2203E-2)**
	5	3.0822E+0(3.6658E-1)‡	1.1359E+0(1.5299E-1)‡	6.2743E-1(2.4932E-2)‡	6.2983E-1(3.4558E-2)‡	1.3975E+0(7.1725E-2)‡	**4.4959E-1(6.3168E-2)**
	8	6.6256E+0(6.9786E-1)‡	4.8593E+0(6.4988E-1)‡	1.4786E+1(5.2920E-1)‡	3.9291E+0(8.5793E-1)‡	3.6385E+0(1.3014E-1)‡	**7.7559E-1(1.7947E-1)**
	12	1.2621E+1(1.3872E+0)‡	1.2125E+1(8.1783E-1)‡	2.4081E+1(1.6969E-2)‡	9.5723E+0(7.4981E-1)‡	7.5282E+0(2.7831E-1)‡	**25918E+0(1.5246E+0)**
WFG9	2	1.1042E-1(2.1126E-1)‡	2.5073E-2(2.0117E-3)‡	1.1382E-1(9.2790E-2)‡	4.1288E-2(3.6557E-2)‡	4.6615E-2(4.5168E-2)‡	**2.3333E-2(1.8028E-3)**
	3	8.5309E-1(3.9767E-1)‡	2.0280E-1(5.3020E-3)‡	3.6735E-1(8.0129E-2)‡	2.1455E-1(2.8924E-2)‡	2.8436E-1(2.9944E-2)‡	**1.3565E-1(6.0332E-2)**
	5	2.2225E+0(5.9275E-1)‡	9.0468E-1(2.3965E-2)‡	5.6825E-1(5.1271E-2)‡	4.7457E-1(2.0885E-2)†	1.2446E+0(1.0233E-1)‡	**4.4527E-1(1.0612E-1)**
	8	5.6313E+0(2.0539E+0)‡	2.3506E+0(1.8201E-1)‡	1.4075E+1(3.1523E+0)‡	1.4547E+0(7.5551E-1)‡	3.2874E+0(2.1696E-1)‡	**1.0596E+0(2.1459E-1)**
	12	1.1425E+1(4.1445E+0)‡	9.0759E+0(1.1149E+0)‡	2.3886E+1(1.0426E-1)‡	7.0250E+0(2.2799E+0)‡	6.8976E+0(3.8740E-1)‡	**1.9587E+0(4.1090E-1)**

‡ and † indicate SPEA/R performs significantly better than and equivalently to the corresponding algorithm, respectively.

outperforms the others for relatively low-dimensional cases (in objective space). But, it is defeated by NSGA-III on the 12-objective WFG1 in terms of the HV metric.

WFG2 challenges algorithms' ability to locate all disconnected PF segments and handle nonseparable variable dependencies. For this problem, all the algorithms can achieve impressive performance in low-dimensional cases (in objective space), and SPEA/R wins by a clear margin. However, when the number of objectives is over five, the performance of MOEA/D and NSGA-III degrades sharply whereas SPEA/R continues to yield good results. SPEA/R wins in the 8-objective case and can compete with HypE and SPEA2+SDE in the 12-objective case, as indicated by both IGD and HV metrics. This means SPEA/R can deal with disconnectivity.

WFG3 features a degenerated and linear PF shape, and its variables are nonseparable as well. For this problem, while SPEA/R performs best for the 2-objective case, its performance degrades sharply when the number of redundant objectives increases, which is also the case for the other algorithms except PICEA-g. PCIEA-g is roughly the best performer for this problem because it generates nondominated reference points in the ob-

Table 5.4 Mean and standard deviation HV values obtained by six algorithms for WFG problems

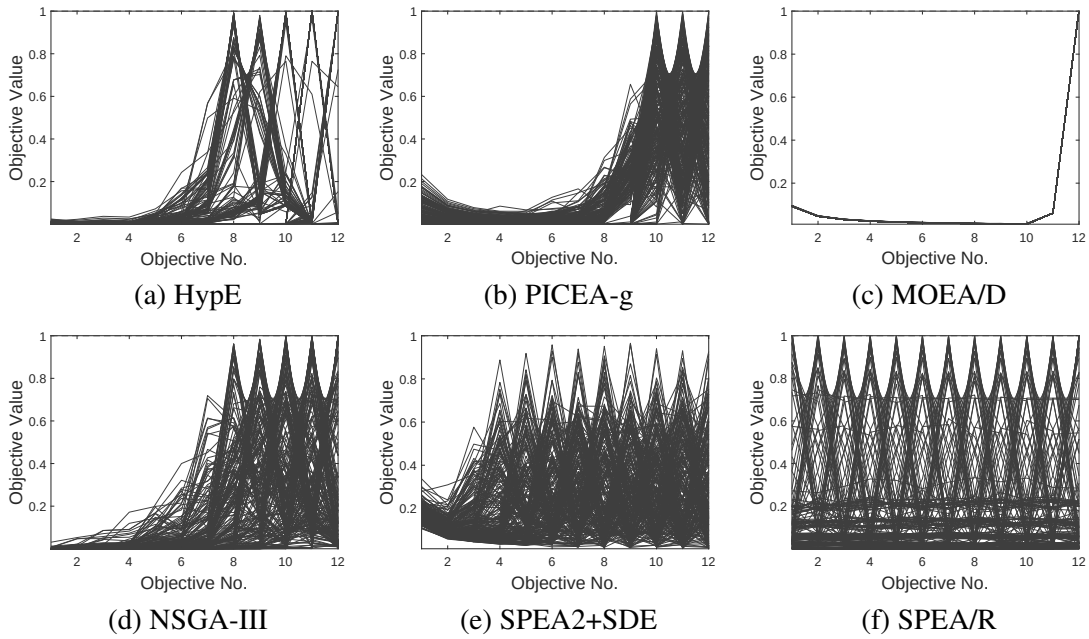
Prob.	M	HypE	PICEA-g	MOEA/D	NSGA-III	SPEA2+SDE	SPEA/R
WFG1	2	6.1349E-1(8.4520E-3)†	6.2383E-1(2.1342E-2)	5.2963E-1(2.6156E-2)‡	5.2796E-1(2.1870E-2)‡	**6.3186E-1(1.1067E-2)**	6.0736E-1(5.9395E-3)
	3	5.9929E-1(8.2784E-3)‡	**7.3526E-1(2.6053E-2)**	5.8131E-1(2.7399E-2)‡	5.4419E-1(2.4954E-2)‡	6.0147E-1(6.0920E-3)‡	6.2420E-1(6.0584E-3)
	5	5.2153E-1(7.6354E-3)‡	**9.1200E-1(5.4121E-2)**	5.7746E-1(7.5357E-3)‡	5.4737E-1(1.7873E-2)‡	5.0145E-1(3.6774E-3)‡	5.7919E-1(5.4070E-3)
	8	4.3738E-1(1.3688E-2)‡	**9.3707E-1(2.1198E-2)**	4.3174E-1(4.4287E-2)‡	4.8199E-1(1.7796E-2)‡	4.2294E-1(2.3498E-3)‡	6.2128E-1(7.8831E-2)
	12	3.7561E-1(4.6241E-3)‡	**9.2162E-1(2.9084E-2)**	3.2474E-1(7.3978E-2)‡	7.4386E-1(4.0646E-2)	3.6416E-1(3.7633E-3)‡	4.2884E-1(5.9670E-2)
WFG2	2	7.9746E-1(3.0072E-2)‡	8.1726E-1(3.3376E-2)‡	6.5750E-1(5.0778E-2)‡	8.0801E-1(3.4064E-2)‡	8.1659E-1(3.4444E-2)‡	**8.3092E-1(2.9447E-2)**
	3	9.2195E-1(5.6280E-2)‡	9.0282E-1(6.9396E-2)‡	6.4363E-1(7.0612E-2)‡	9.1538E-1(6.8576E-2)‡	9.0856E-1(6.6674E-2)‡	**9.4106E-1(6.0781E-2)**
	5	9.2808E-1(7.4718E-2)‡	9.4948E-1(7.9765E-2)‡	6.2190E-1(8.7394E-2)‡	9.6131E-1(6.1941E-2)‡	9.7887E-1(3.5391E-2)‡	**9.8674E-1(3.7587E-2)**
	8	9.7630E-1(3.8039E-3)‡	9.8194E-1(5.5797E-2)‡	5.5675E-1(2.5975E-1)‡	9.1862E-1(1.0630E-1)‡	9.8544E-1(2.1334E-3)‡	**9.9759E-1(6.6128E-4)**
	12	9.6801E-1(1.1392E-2)‡	**9.9860E-1(8.5921E-4)**	3.3908E-1(3.1366E-1)‡	7.9068E-1(1.6581E-1)‡	9.7314E-1(3.9702E-2)‡	9.8832E-1(9.2631E-2)
WFG3	2	8.1688E-1(4.4403E-3)‡	8.2899E-1(4.8416E-4)†	8.2486E-1(4.3417E-3)‡	8.2634E-1(1.1226E-3)‡	8.2739E-1(1.0559E-3)‡	**8.2945E-1(1.3184E-3)**
	3	7.5060E-1(9.0712E-3)‡	**8.9256E-1(7.5315E-4)**	7.9356E-1(5.8079E-3)	7.8197E-1(4.1079E-3)†	7.8383E-1(4.3288E-3)‡	7.7947E-1(8.0418E-3)
	5	6.6263E-1(2.6816E-2)‡	**9.5989E-1(1.2055E-3)**	7.3844E-1(5.3374E-3)	7.1730E-1(7.5465E-3)†	6.8788E-1(1.3525E-2)†	6.8696E-1(1.1909E-2)
	8	6.0613E-1(2.3359E-2)	**9.8966E-1(5.0622E-4)**	1.2210E-1(1.2553E-3)‡	4.6929E-1(1.0987E-1)‡	4.8493E-1(3.3216E-2)†	4.9757E-1(4.4357E-2)
	12	5.6477E-1(2.7196E-2)	**9.5252E-1(6.7478E-2)**	8.6302E-2(1.5356E-3)‡	3.3546E-1(3.7684E-2)‡	3.2742E-1(6.1689E-3)‡	4.8913E-1(4.4723E-2)
WFG4	2	7.2511E-1(6.1498E-3)‡	7.3428E-1(5.4716E-4)†	7.2972E-1(1.3665E-3)‡	7.3314E-1(9.9494E-4)‡	7.1096E-1(1.1254E-2)‡	**7.3497E-1(6.0901E-4)**
	3	8.2289E-1(2.1555E-2)‡	8.4966E-1(1.1654E-3)†	8.3709E-1(2.2772E-3)‡	8.3960E-1(1.2736E-3)‡	7.6916E-1(1.1329E-2)‡	**8.5842E-1(9.0221E-4)**
	5	7.4786E-1(3.9699E-2)‡	9.1105E-1(7.2895E-2)‡	9.1690E-1(1.2970E-3)‡	8.7615E-1(4.9747E-2)‡	7.5415E-1(1.2288E-2)‡	**9.2169E-1(1.2998E-3)**
	8	5.8468E-1(6.8400E-2)‡	7.0094E-1(7.4721E-2)‡	1.3762E-1(8.1625E-2)‡	8.0583E-1(4.5507E-2)‡	5.8432E-1(2.5894E-2)‡	**9.5797E-1(3.9059E-3)**
	12	5.3389E-1(5.8040E-2)‡	6.6260E-1(7.2208E-2)‡	7.6923E-2(5.9845E-9)‡	7.1332E-1(3.0720E-2)‡	5.3058E-1(3.7951E-2)‡	**9.1609E-1(7.8544E-2)**
WFG5	2	6.8762E-1(4.8597E-3)‡	7.0799E-1(2.7043E-3)‡	7.0280E-1(1.5455E-3)†	7.0558E-1(3.3458E-3)†	6.8987E-1(7.5992E-3)‡	**7.1227E-1(9.3626E-4)**
	3	7.6888E-1(2.2719E-2)‡	8.2305E-1(1.5867E-3)†	8.1368E-1(2.3212E-3)†	8.0667E-1(2.8385E-3)‡	7.7206E-1(7.9184E-3)‡	**8.2437E-1(2.0573E-3)**
	5	7.4369E-1(5.0270E-2)‡	**9.0566E-1(1.5576E-3)**	8.7778E-1(2.3524E-3)†	8.4922E-1(3.6613E-3)‡	7.8108E-1(1.0898E-2)‡	8.8792E-1(1.1292E-3)
	8	6.0049E-1(7.9653E-2)‡	8.4677E-1(4.8498E-2)‡	1.0334E-1(5.3172E-3)‡	8.5495E-1(6.6536E-2)‡	6.9019E-1(1.6596E-2)‡	**9.0023E-1(2.9488E-3)**
	12	4.3256E-1(6.7488E-2)‡	7.3490E-1(4.4608E-2)‡	6.8603E-2(8.1182E-4)‡	6.3626E-1(5.0428E-2)‡	5.9168E-1(2.0913E-2)‡	**9.0236E-1(6.6030E-3)**
WFG6	2	6.9937E-1(1.1071E-2)†	7.0821E-1(5.5304E-3)†	6.9412E-1(1.0969E-2)‡	**7.1557E-1(3.0357E-3)**	6.9758E-1(1.5288E-2)†	7.1001E-1(6.5120E-3)
	3	7.8656E-1(1.1842E-2)‡	8.1866E-1(6.8386E-3)‡	8.0201E-1(1.2884E-2)‡	8.1457E-1(4.2668E-3)‡	7.7216E-1(1.2142E-2)‡	**8.2552E-1(6.3065E-3)**
	5	8.0206E-1(2.2080E-2)‡	8.9790E-1(1.0352E-2)†	8.6441E-1(1.4974E-2)‡	8.6594E-1(5.3327E-3)‡	7.7777E-1(1.3556E-2)‡	**8.9930E-1(7.7577E-3)**
	8	7.0169E-1(5.5025E-2)‡	9.2290E-1(7.9111E-3)†	1.9017E-1(1.8471E-1)‡	9.0534E-1(9.3956E-3)‡	7.2350E-1(1.6221E-2)‡	**9.2855E-1(1.5045E-2)**
	12	5.1133E-1(4.6498E-2)‡	8.2963E-1(5.2905E-2)‡	6.8017E-2(2.6487E-3)‡	9.0514E-1(3.8475E-2)†	6.3134E-1(2.2436E-2)‡	**9.1604E-1(1.2921E-2)**
WFG7	2	7.1926E-1(4.5563E-3)‡	7.3505E-1(2.4614E-4)†	7.3136E-1(1.0965E-3)‡	7.3510E-1(3.1131E-4)†	7.2082E-1(8.9974E-3)‡	**7.3615E-1(1.1625E-3)**
	3	8.1856E-1(1.4867E-2)‡	8.5110E-1(8.3476E-4)†	8.3278E-1(1.5401E-2)‡	8.4263E-1(1.1488E-3)‡	7.9338E-1(1.3433E-2)‡	**8.5171E-1(3.4081E-4)**
	5	7.4819E-1(5.6439E-2)‡	**9.4257E-1(9.5013E-4)**	9.2071E-1(1.3121E-3)‡	8.9358E-1(4.7463E-3)‡	7.4066E-1(1.4528E-2)‡	9.2528E-1(6.1088E-4)
	8	5.8792E-1(7.2407E-2)‡	8.2636E-1(8.6834E-2)‡	7.4490E-1(2.7588E-1)‡	8.4280E-1(2.5436E-2)‡	6.9531E-1(1.9407E-2)‡	**9.5110E-1(1.7669E-2)**
	12	4.6000E-1(8.2486E-2)‡	7.6346E-1(6.1408E-2)‡	2.7676E-1(2.8129E-1)‡	7.5706E-1(3.6351E-2)‡	6.3525E-1(2.3553E-2)‡	**9.4407E-1(9.1515E-3)**
WFG8	2	6.7904E-1(1.0189E-2)‡	6.7029E-1(3.8045E-3)‡	6.6299E-1(2.3559E-2)‡	6.9584E-1(1.4755E-3)‡	6.8948E-1(4.3227E-3)‡	**7.0318E-1(3.3615E-3)**
	3	7.2386E-1(2.5175E-2)‡	7.7446E-1(2.1990E-3)‡	7.6918E-1(4.6432E-3)‡	7.9129E-1(2.5827E-3)‡	7.2355E-1(8.3819E-3)‡	**8.0651E-1(5.3416E-3)**
	5	6.2664E-1(4.0359E-2)‡	8.3085E-1(1.5580E-2)‡	7.9897E-1(1.7626E-3)‡	7.9816E-1(5.4098E-3)‡	7.0110E-1(1.1835E-2)‡	**8.5983E-1(1.8504E-2)**
	8	5.6633E-1(4.5007E-2)‡	8.0925E-1(4.2507E-2)‡	1.2037E-1(2.3436E-2)‡	7.3449E-1(1.9296E-2)‡	6.3416E-1(1.7580E-2)‡	**8.8805E-1(3.8120E-2)**
	12	5.4577E-1(6.2990E-2)‡	7.4233E-1(4.8267E-2)‡	7.6453E-2(2.1549E-3)‡	6.9534E-1(3.5933E-2)‡	5.5797E-1(1.9437E-2)‡	**8.7084E-1(2.2243E-1)**
WFG9	2	6.9556E-1(3.9680E-2)‡	7.1483E-1(8.6702E-4)†	6.8383E-1(2.8965E-2)‡	7.0797E-1(1.1834E-2)‡	7.0116E-1(1.4582E-2)‡	**7.1499E-1(1.4646E-3)**
	3	7.4158E-1(7.1225E-2)‡	8.1680E-1(2.8524E-3)†	7.2575E-1(2.5631E-2)‡	7.8864E-1(1.2055E-2)‡	7.6490E-1(1.6052E-2)‡	**8.2240E-1(2.2316E-2)**
	5	6.4243E-1(1.0678E-1)‡	8.5030E-1(5.3667E-2)†	7.5157E-1(2.2857E-2)‡	7.9189E-1(3.9893E-3)‡	7.1535E-1(2.8081E-2)‡	**8.6833E-1(4.2512E-2)**
	8	4.4778E-1(1.0543E-1)‡	**8.4756E-1(5.2963E-2)**	1.2548E-1(1.6679E-1)‡	7.5797E-1(2.7755E-2)‡	6.3961E-1(1.9066E-2)‡	8.1250E-1(4.0473E-2)
	12	3.8782E-1(1.3885E-1)‡	7.7356E-1(4.2001E-2)‡	5.9415E-2(6.0946E-3)‡	6.6920E-1(3.6199E-2)‡	5.9417E-1(2.9346E-2)‡	**7.8937E-1(4.7711E-2)**

‡ and † indicate SPEA/R performs significantly better than and equivalently to the corresponding algorithm, respectively.

jective space to guide the search in every generation. The other algorithms to a certain extent try to spread the population over the whole objective space for the sake of diversity, leading to a very limited number of points on the degenerated PF. Despite that, SPEA/R outperforms MOEA/D and NSGA-III and performs competitively with SPEA2+SDE for the 8- and 12-objective cases. This may be because fitness assignment in SPEA/R favours nondominated solutions.

The problems WFG4 to WFG9 have an identical hyperellipse surface, but they differ in some other characteristics. To be specific, WFG4 introduces multimodality to test algorithms' ability to escape from local optima, and WFG5 is a deceptive problem, and the difficulty lies in the large "aperture" size of the well/basin leading to the global minimum. WFG6 has a significant nonseparable reduction, and WFG7-9 all introduce some bias to challenge algorithms' diversity, but WFG8-9 are nonseparable. Also, variable linkages in WFG8 are much more difficult than that in WFG9.

For WFG4-WFG9, SPEA/R wins nearly all the tested cases in terms of IGD and HV, showing its high ability to deal with a number of considered characteristics in these



(a) HypE (b) PICEA-g (c) MOEA/D

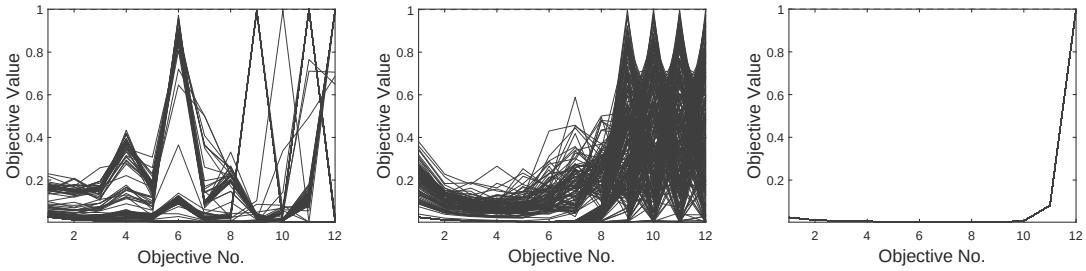
(d) NSGA-III (e) SPEA2+SDE (f) SPEA/R

Fig. 5.8 Parallel coordinates of final solutions obtained by six algorithms for the 12-objective WFG4 instance.

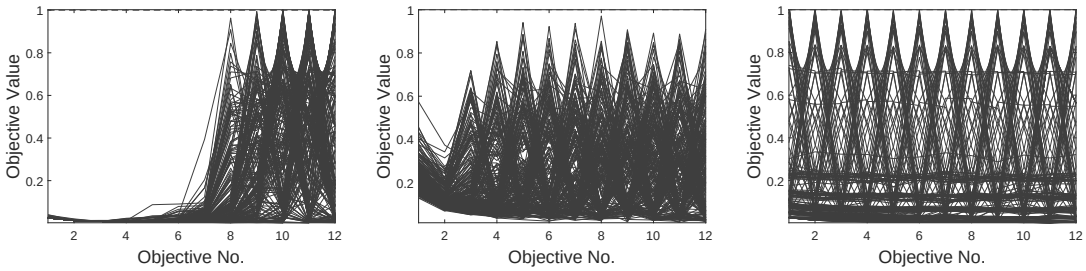
problems. Considering the HV metric, PICEA-g also achieves very competitive results on these problems, and outperforms or compares well with SPEA/R in some cases. However, none of the other algorithms can compete with SPEA/R.

The above experimental studies show that the tested algorithms' performance can be influenced by at least two factors, i.e., problem characteristics and the number of objectives. Clearly, degeneration in WFG3 poses a big challenge to reference-based algorithms, i.e., MOEA/D, NSGA-III, and SPEA/R, as they roughly pursue diversified population over the whole objective space. On the other hand, an increase in the number of objectives to some extent influences all the tested algorithms. MOEA/D is the most influenced one among six algorithms, which experiences a sharp drop when the number of objectives increases from five to twelve, as indicated by the deterioration of IGD and HV. This is consistent with some recent studies [112, 185]. This observation shows MOEA/D struggles to solve difficult many-objective WFG problems.

To understand why SPEA/R generally performs better than the other algorithms, we graphically plot the parallel coordinates (normalized by the nadir point) of final solutions obtained by each algorithm for the 12-objective WFG4–6 in Figs. 5.8–5.10. The figures clearly show that SPEA/R is able to obtain a good spread of solutions in the entire range of the PF ($f_i \in [0, 2i]$, for all i), whereas HypE, PICEA-g, MOEA/D, and NSGA-III miss some parts of the PF. Due to effective density estimation, SPEA2+SDE shows very competitive diversity performance, but it does not cover well the entire PF. Thus, we can conclude that the outperformance of SPEA/R over the other algorithms results largely from

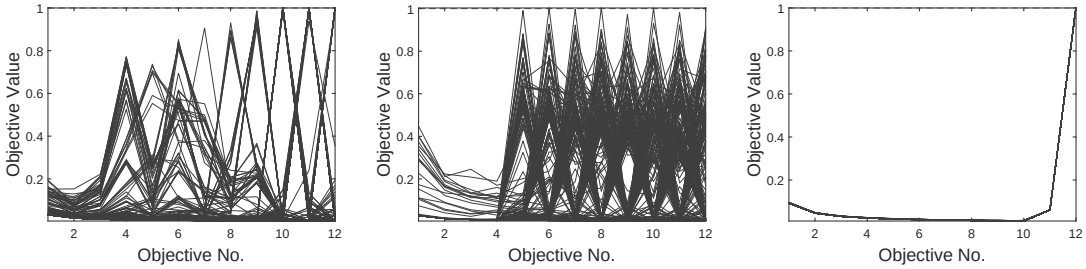


(a) HypE (b) PICEA-g (c) MOEA/D

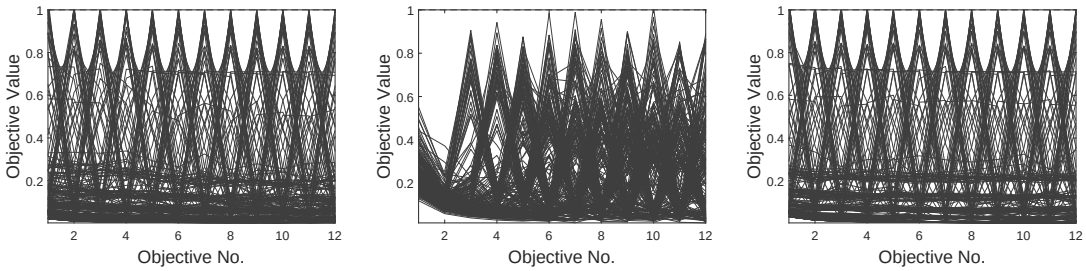


(d) NSGA-III (e) SPEA2+SDE (f) SPEA/R

Fig. 5.9 Parallel coordinates of final solutions obtained by six algorithms for the 12-objective WFG5 instance.



(a) HypE (b) PICEA-g (c) MOEA/D



(d) NSGA-III (e) SPEA2+SDE (f) SPEA/R

Fig. 5.10 Parallel coordinates of final solutions obtained by six algorithms for the 12-objective WFG6 instance.

its sound diversity maintenance and its effective fitness assignment, which are capable of balancing diversity and convergence during the search.

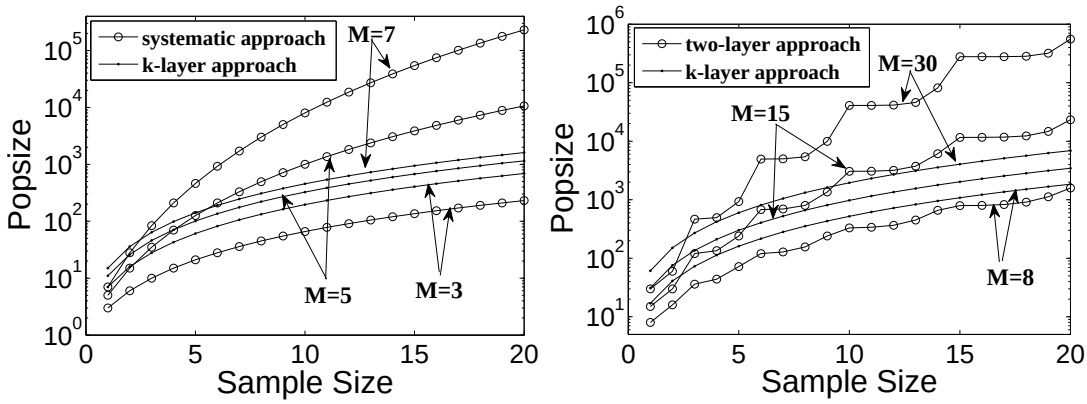


Fig. 5.11 Comparison of the population size required by different methods: (a) the systematic approach and k-layer approach for low-dimensional cases; (b) the two-layer approach and k-layer approach for high-dimensional cases.

5.5 Further Investigations

5.5.1 Comparison of Reference Direction Generation Approaches

As the population size (Popsize) of MOEAs is closely associated with the number of reference directions, we compare our proposed k-layer approach with the systematic approach used in MOEA/D [188] and the two-layer approach used in NSGA-III [43] in terms of required Popsize for different numbers of objectives. Since the two-layer approach is an improved version of the systematic approach for generating reference directions in the case of 7 or more objectives, we just need to compare our k-layer approach with the former and the latter in low-dimensional cases and high-dimensional cases, respectively.

The results are given in Fig. 5.11, where for each approach, 20 different levels of Popsize are continuously sampled. Clearly, for three objectives, the systematic approach shows better Popsize settings than the k-layer approach. However, when M is increased from 5 to 7, the k-layer approach has more choice to set the population size in the range of 10 to 1000. For 8-objective problems, the two-layer approach works slightly better than the k-layer method, but for much higher objectives, the Popsize generated by the two-layer approach grows very fast, particularly for 30 objectives. In this case, the k-layer approach gives more options to configure a desirable population size. Thus, in comparison with the other two approaches, the k-layer approach appears more suitable for generating a reasonable size of population for MaOPs with a large number of objectives.

5.5.1.1 Uniformity

Unlike the simplex-lattice design (SLD) method used in MOEA/D and NSGA-III, our k-layer method generates the reference direction set from the subsimplex's point of view.

Table 5.5 Population size settings for SLD and k-layer for different numbers of objectives

M	SLD				k-lyer			
3	45	105	231	528	43	106	232	511
4	56	120	286	560	57	109	261	541
5	70	126	210	495	71	101	221	451
6	56	126	252	462	55	121	211	463
7	28	84	210	462	36	99	190	456
8	36	120	240	660	41	113	281	521
9	45	165	210	495	46	127	244	487
10	55	110	230	440	51	141	201	441
15	30	135	240	680	31	136	211	526

Table 5.6 Centred L_2-discrepancy values of SLD and k-layer for different population sizes

M	SLD				k-layer			
3	1.13E-01	**1.11E-01**	**1.10E-01**	**1.11E-01**	**1.12E-01**	1.12E-01	1.12E-01	1.12E-01
4	4.87E-01	5.17E-01	5.52E-01	5.89E-01	**4.20E-01**	**4.14E-01**	**4.14E-01**	**4.14E-01**
5	1.18E+00	1.28E+00	1.32E+00	1.43E+00	**9.93E-01**	**9.79E-01**	**9.78E-01**	**9.78E-01**
6	2.39E+00	2.44E+00	2.70E+00	2.80E+00	**1.92E+00**	**1.92E+00**	**1.93E+00**	**1.93E+00**
7	3.78E+00	4.43E+00	4.58E+00	5.05E+00	**3.61E+00**	**3.50E+00**	**3.47E+00**	**3.45E+00**
8	6.46E+00	7.72E+00	**5.46E+00**	6.21E+00	**6.09E+00**	**5.92E+00**	5.84E+00	**5.86E+00**
9	1.26E+01	1.27E+01	1.08E+01	1.43E+01	**9.91E+00**	**9.67E+00**	**9.61E+00**	**9.55E+00**
10	1.98E+01	1.45E+01	1.89E+01	**1.36E+01**	**1.58E+01**	**1.54E+01**	**1.52E+01**	1.53E+01
15	1.58E+02	1.62E+02	**1.31E+02**	1.61E+02	**1.39E+02**	**1.32E+02**	1.33E+02	**1.32E+02**

As a result, the k-layer method can give more options for setting Popsize when the number of objectives is very large. Here, we investigate the uniformity of reference direction set generated by different methods. Since the coupling between Popsize and the number of objectives (M) exists in both SLD and the k-layer method, it is very unlikely to generate the same Popsize using these two methods. A feasible way to compare the uniformity between SLD and the k-layer method is to generate similar Popsize. Also, we consider four levels of Popsize, that is, Popsize approximately equals 50, 100, 250, and 500, which is shown in Table 5.5. Note that, SLD can generate each considered Popsize of points by choosing different settings for the two-layered method of NSGA-III. The most widely used discrepancy measure, i.e., centred L_2-discrepancy (CD_2) [78], is adopted to evaluate the uniformity of a reference direction set, and the smaller CD_2 is, the more uniform a reference direction set will be.

Table 5.6 presents the CD_2 values of SLD and k-layers on different population sizes, where the best value for each population size is marked in boldface. It can be observed that, in the 3-objective case, SLD has slightly better distribution than the k-layer method, whereas for $M > 3$, the k-layer method gives much better CD_2 values than SLD in most

cases. It is understandable that the k-layer method generates a more uniform reference direction set in higher-dimension cases (in objective space), because it decomposes the whole simplex into M subsimplexes, and each subsimplex can get a number of uniformly-distributed points. In contrast, SLD use a two-layered structure to distribute points in high-dimensional cases (in objective space). It generates points only on the layers, and cannot generate intermediate or near intermediate points on the simplex. So, the distribution of SLD is not very uniform in high-dimensional cases (in objective space).

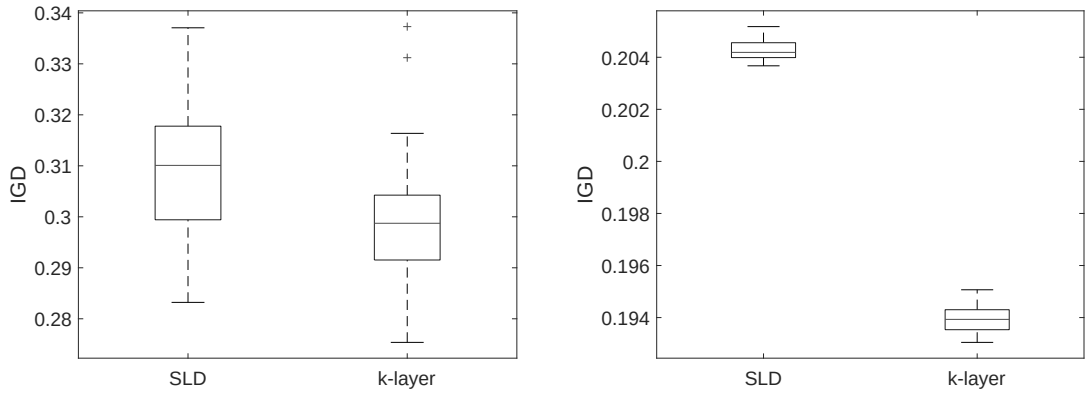
The k-layer method is just an example of using subsimplexes to design the reference direction set. The whole simplex consists of M subsimplexes for the M-objective case, and each subsimplex is a two-dimensional simplex. So, it is easier to distribute points on these subsimplexes than on the whole M-dimensional simplex. However, it is far from being satisfying because there is still a coupling between the Popsize and the number of objectives. Subsimplex-based design methods are promising for reducing the coupling, and more work is required to generate an arbitrary Popsize free from the influence of the number of objectives. It should be noted that a uniformly-distributed reference direction set does not necessarily produce a uniform distribution of solutions on the PF. This is because the task to uniform a reference set on the simplex is not equivalent to that to well distribute solutions on the PF that has various geometries.

5.5.1.2 Influence on Search Performance

Next, we analyze the search performance of algorithms with different reference direction generation methods, i.e., the SLD method and the k-layer method. The two methods are tested on MOEA/D-ACD and SPEA/R. Since both SLD and k-layer methods have a close coupling with Popsize, it is desirable to choose similar Popsizes to make a fair comparison between these two methods. For this reason, we use a Popsize of 105 and 106 for SLD and k-layer, respectively, in 3-objective cases, which has been shown in Table 5.5.

The influence of SLD and k-layer is examined on 3-objective WFG4, and the IGD results obtained by MOEA/D-ACD and SPEA/R are presented in Fig. 5.12. Note that, 5050 uniformly-sampled points from the true PF of the 3-objective WFG4 are used to calculate the IGD metric. Two interesting observations can be obtained from the figure. First, SPEA/R outperforms MOEA/D-ACD in terms of IGD, regardless of the reference direction generation methods. Second, for both algorithms, the k-layer method helps yield better IGD values than the SLD method. Thus, the proposed reference direction generation method is very effective to guide the search toward the 3-dimensional PF (in objective space).

Afterwards, we examine the effectiveness of the k-layer method on high-dimensional cases (in objective space). To do this, we compare the k-layer method with SLD on 8-objective WFG4, and the Popsize of SPEA/R is set to 161. Note that, here SLD represents



(a) MOEA/D-ACD (b) SPEA/R

Fig. 5.12 Boxplots of the IGD results obtained by algorithms using the SLD and k-layer approaches for the 3-objective WFG4 instance.

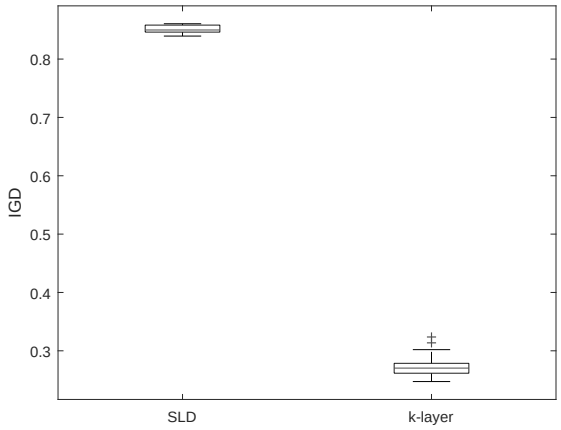
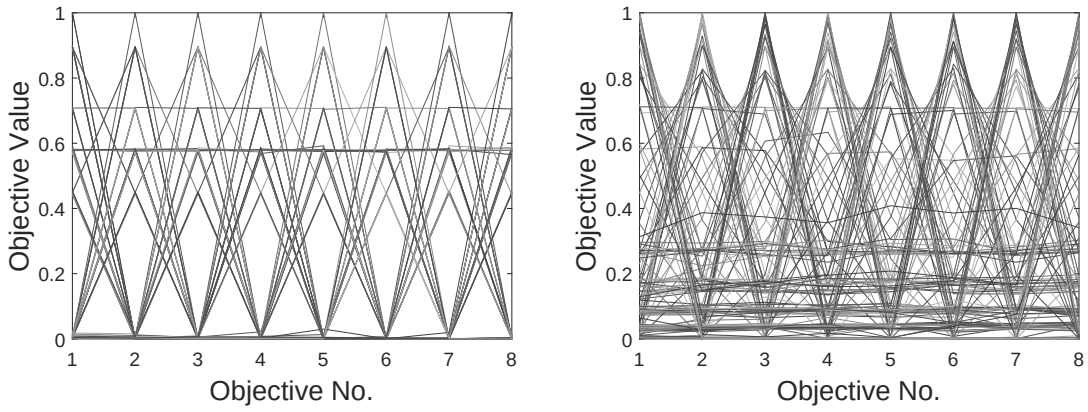


Fig. 5.13 Boxplots of the IGD results obtained by SPEA/R using the SLD and k-layer approaches for the 8-objective WFG4 instance.

the two-layered approach proposed in NSGA-III. Fig. 5.13 presents boxplots of the IGD values obtained by SPEA/R with different reference direction generation methods, showing the k-layer method help achieve better IGD values than SLD. Additionally, the normalized parallel coordinate plot of SPEA/R with the two different methods is displayed in Fig. 5.14. We can observe from the figure that both methods are able to provide a set of solutions residing in the entire PF. While the k-layer method covers well the PF, SLD misses a small part of the PF, e.g., there is no solution in the region where $f_i \in [0.1, 0.4]$ for all i. Therefore, the proposed reference direction generation method can guide the search well in high-dimensional cases (in objective space).



(a) SLD (b) k-layer

Fig. 5.14 Parallel coordinates of final solutions obtained by different reference direction generation methods for the 8-objective WFG4 instance.

5.5.2 SPEA/R vs NSGA-III

SPEA/R and NSGA-III share some similarities in the way that they keep diversity with the aid of reference directions. Besides different methods for constructing reference directions, there are several key differences between them, resulting in distinct search behaviours. First, SPEA/R introduces a restricted mating selection to enhance reproduction instead of random selection, which is very helpful for many-objective optimization where distant parents are not likely to generate good solutions. Second, SPEA/R uses a simple normalization method based on the worst value of each objective, whereas normalization in NSGA-III requires intercept computation and hyperplane construction, which are computationally expensive, particularly for many-objective problems, and NSGA-III may also have difficulty in hyperplane construction due to duplicate extreme points. Third, niche preservation strategies are different in SPEA/R and NSGA-III. Whenever preserving a member from the last front considered, NSGA-III tries to repeatedly identify reference directions having the worst niche count, which is the number of members associated with these reference directions that has been preserved in higher fronts (the higher, the better). This procedure is computationally inefficient. Furthermore, this strategy may result in some isolated but promising members in lower fronts being abandoned if nondominated sorting terminates before considering these lower fronts, implying that population diversity in NSGA-III is still not well maintained. In contrast, as illustrated in Section 5.4.1.2, SPEA/R intentionally gives higher priority to diversity than convergence when performing the environmental selection, leading to impressive performance on MOP test instances, and the niche preservation strategy in SPEA/R has also been further validated on multi- and many-objective WFG problems.

Generally, normalization and niche preservation are all aimed to help keep diversity. To understand the second and third differences, we tested SPEA/R and NSGA-III on dis-

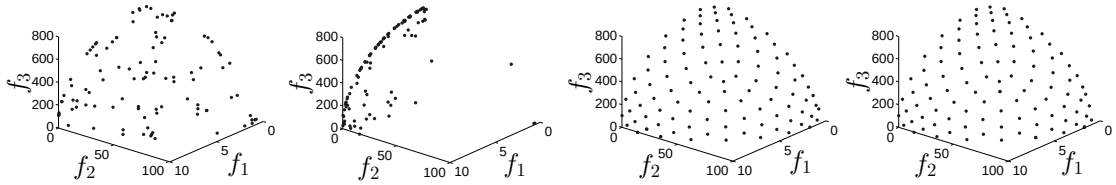


Fig. 5.15 PF approximations for scaled WFG5 in median and worst cases.

parately scaled three-objective WFG5. That is, the objectives f_1, f_2, and f_3 are multiplied with 5, 5^2, and 5^3, respectively. Fig. 5.15 plots PF approximations of the median and worst IGD values over 31 runs, showing that the simple normalization method used in SPEA/R can deal with scaling objectives and the new DFCS strategy is capable of providing a uniform distribution of solutions. NSGA-III, however, struggles to solve the scaled WFG5. In the median case of NSGA-III, the intercept-based normalization is able to diversify points over the whole PF but the niche preservation cannot provide a good distribution. In the worst case, NSGA-III drives the majority of points toward the $f_1 f_3$ plane, and misses a large part of the PF. One reason for this is that NSGA-III tends to preserve members in higher fronts that have better convergence, and less-converged isolated ones in lower fronts are likely to leave unconsidered, leading to poor diversity during the search. Therefore, NSGA-III cannot compete with SPEA/R in terms of diversity.

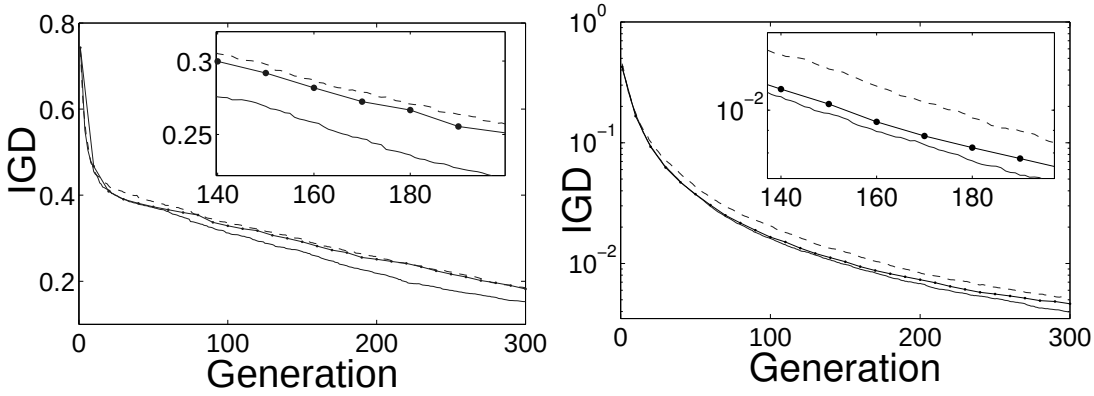
5.5.3 Influence of Fitness Assignment and Niche Preservation

Although Section 5.4.1.2 has revealed that good population diversity contributes to the performance of SPEA/R, in this subsection we try to unveil more reasons behind the high performance of SPEA/R.

Fitness assignment and diversity preservation are the core of SPEA/R, which control the balance between convergence and diversity. To understand why our strategy yields high performance, we further design three other SPEA/R variants that use different strategies. The first one, called SPEA/R-A, removes global fitness from Eq. (5.9) when calculating individuals' fitness. Instead of removing global fitness, the second variant, i.e., SPEA/R-B, removes local fitness, so an individual's final fitness is composed of global fitness and density. The third variant (named SPEA/R-C) sorts individuals and does selection according to fitness values. Thus, SPEA/R-A favours diversity whereas SPEA/R-C favours convergence, and SPEA/R-B does not consider local convergence. The variants are compared with the original SPEA/R on 7 MOP problems and 9 WFG problems with 2, 3, 5, 8, and 12 objectives. The HV results of each algorithm for a total of 52 instances are calculated, and the statistical testing result based on HV is given in Table 5.7, where statistical significance is shown by the Wilcoxon signed-rank test [179] at the 0.05 signifi-

Table 5.7 Statistical difference between SPEA/R and two variants

	Sign	SPEA/R-A	SPEA/R-B	SPEA/R-C
SPEA/R vs.	B	12	4	51
	E	40	47	1
	W	0	1	0



(a) MOP3 (b) WFG4

Fig. 5.16 The IGD metric against the number of generations for two instances: SPEA/R (solid); SPEA/R-A (dashed); SPEA/R-B (dotted).

cance level with Bonferroni correction. In the table, the signs 'B', 'E', and 'W' represent SPEA/R is significantly better than, equivalent to, and significantly worse than the compared variant, respectively.

It is clear that SPEA/R generally performs better than the other variants. Specifically, SPEA/R outperforms SPEA/R-A in a number of cases, indicating that the use of global fitness is a good choice for SPEA/R in some situations. The comparison between SPEA/R and SPEA/R-B shows that the use of local fitness does not make a big difference but may help SPEA/R achieve slightly better performance for a few instances. For SPEA/R-C, the lack of diversity maintenance induces a significant lag behind SPEA/R. This observation further confirms that the high performance of SPEA/R is mainly due to sound diversity preservation.

Since SPEA/R, SPEA/R-A, and SPEA/R-B differ only in fitness assignment, one would wonder why SPEA/R works better (though not significantly better in most cases) than the other two variants. To investigate this, we plot the mean IGD curves of these variants against the first 300 generations on MOP3 and 2-objective WFG4, as shown in Fig. 5.16. It can be observed that SPEA/R converges fastest, followed by SPEA/R-B, and SPEA/R-A ranks last. SPEA/R is better than SPEA/R-A because adding local fitness can strengthen discrimination between individuals so that more-converged individuals can be preserved. In contrast, without the use of global fitness, SPEA/R-A converges relatively slower than SPEA/R and SPEA/R-B. This illustrates that the joint use of global fitness and local fitness can speed up the search process, although not very significantly. However,

we should point out that there is no much difference between SPEA/R and SPEA/R-B in MaOPs. This is because the local fitness value will be zero when the majority of individuals are nondominated in the case of many objectives. In other words, SPEA/R may degenerate to SPEA/R-B in this situation.

5.5.4 Influence of Restricted Mating

Restricted mating selection is somewhat similar to the concept of neighbourhood used in MOEA/D, in which close parents are likely to generate good offspring. It has a key parameter K, i.e., the number of parent candidates, and the influence of this parameter is investigated on four WFG problems. Table 5.8 reports the HV values obtained by SPEA/R with different settings of K. It can be observed that, $K = 20$ (10%–20% of population size) yields better results than the other settings for all the cases except the 8- and 12-objective WFG5 and the 2-objective WFG6. Particularly, for many-objective problems, e.g., the 8- and 12-objective cases, there is a noticeable improvement on the HV metric when K is increased from 2 to 20. This means proper restricted mating can benefit population reproduction, thereby promoting algorithms' performance for many-objective optimization.

The above experiment has shown that proper restricted mating is good for population reproduction. However, we should point out that restricted mating can be used only when population diversity is well maintained. This is because, if population individuals are not well distributed, then restricted mating can cause overexploitation in overcrowded regions so that isolated regions may be left under-explored or even unexplored, resulting in a further deterioration of population diversity. This has been illustrated in Section 5.5.3, where the overlook of diversity maintenance makes SPEA/R-C significantly worse than SPEA/R although restricted mating has been employed there.

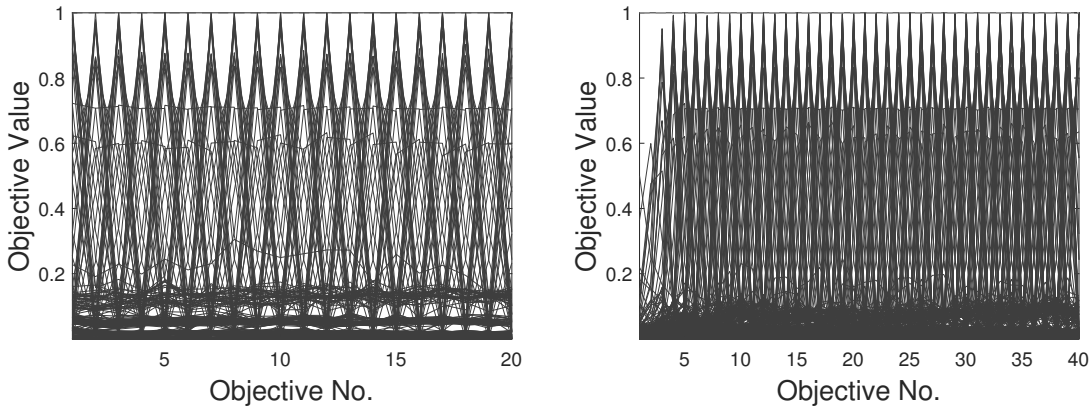
5.5.5 Peformance of SPEA/R on Problems with More Objectives

SPEA/R has the advantage of population diversity maintenance so that it can handle MaOPs. To further investigate whether this advantage can deal with problems with more objectives, we tested SPEA/R on WFG4 with 20 and 40 objectives. This means the difficulty of the problem is massively increased as nearly all population members are nondominated with respect to each other. The population size was set to 280 and 560 for 20 and 40 objectives, respectively. Due to the increase of the difficulty of the problem, SPEA/R should be given more computational resources. Hence, the maximum number of generations was set to 3000 and 5000 for 20 and 40 objectives, respectively.

Fig. 5.17 shows the normalized parallel coordinates of final solutions obtained by SPEA/R for two instances. Clearly, on both 20 and 40 objectives, SPEA/R can still obtain

Table 5.8 Mean and standard deviation HV values obtained by SPEA/R with different K values for four WFG problems

Prob.	M	K=2	K=5	K=10	K=20	K=40	K=50
WFG5	2	7.0444E-1(2.6807E-3)	7.0331E-1(1.4893E-3)	7.0290E-1(1.5121E-3)	**7.1227E-1(9.3626E-4)**	7.0240E-1(1.0190E-3)	7.0225E-1(1.2918E-3)
	3	8.1491E-1(2.2212E-3)	8.1506E-1(2.5168E-3)	8.1509E-1(2.6994E-3)	**8.2437E-1(2.0573E-3)**	8.1398E-1(1.3143E-3)	8.1376E-1(1.1227E-3)
	5	8.7474E-1(2.0417E-3)	8.7699E-1(2.8228E-3)	8.7754E-1(1.4904E-3)	**8.8792E-1(1.1292E-3)**	8.7778E-1(1.3177E-3)	8.7786E-1(1.0863E-3)
	8	8.8822E-1(8.3131E-3)	8.9728E-1(3.1274E-3)	8.9833E-1(1.2579E-2)	9.0023E-1(2.9488E-3)	**9.0064E-1(3.1272E-3)**	8.9896E-1(3.0760E-3)
	12	8.2542E-1(4.0708E-2)	8.3170E-1(6.4925E-2)	8.6648E-1(9.6071E-2)	9.0236E-1(6.6030E-3)	**9.0357E-1(5.3041E-3)**	8.9903E-1(1.3462E-2)
WFG6	2	7.0956E-1(7.4317E-3)	**7.1207E-1(4.9637E-3)**	7.1049E-1(6.8942E-3)	7.1001E-1(6.5120E-3)	7.1125E-1(4.4444E-3)	7.1112E-1(6.4365E-3)
	3	8.1414E-1(8.6380E-3)	8.1048E-1(6.0997E-3)	8.1501E-1(5.5185E-3)	**8.2552E-1(6.3065E-3)**	8.1704E-1(6.4877E-3)	8.1612E-1(6.9906E-3)
	5	8.7020E-1(1.2008E-2)	8.7366E-1(1.2462E-2)	8.7589E-1(8.0459E-3)	**8.9930E-1(7.7577E-3)**	8.7429E-1(1.3280E-2)	8.7756E-1(8.3509E-3)
	8	8.8848E-1(2.9622E-2)	8.9990E-1(1.9014E-2)	9.0585E-1(1.0637E-2)	**9.2855E-1(1.5045E-2)**	9.0146E-1(1.2872E-2)	9.0041E-1(1.2033E-2)
	12	8.6720E-1(8.0397E-2)	8.8930E-1(4.9023E-2)	8.9676E-1(5.2199E-3)	**9.1604E-1(1.2921E-2)**	9.0072E-1(2.5975E-2)	8.9359E-1(6.1651E-2)
WFG7	2	7.3554E-1(5.8256E-4)	7.3573E-1(4.3877E-4)	7.3570E-1(4.2235E-4)	**7.3615E-1(1.1625E-3)**	7.3531E-1(5.4513E-4)	7.3455E-1(1.5055E-3)
	3	8.5109E-1(3.6231E-4)	8.5135E-1(2.8815E-4)	8.5120E-1(3.8447E-4)	**8.5171E-1(3.4081E-4)**	8.5022E-1(3.7511E-4)	8.5007E-1(3.9649E-4)
	5	9.2148E-1(1.0175E-3)	9.2385E-1(5.7867E-4)	9.2498E-1(6.6708E-4)	**9.2528E-1(6.1088E-4)**	9.2504E-1(8.4886E-4)	9.2451E-1(1.7558E-3)
	8	8.1636E-1(7.3633E-2)	8.8675E-1(5.2499E-2)	9.1107E-1(6.1033E-2)	**9.5110E-1(1.7669E-2)**	9.0279E-1(3.2266E-2)	8.4598E-1(6.7261E-2)
	12	9.0203E-1(1.1885E-1)	9.2156E-1(8.3763E-2)	9.3326E-1(3.5592E-2)	**9.4407E-1(9.1515E-3)**	8.8898E-1(8.1680E-2)	8.5109E-1(1.4778E-1)
WFG8	2	6.9842E-1(2.5016E-3)	7.0146E-1(3.4153E-3)	7.0138E-1(3.2326E-3)	**7.0318E-1(3.3615E-3)**	7.0262E-1(3.5139E-3)	7.0264E-1(3.2564E-3)
	3	7.9341E-1(3.4538E-3)	8.0085E-1(6.2528E-3)	8.0478E-1(7.1091E-3)	**8.0651E-1(5.3416E-3)**	8.0817E-1(5.9836E-3)	8.0802E-1(5.7856E-3)
	5	8.0213E-1(3.0860E-3)	8.0994E-1(3.9769E-3)	8.2573E-1(2.0662E-2)	**8.5983E-1(1.8504E-2)**	8.5576E-1(3.6934E-2)	8.4240E-1(1.7857E-2)
	8	8.6003E-1(6.8351E-2)	8.6155E-1(4.5106E-2)	8.7114E-1(4.3606E-2)	**8.8805E-1(3.8120E-2)**	8.6770E-1(5.7081E-2)	8.5679E-1(6.3412E-2)
	12	8.4546E-1(5.2716E-2)	8.5492E-1(2.0422E-1)	8.5888E-1(2.2243E-1)	**8.7084E-1(2.2243E-1)**	8.5445E-1(1.6750E-1)	7.4837E-1(1.6586E-1)



(a) (b)

Fig. 5.17 Parallel coordinates of final solutions obtained by SPEA/R for WFG with 20 (a) and 40 (b) objectives.

a set of diverse solutions in the entire range of the PF. Thus, the proposed DFCS-based selection in SPEA/R is very promising for solving many-objective problems.

5.5.6 Further Discussion

It has been well recognized that convergence and diversity are two main but hard-to-balance goals in designing MOEAs. Any bias toward one goal will inevitably aggravate the other. In many-objective optimization, the balance between them is still of great importance. However, when handling MaOPs, most MOEAs inherit elitist preservation from their counterparts of multiobjective optimization that emphasizes nondominated solutions in the population, resulting in very little room left for diversity maintenance. Even if these MOEAs did not intentionally emphasize convergence, they could not elude the fact that an increasingly large fraction of population becomes nondominated with an increase in the number of objectives. In other words, they perform environmental selection in a convergence-first-and-diversity-second manner. As a result, when the MOEAs are applied to many-objective optimization, there will be a large number of nondominated individuals after the convergence-first selection, and diversity preservation will be performed only on the nondominated individuals. Correspondingly, some regions occupied by dominated individuals will be scarcely explored, and diversity preservation becomes of limited use in this case. In contrast, SPEA/R adopts a DFCS strategy to perform environmental selection, at an attempt to maximize population diversity and strengthen exploitation in less-converged regions during the search. Our experiments have shown its promise for both multiobjective and many-objective optimization.

However, we may wonder why SPEA/R can work well on problems with over 12 objectives, where nearly all individuals (over 95% of population) are nondominated [87].

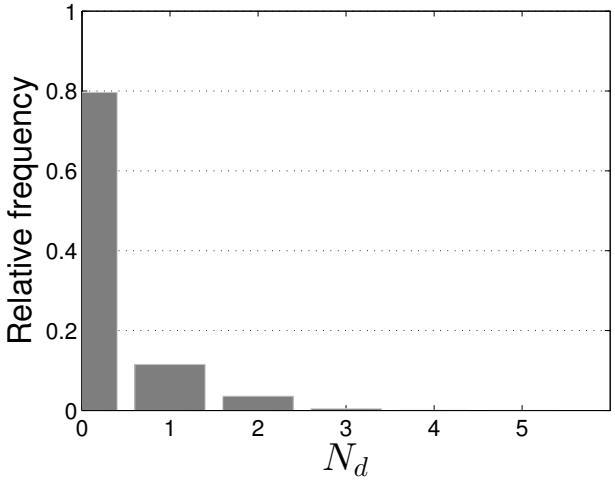


Fig. 5.18 The relative frequency of the number of subregions occupied only by dominated solutions.

This means, in this situation, the DFCS strategy in SPEA/R has no advantage over other MOEAs in diversity preservation because there is hardly any region that can be occupied by very few dominated individuals. There is no doubt that, when population is randomly generated, the fraction of dominated individuals is close to zero for 10 objectives and over [87]. But, what if the population is a combination of parent and offspring populations, which is the case with MOEAs? To investigate this, we consider the search behaviour of SPEA/R on the 12-objective WFG5 over 2000 generations. In every generation, SPEA/R distributes a combined population toward H_M^k subregions (which equals the total number of reference directions) of the objective space, and the number (N_d) of subregions in which only dominated solutions reside is recorded. Fig. 5.18 shows the relative frequency of different N_d values over 2000 generations. Clearly, in the majority of generations dominated solutions do not solely occupy any subregions. In this situation, dominated solutions make little contribution to diversity as nondominated solutions covers all subregions of the evolving population. However, there are also over 20% generations in which some subregions are occupied by dominated but not nondominated solutions. In this case, dominated solutions make a difference to population diversity. Additionally, we also compute the percentage of dominated solutions in the combined population of every generation of SPEA/R for a single run, as shown in Fig. 5.19. It can be observed from the figure that, there is still a noticeable proportion of dominated solutions in the combined population before the population converges to the POF. All these observations clearly confirm that preservation of dominated solutions for diversity promotion through the DFCS strategy is still beneficial to SPEA/R when handing many objectives.

On the other hand, Fig. 5.19 can also be used to explain why the compared MOEAs in this work cannot compete with SPEA/R. As shown in this figure, there are at least 50% nondominated solutions in the combined population nearly every generation. Since

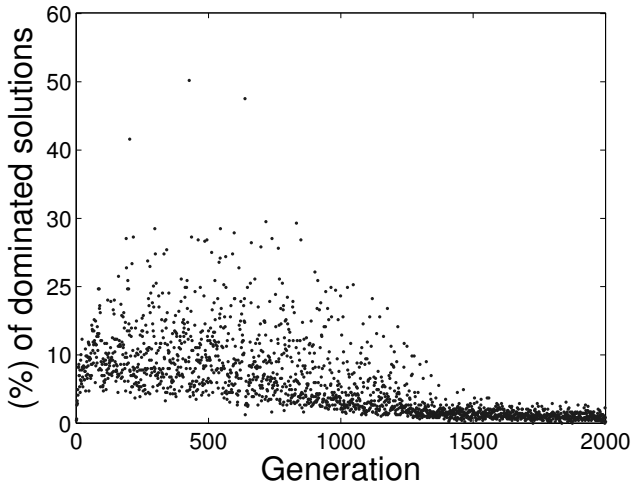


Fig. 5.19 The percentage of dominated solutions in every generation of SPEA/R for 12-objective WFG5.

HypE, MOEA/D, PICEA-g, NSGA-III, and SPEA2+SDE all prefer nondominated solutions, there is no room for them to preserve dominated but diverse solutions when selecting only half of the combined population for next generation. As a consequence, regions occupied by dominated solutions will be left unexplored, which can cause diversity loss in the population.

The reason why our observation is inconsistent with the study of [87] in terms of the proportion of dominated solutions is that, the combined population comprises parent and offspring members, and there is a close relationship between them. Thus, there are more dominated members than expected. However, we should be aware that there might be very few or even no dominated solutions if the number of objectives is considerably large, e.g., 100. In this case, the DFCS selection strategy may be of limited use.

5.6 Summary

It has been repeatedly reported that conventional Pareto-dominance based MOEAs may be unsuitable for many-objective optimization, although they can successfully solve two- or three-objective problems. In this chapter, we have suggested a diversity-first-and-convergence-second method, i.e., DFCS, for population sorting. DFCS is used to revive an early SPEA algorithm for handling both MOPs and MaOPs. Through incorporating a set of predefined reference directions, the proposed algorithm, i.e., SPEA/R, partitions the objective space into a number of subregions of interest, and individuals in each subregion are guided along predefined search directions. Unlike most existing MOEAs preferring nondominatd solutions, SPEA/R adopts the DFCS selection strategy, which can increase the selection pressure for many-objective optimization where a large fraction of popu-

lation is nondominated. SPEA/R also employs a restricted mating scheme to improve reproduction efficiency. Besides, the proposed framework has significantly reduced the computational effort of SPEA-based methods, providing the overall computational complexity bounded by $O(MN^2)$.

Our experimental study has demonstrated the efficacy of SPEA/R on a number of MOP and WFG test problems with 2 to 40 objectives and various optimization difficulties. A fair comparison with several state-of-the-art MOEAs suggests that SPEA/R is very comparative for both multiobjective and many-objective optimization. This implies that giving high priority to diversity over convergence can be another effective way to handle many-objective optimization.

Although SPEA/R has provided encouraging performance on the test problems considered in this work, it needs to be examined on a wider range of problems (e.g., complicated PS and PF shapes). Also, as the research on many-objective optimization is still in its infancy, there are some open issues remaining to be solved, such as the computationally expensive calculation of performance metrics and visualization of a many-objective trade-off front. Therefore, these should be very interesting topics for our future work.

Chapter 6

A Test Environment for Dynamic Multiobjective Optimization

Artificial benchmark problems have played a fundamental role in determining whether a dynamic MOEA has the ability to solve DMOPs. Furthermore, benchmark problems contribute to analyzing and identifying the strengths and weaknesses of an MOEA in order to modify it and improve its performance. However, one of the main issues in the field of evolutionary dynamic multiobjective optimization (EDMO) is that there is a lack of effective standard test functions, and a few publications [55, 73, 76, 130, 183] pointed out that there should be more investigations of DMOPs to promote the research of EDMO. Such issue is, however, yet to gain as much attention as the fields of static MOPs [102, 116, 142] and dynamic single-objective optimization problems [13, 37, 40, 69, 96, 111, 118, 133, 161, 173, 183].

Beside the lack of effective and diverse test environments for EDMO, there is little work on understanding the strengths and weaknesses of MOEAs in dynamic environments. Without any knowledge of these, it will be hard for researchers and practitioners to develop effective techniques that can handle DMOPs well.

This chapter makes an attempt to fill the above-mentioned gaps. First, a new test environment is proposed to facilitate theoretical analysis for EDMO. Then, extensive algorithm comparisons are conducted based on the proposed test suite.

This chapter is organized as follows. Section 6.1 introduce some related work and the incentive of this research. Section 6.2 presents the proposed test environment, where details of each test problem are explained and a comparison with other existing test suites is illustrated. In Section 6.3, algorithm comparisons based on the proposed test suite are presented. Section 6.4 concludes the chapter and points out future research directions in EDMO.

6.1 Introduction

The first research on constructing DMOPs was conducted by Jin and Sendhoff [96], who proposed to aggregate different objectives of existing static MOPs and vary the weights dynamically. Later, Farina *et al.* [55] made a clear classification of DMOPs according to the possible effects of environmental changes on the PF/PS, and they also considered several dynamic scenarios that may appear in dynamic environments. Following those scenarios, they further suggested a test suite of five FDA test functions based on the existing ZDT [200] and DTLZ [48] test suites of static MOPs. Since then, various dynamic benchmarks have sprung up in the literature [11, 23, 33, 67, 81, 93, 130].

Analyzing those existing DMOPs, we can observe that most of commonly used DMOPs are adapted from the ZDT [200] and DTLZ [48] test suites. In other words, they are the variants of the FDA [55] test problems. As a consequence, they share more or less same or similar properties (e.g., same objective functions). Furthermore, few have taken into account the following characteristics: (1) mixed PFs in terms of convexity and concavity that change over time; (2) complicated diversity-resistant schemes that hinder a set of diverse solutions; (3) problems that can change between different types during the evolution; (4) non-monotonic and time-varying relationship between variables instead of static monotonic variable-linkage used in the literature. The lack of the above-mentioned characteristics in DMOPs implies that the DMOPs currently used in the literature are not sufficiently diverse and challenging. Therefore, developing a new set of test functions that covers those characteristics to compare the performance of DMOAs becomes meaningful and essential, which is greatly needed in the domain of EDMO.

Based on the understanding of the limitations of current DMOP test problems and inspired by the works of [48, 55, 82, 200], this work attempts to design a diverse EDMO test environments in order to facilitate theoretical analysis.

6.2 Proposed Test Suites

6.2.1 The Proposed Benchmark Generator

The proposed benchmark generator is based on the following framework:

$$JY: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(h(\mathbf{x_I}) + A_t \sin(W_t \pi h(\mathbf{x_I})))^{\alpha_t} \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-h(\mathbf{x_I})+A_t \sin(W_t \pi h(\mathbf{x_I})))^{\beta_t}, \end{cases} \tag{6.1}$$

where $0 \leq h(\mathbf{x_I}) \leq 1$, and $\mathbf{x_I}$ and $\mathbf{x_{II}}$ are sub-vectors of the decision vector $\mathbf{x}$. A_t and W_t are two parameters to control the local shape of the PF, with A_t adjusting the curvature and W_t controlling the number of mixed convex and concave segments on the PF. A large value of W_t causes the PF to have disconnected regions, while a small value produces a continuous PF. Here, W_t is recommended to be an integer. α_t and β_t ($\alpha_t > 0$, $\beta_t > 0$) are parameters that control the overall shape of the PF: when $\alpha_t > 1$ and $\beta_t > 1$ or $\alpha_t < 1$ and $\beta_t < 1$, the overall shape is convex or concave, respectively; when $\alpha_t = \beta_t = 1$, the overall shape is linear; otherwise, the overall shape is mixed. $g(\mathbf{x_{II}}, t)$ is a non-negative function, hindering algorithms from converging towards the true PF. The minimum of $g(\mathbf{x_{II}}, t)$ is zero. Thus, Eq. (6.1) can produce various PF geometries by properly configuring relevant parameters. Generally, the mathematical description of the continuous PF for Eq. (6.1) is as follows:

$$f_1^{\frac{1}{\alpha_t}} + f_2^{\frac{1}{\beta_t}} = 1 + 2A_t \sin\left(W_t \pi \frac{f_1^{\frac{1}{\alpha_t}} - f_2^{\frac{1}{\beta_t}} + 1}{2}\right), \tag{6.2}$$

where the values of A_t and W_t must enable Eq. (6.1) to be a continuous PF. To have a better understanding of the proposed generator, we denote $F_1 = \sqrt[\alpha_t]{f_1} - \sqrt[\beta_t]{f_2}$ and $F_2 = \sqrt[\alpha_t]{f_1} + \sqrt[\beta_t]{f_2}$. This means that a clockwise rotation with an angle $\pi/4$ is made from the current coordinate axis. Then, Eq. (6.2) can be rewritten as:

$$F_2 = 1 + 2A_t \sin\left(W_t \pi \frac{F_1 + 1}{2}\right), \tag{6.3}$$

where a sine wave is described if $W_t \neq 0$ and $A_t \neq 0$. Thus, the proposed generator has a wave-like geometry, containing both concave and convex regions. Figure 6.1 illustrates examples of PFs of JY with linear and non-linear overall shapes.

6.2.2 Test Instances

In this work, we concentrate on $h(\mathbf{x_I}) = x_1$, although we recognize that movement across the PF can be achieved by adjusting a number of variables, i.e., the use of rotation matrices for $h(\mathbf{x_I})$ and the normalization of $h(\mathbf{x_I})$. Below, we provide ten benchmark instances with detailed information for a number of types of changes in the environment.

$JY1$, as shown in Eq. (6.4), is a Type I problem, where the PS changes over time in a regular pattern, with $x_i = G(t)$, $\forall x_i \in \mathbf{x_{II}}$. It mainly tests the convergence speed and reactivity of an algorithm, and fast-convergence algorithms can easily solve this problem. The control of processing plants [81], where the PS varies in different time-dependent scenarios and controllers are required to have a fast response speed, can be a corresponding

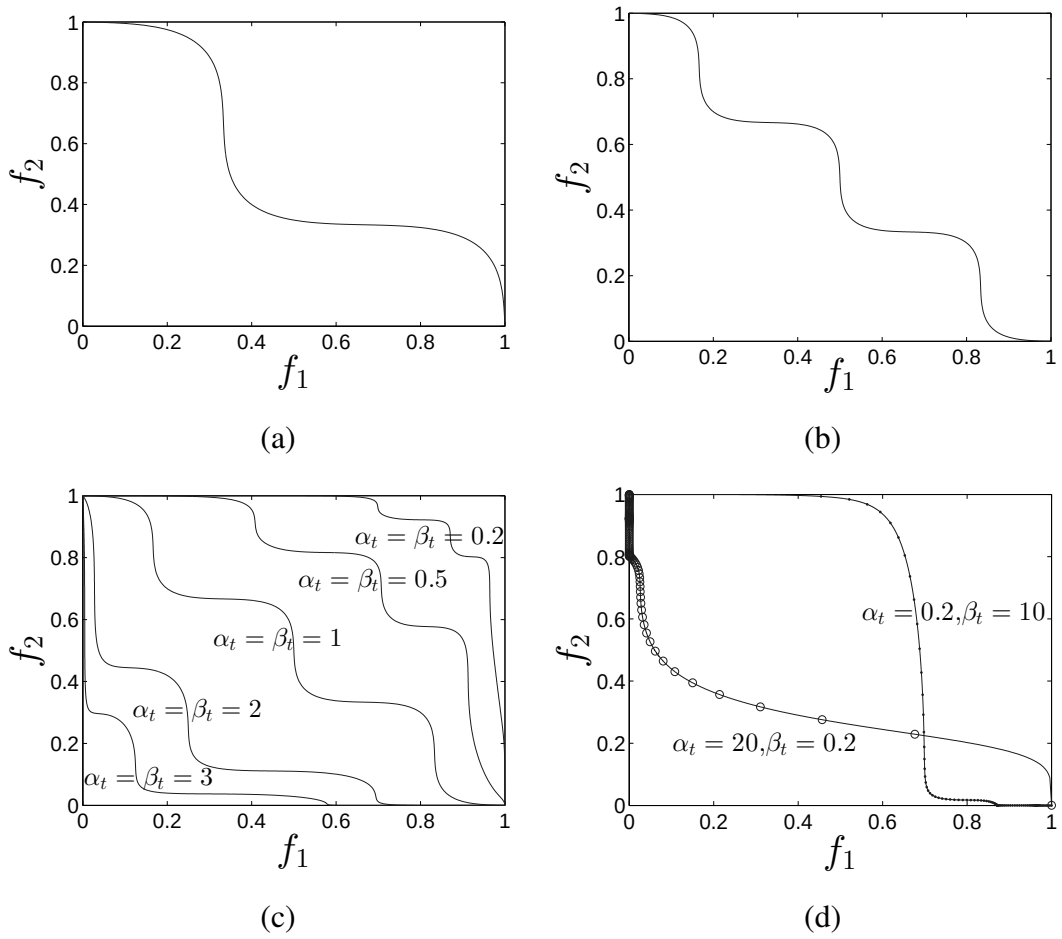


Fig. 6.1 PFs of *JY* with different overall shapes: (a) $\alpha_t = \beta_t = 1$, $A_t = 0.1$, and $W_t = 3$; (b) $\alpha_t = \beta_t = 1$, $A_t = 0.05$, and $W_t = 6$; (c) convex or concave overall shapes with $A_t = 0.05$ and $W_t = 6$; (d) mixed overall shapes with $A_t = 0.05$ and $W_t = 6$.

real-world application of $JY1$.

$$JY1: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2, G(t) = sin(0.5\pi t) \\ A_t = 0.05, \quad W_t = 6 \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.4}$$

$JY2$, as shown in Eq. (6.5), is a Type II problem with dynamic PFs and PSs. The PS changes over time, and the objective vector oscillates among several modes. As a result, the PF, as illustrated in Fig. 6.2, changes its shape over time (refer to Eq. (6.2)). A similar real-world application is the electric power supply problem [104], where the objective

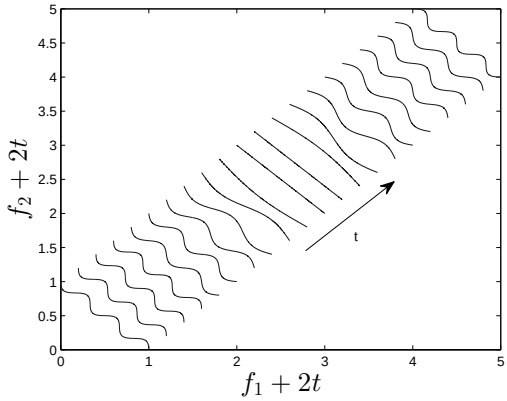


Fig. 6.2 PF of $JY2$ with 21 time windows varying from 0 to 2. For a better visualization, f_1+2t and f_2+2t are shown on the x and y axes, respectively.

functions oscillate among several optimization modes, resulting in the change of optimal solutions in real time.

$$JY2: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2, G(t) = sin(0.5\pi t) \\ A_t = 0.05, W_t = \lfloor 6sin(0.5\pi(t-1)) \rfloor \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2,...,x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.5}$$

$JY3$, as shown in Eq. (6.6), introduces time-varying non-monotonic dependencies between any two decision variables, which is close to real-world problems like the greenhouse system [165]. The PF of $JY3$ is similar to that of $JY2$, and the PS is $y_1 = |x_1 sin((2\alpha+0.5)\pi x_1)|$, $\alpha = \lfloor 100 sin^2(0.5\pi t) \rfloor$, $y_i = \sqrt{y_{i-1}}$, $i = 2,...,n$, meaning that each variable has different amount of change.

$$JY3: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(y_1 + A_t \sin(W_t \pi y_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-y_1 + A_t \sin(W_t \pi y_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (y_i^2 - y_{i-1})^2, A(t) = 0.05 \\ W_t = \lfloor 6\sin(0.5\pi(t-1)) \rfloor, \alpha = \lfloor 100\sin^2(0.5\pi t) \rfloor \\ y_1 = |x_1 \sin((2\alpha+0.5)\pi x_1)|, y_i = x_i, i = 2,...,n \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2,...,x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.6}$$

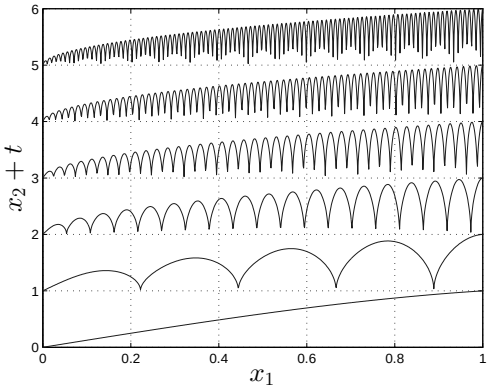


Fig. 6.3 PS of $JY3$ for the first two variables with 6 time windows varying from 0 to 0.5. For a better visualization, x_1 and x_2+t are shown on the x and y axes, respectively.

The PS for variables x_1 and x_2 is shown in Fig. 6.3, where the dependency between these two variables is non-monotonic and becomes increasingly complicated as time elapses. Furthermore, the density of solutions also changes over time. Therefore, $JY3$ not only assesses the effect of variable-linkage but also tests the diversity performance of an algorithm in a dynamic environment.

$JY4$, as shown in Eq. (6.7), is constructed to have a time-changing number of disconnected PF segments, which is the case with hydro-thermal power scheduling [44] where the PF is discontinuous. This problem may pose challenges to some algorithms to find all the PF components. As illustrated in Fig. 6.4, the PF of $JY4$ is subject to the definition of Eq. (6.2), but has a number of disconnected segments. The PS is $x_i = G(t), \forall x_i \in \mathbf{x_{II}}$.

$$JY4: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i - G(t))^2, G(t) = sin(0.5\pi t) \\ A_t = 0.05, \quad W_t = 10^{1+|G(t)|} \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.7}$$

Contrary to the above problems, $JY5$ (as shown in Eq. (6.8)) does not have a mixed PF and is a Type III problem. Its PF is very simple and changes from convex geometry to concave geometry. The PF is defined in Eq. (6.2) and illustrated in Fig. 6.5. A similar real-life problem that has changing PF shapes can be referred to the route guidance system [168].

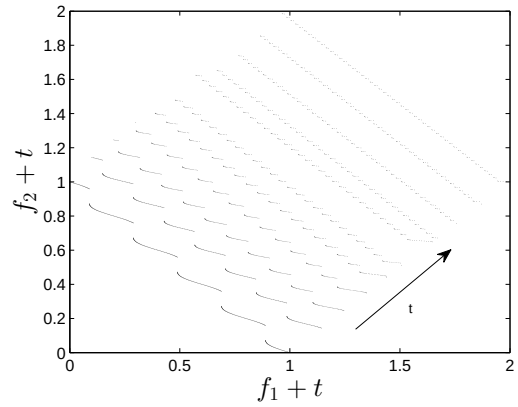


Fig. 6.4 PF of $JY4$ with 11 time windows varying from 0 to 2. For a better visualization, f_1+t and f_2+t are shown on the x and y axes, respectively.

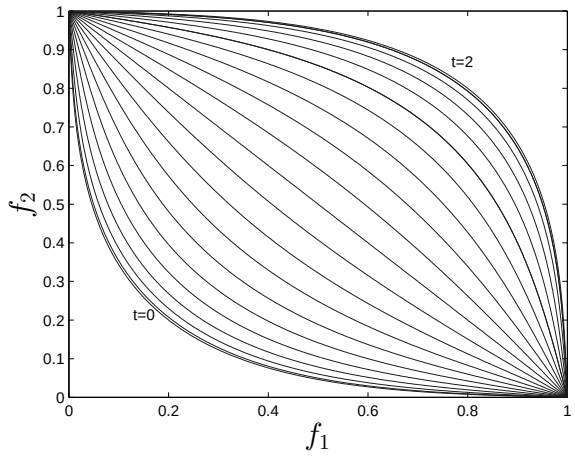


Fig. 6.5 PF of $JY5$ with 21 time windows varying from 0 to 2.

$$JY5: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} x_i^2, A_t = 0.3 \sin(0.5\pi(t-1)), W_t = 1 \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2,...,x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.8}$$

The above-generated instances are all unimodal, and they are not sufficiently challenging. In contrast, $JY6$ is a multimodal problem, where not only the number of local optima changes over time, but also the PS is dynamically shifted. The PF of JY6 remains stationary, and its PS is $x_i = G(t), \forall x_i \in \mathbf{x_{II}}$.

$$JY6: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (4y_i^2 - \cos(K_t \pi y_i) + 1) \\ A_t = 0.1, \quad W_t = 3, K_t = 2 * \lfloor 10 * |G(t)| \rfloor \\ G(t) = \sin(0.5\pi t), y_i = x_i - G(t), \quad i = 2, ..., n \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.9}$$

$JY7$ takes into account the shift of PS, multimodality, and the overall shape of the PF. Different from $JY6$, the number of local optima in $JY7$ remains fixed, and the overall PF shape can be concave or convex due to environmental changes. The PS is $x_i = G(t)$, $\forall x_i \in \mathbf{x_{II}}$, and the PF is mathematically described in Eq. (6.2) and similarly illustrated in Fig. 6.1(c). A multimodal real-world problem similar to $JY7$ can be the dynamic speed reducer design [180, 191].

$$JY7: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1))^{\alpha_t} \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1))^{\beta_t} \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (y_i^2 - 10\cos(2\pi y_i) + 10) \\ A_t = 0.1, W_t = 3, \alpha_t = \beta_t = 0.2 + 2.8 * |G(t)| \\ G(t) = \sin(0.5\pi t), y_i = x_i - G(t), \quad i = 2, ..., n \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.10}$$

$$JY8: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1))^{\alpha_t} \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1))^{\beta_t} \\ g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} x_i^2, G(t) = \sin(0.5\pi t) \\ A_t = 0.05, W_t = 6, \alpha_t = \frac{2}{\beta_t}, \beta_t = 10 - 9.8 * |G(t)| \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}. \end{cases} \tag{6.11}$$

In $JY8$ (see Eq. (6.11)), the PS remains static, but the PF changes over time. The dynamism of $JY8$ lies in the change of its overall PF shape, in which the geometry and

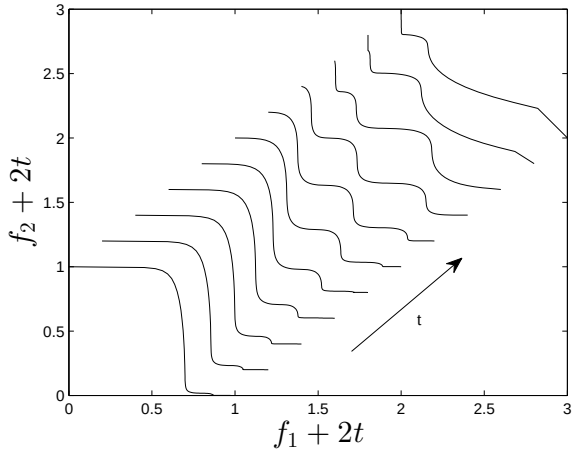


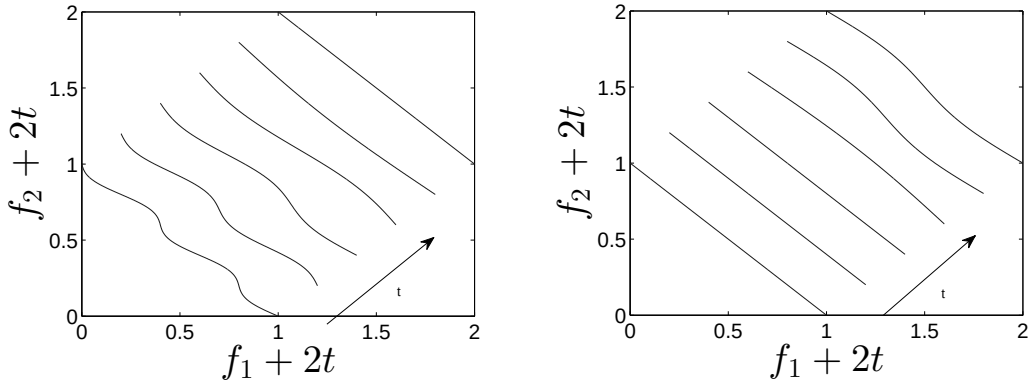
Fig. 6.6 PF of $JY8$ with 11 time windows varying from 0 to 1. For a better visualization, $f_1 + 2t$ and $f_2 + 2t$ are shown on the x and y axes, respectively.

the number of mixed segments of the PF vary over time. The PF is defined in Eq. (6.2) and illustrated in Fig. 6.6.

The first three types of change can be easily realized when constructing test functions, and they have been widely reported in the literature. However, many real-world optimization problems with dynamic characteristics, e.g., the controller selection for dynamic plants [81], may jump between types. To the best of our knowledge, none of this kind of test function has been introduced into the family of DMOPs. In this work, we propose such a problem that cyclically switches from Type I to Type II, then to Type III. Technically, this kind of problem is macroscopically a Type II problem from the perspective of the whole period of changes. Despite that, we would refer to this kind of change as the Mixed Type from a microscopic angle, which can help us to analyze the performance of algorithms on a problem with changing types. This type of problem can be formulated as follows:

$$JY9: \begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x_{II}},t) = \sum\limits_{x_i \in \mathbf{x_{II}}} (x_i + \sigma - G(t))^2, G(t) = |sin(0.5\pi t)| \\ A_t = 0.05, \quad W_t = \lfloor 6 \sin^{\sigma}(0.5\pi(t-1)) \rfloor, \quad \sigma \equiv \lfloor \frac{\tau}{\tau_t \rho_t} \rfloor \pmod 3 \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1,1]^{n-1}, \end{cases} \tag{6.12}$$

where ρ_t represents the frequency of type change, and is suggested as $\rho_t = 5$, meaning that the current type lasts 5 time windows. If $\sigma = 0$, $JY9$ is a Type I problem, and the PS is $x_i = G(t)$, $\forall x_i \in \mathbf{x_{II}}$, the PF is referred to Eq. (6.2) and similar to Fig. 6.1(b). If $\sigma = 1$,



(a) (b)

Fig. 6.7 PF of $JY9$ with 12 time windows varying from: (a) 0.5 to 1; (b) 1 to 1.5. For a better visualization, $f_1 + 2t$ and $f_2 + 2t$ are shown on the x and y axes, respectively.

$JY9$ belongs to Type II, where the PS is $x_i = G(t) - 1$, $\forall x_i \in \mathbf{x_{II}}$, and the PF is referred to Eq. (6.2) and illustrated in Fig. 6.7(a). If $\sigma = 2$, $JY9$ is a Type III problem with the PS being $x_i = -1$, $\forall x_i \in \mathbf{x_{II}}$, and the PF not being Eq. (6.2) since $g(\mathbf{x_{II}}, t) \neq 0$. In this case, the minimum of $g(\mathbf{x_{II}}, t)$ is $g^*(t) = (n-1)(1-G(t))^2$. Thus, the PF is:

$$f_1 + f_2 = (1 + g^*(t))(1 + 2A_t \sin(W_t \pi(\frac{f_1 - f_2}{2(1 + g^*(t))} + \frac{1}{2}))), \tag{6.13}$$

where the PF is illustrated in Fig. 6.7(b).

To increase the flexibility of type changes, that is, the problem can be any type after a type change, we introduce a more challenging problem as follows:

$$JY10: \begin{cases} min \quad F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\ f_1(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}, t))(x_1 + A_t \sin(W_t \pi x_1))^{\alpha_t} \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x_{II}}, t))(1 - x_1 + A_t \sin(W_t \pi x_1))^{\beta_t} \\ g(\mathbf{x_{II}}, t) = \sum\limits_{x_i \in \mathbf{x_{II}}} (x_i + \sigma - G(t))^2, G(t) = |sin(0.5\pi t)| \\ A(t) = 0.05, \quad W(t) = 6 \\ \alpha_t = \beta_t = 1 + \sigma G(t), \sigma \equiv (\lfloor \frac{\tau}{\tau_t \rho_t} \rfloor + R) \pmod 3 \\ \mathbf{x_I} = (x_1) \in [0, 1], \mathbf{x_{II}} = (x_2, ..., x_n) \in [-1, 1]^{n-1}, \end{cases} \tag{6.14}$$

where $JY10$ is almost defined the same as $JY9$, however, in addition to two time-dependent parameters α_t and β_t that control the simple overall PF shape, $JY10$ introduces a random integer $R \in [1, 3]$ to switch the problem into a random type of change every ρ_t time windows. This randomness makes the test problem hard to optimize. The PS of

$JY10$ is the same as that of $JY9$, but the PF is a little different, which is defined as:

$$f_1^{\frac{1}{\alpha_t}} + f_2^{\frac{1}{\beta_t}} = (1+g^*(t))(1+2A_t\sin(W_t\pi(\frac{f_1^{\frac{1}{\alpha_t}} - f_2^{\frac{1}{\beta_t}}}{2(1+g^*(t))} + \frac{1}{2}))), \tag{6.15}$$

where $g^*(t)$ is defined the same as in $JY9$, and the PF is similar to that of $JY9$ (as illustrated in Eq. (6.13)) except that there are two more time-varying parameters, i.e., α_t and β_t, in Eq. (6.15). When $JY10$ is in Type III ($\sigma = 2$), the PF not only has a time-varying overall shape, but also shifts over time because of the time-changing value of $g^*(t)$. Due to the stochastic nature of change, it is not possible to draw the actual time-changing PFs of $JY10$.

In total, ten test instances are developed in this chapter. In practice, more complicated benchmark problems can be generated by further varying parameters A_t, W_t, α_t, or β_t over time.

6.2.3 Comparion with Other Benchmarks

Table 6.1 presents a comparison between some existing test suites and the proposed one, where the main characteristics of each test suite are briefly tabulated. On the basis of the comparison, we would like to highlight the following features of the JY test suite:

1. The JY test suite introduces a new type of change, i.e., Mixed Type, to help classify DMOPs. Correspondingly, two Mixed Type instances are included in this test suite.

2. In addition to sharing some common PF properties, most of the JY instances have mixed convex/concave components on the PF, and the number of mixed components is controllable and can be time-changing.

3. There is a JY problem with nonlinear, time-varying and non-monotonic variable linkages, while variable linkages in other test suites, e.g., UDF and ZJZ, are monotonic, which may be easy to crack by hill-climbing methods.

4. JY develops a problem with random changes on the problem type, that is, the problem can be any types when a change occurs.

5. JY includes a problem with time-varying multimodality. This means the difficulty of the problem changes over time due to the changing number of local optima.

6. JY problems are newly developed whereas many existing DMOPs are adapted from their static counterparts or from the FDA test suite.

Table 6.1 Comparison of characteristics involved in some existing test suites for EDMO

Test Suite	Problem Type	PF Geometry	Variable linkage	Randomness	Other Features	Origination
FDA[55]	Type I, Type II, Type III	continuous, concave, convex, time-varying curvature and/or location	no	no	unimodal	ZDT[200], DTLZ[48]
dMOP[61]	Type I, Type II, Type III	continuous, convex, time-varying curvature	no	random selection of diversity-related variables for dMOP3	unimodal	FDA[55]
DSW[130]	Type II	continuous, convex, time-varying location	no	no	unimodal, disconnected PS	Schaffer[149]
DIMP[105]	Type I	continuous, concave, convex	no	no	unimodal, multimodal, variables have different amount of change	ZDT[200]
T[81]	Type III, Type IV	continuous, concave, convex	no	no	unimodal, multimodal, time-varing number of objectives/variables	DTLZ[48]
HE[76]	Type III	continuous, discontinous, time-varying curvature	static and nonlinear linkage	no	unimodal, multimodal, isolated, deceptive	ZDT[200], LZ[120], WFG[82]
UDF[11]	Type II, Type III	continuous, discontinuous, time-varying curvature and/or location	nonlinear, time-varying and monotonic linkage	random selection of PF variations for UDF9 and UDF10	unimodal, multimodal, variables have different amount of change	UF[193]
ZJZ[197]	Type II	continuous, time-varying curvature	nonlinear, time-varying and monotonic linkage	no	unimodal	FDA[55]
JY	Type I, Type II, Type III, Mixed Type	mixed PF components, continuous, discontinuous, time-varying curvature and/or location	nonlinear, time-varying and non-monotonic linkage	random selection of problem types for JY10	unimdal, multimodal, time-varying multimodality, variables have different amount of change	New

6.2.4 Discussions

Changing factors in real-world applications vary from problem to problem. Generally speaking, changing factors can be objective functions, decision variables, time-linkage, constraints, switch-mode changes, as revealed by the survey in [133]. As a consequence, real-life problems present various dynamic characteristics, such as time-changing PS and/or PF, disconnectivity, multimodality, undetectability, periodicity, and predictability. On the other hand, it is often hard or even impossible to use only a few artificial benchmarks to simulate all dynamic real-life problems as problems have a variety of mathematical definitions. Therefore, in our work, the JY test problems are mainly aimed to imitate several representative dynamics of real-world applications instead of defining similar mathematical formulations, and these test problems have some dynamic properties in common with the referenced real-world examples.

Although there is a lack of clear link that to what extent test problems reflect real-world scenarios, it is not trivial to use these test problems to test and improve the performance of MOEAs. After all, what if these test problems appear in real-world applications and algorithms struggle to handle them? In this sense, the test problems can serve as a useful tool to identify strengths and weaknesses of algorithms before these algorithms can be applied to real-world DMOPs. Nevertheless, the test problems are far from being realistic, and it is necessary to further investigate their relation to real-world applications.

6.3 Experimental Studies

6.3.1 Performance Metrics

Performance metrics are of great importance to assessing MOEAs. The averaged IGD [196] is widely used for performance assessment in EDMO. Beside this indicator, we develop several new EDMO performance metrics based on existing static metrics. The following is devoted to detailing these new metrics.

6.3.1.1 Averaged Spacing

The spacing (S) metric [150] is an important indicator for quantifying algorithms' performance in static multiobjective optimization. However, it has not been employed for dynamice multiobjective optimization. A method of extending the S metric to examine the performance of algorithms for DMOPs is to define the average S metric over all time steps in a run:

$$\overline{S} = \frac{1}{T_s} \sum_{t=1}^{T_s} S(t), \tag{6.16}$$

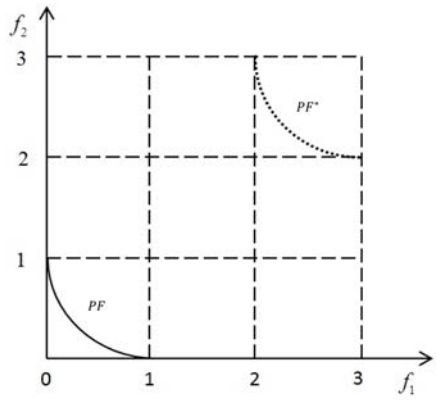


Fig. 6.8 An example of the obtained PF^* far from the PF.

where $S(t)$ refers to the S metric at time instance t and is calculated just before the next change occurs, and T_s is the number of time steps.

6.3.1.2 Maximum Spread

The maximum spread (MS), first introduced by Zitzler *et al.* [200], measures to what extent the extreme members in the true PF have been reached. Goh and Tan [60] proposed a modified version of MS by taking into account the proximity of PF^* towards PF (see Eq. (2.11)).

A large value of MS indicates a good spread of PF^*, and MS will have a value of one when PF^* covers the whole PF [60]. Sometimes, however, a high MS value can be deceptive. Figure 6.8 gives such an example, where PF^* is far from PF, and MS equals one since $min[\overline{PF_k}, \overline{PF_k^*}]$ is smaller than $max[\underline{PF_k}, \underline{PF_k^*}]$ in Eq. (2.11), $k = 1, 2$. In this case, the maximum spread is not justifiable and may cause a misleading understanding of the approximation PF^*. For this reason, we propose a revised maximum spread (RMS) as follows:

$$RMS = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left[\frac{\mu_L([\underline{PF_k^*}, \overline{PF_k^*}] \bigcap [\underline{PF_k}, \overline{PF_k}])}{\mu_L([\underline{PF_k}, \overline{PF_k}])}\right]^2}, \tag{6.17}$$

where $\mu_L(A)$ represents the Lebesgue measure [98] of the set A. Let us consider again the example illustrated in Fig. 6.8, by computing RMS, we can get RMS=0. This is reasonable since PF^* does not converge well, not to mention spread widely over the PF.

In order to apply the RMS metric to evaluate the performance of algorithms for DMOPs, an alternative method is to calculate the average of the RMS values over T_s time steps:

$$\overline{RMS} = \frac{1}{T_s}\sum_{t=1}^{T_s} RMS(t), \tag{6.18}$$

where $RMS(t)$ represents the RMS value at time instance t.

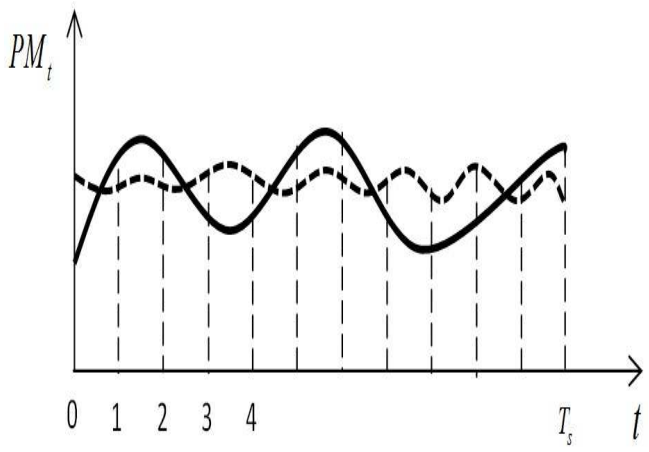


Fig. 6.9 Illustration of performance measure against time.

6.3.1.3 Robustness

In dynamic environments, it is desirable to make algorithms as robust as possible. In other words, algorithms must be able to resist changes and be immune to an amount of uncertainty and perturbation. There have been some studies on robustness performance for dynamic optimization [57, 97], showing that robustness is a quite important goal in dynamic environments. Besides, many existing performance measures are adapted from their static counterparts and may not be suitable for dynamic environments. For example, suppose that PM_t (the smaller, the better) is a performance measure of population P_t at time t, the average value of PM_t over some time steps in a run is commonly used in dynamic optimization [61, 124, 196, 197]. As a compact form of assessment, average values are helpful for measuring algorithms' performance, but it can not reflect the robustness performance. As illustrated in Fig. 6.9, the dashed and solid curves respectively represent the performance of two algorithms. Both algorithms have the same or similar average values in terms of PM_t, but the dashed is more robust than the other one, thus it is clear that the dashed achieves better performance on PM_t.

In this work, the robustness of an algorithm on PM_t can be defined as:

$$R(PM) = \sqrt{\frac{1}{T_s - 1} \sum_{t=1}^{T_s} (PM_t - \overline{PM})^2}, \tag{6.19}$$

where $R(PM)$ denotes the robustness of the metric PM over T_s time steps and $\overline{PM}$ is the average of PM values over T_s time steps, i.e., $\overline{PM} = \sum_{t=1}^{T_s} PM_t / T_s$. A smaller value of $R(PM)$ indicates a better robustness performance on the PM metric. Note that, PM can be any unary performance metric pertinent to the multiobjective optimization goals of proximity, diversity and distribution. With this definition of robustness, in Fig. 6.9, the

dashed algorithm will achieve better performance than the solid one in terms of robustness on the performance measure PM_t.

6.3.2 Compared Algorithms and Parameter Settings

This work considers six MOEAs from the literature and compares their performance on the proposed test suite to assess the property and difficulty of these benchmark functions. These algorithms are classified into two groups. The first group includes four well-known dynamic MOEAs: dynamic multi-objective particle swarm optimization (DMOPSO) algorithm [110], dynamic non-dominated sorting genetic algorithm II (DNSGA-II) [44], dynamic competitive-cooperative coevolutionary algorithm (dCOEA) [61], and population prediction strategy (PPS) [196]. Each algorithm in this group has a mechanism of dealing with dynamism for dynamic optimization. Note that, there are two versions of DNSGA-II, and DNSGA-II with randomly created solutions whenever a change occurs is adopted here. The second group includes two classic MOEAs: strength pareto evolutionary algorithm II (SPEA2) [202] and MOEA based on decomposition (MOEA/D) [188], and they are high-performance algorithms for static multiobjective optimization. To handle dynamic environments, the algorithms of this group adopt the following strategy in our experiments: 10% randomly selected population members are re-evaluated for change detection, and the restart scheme is employed for change response. The parameter settings for all the tested algorithms are inherited from the referenced papers.

The experiments were conducted at different combinations of change severity levels and frequencies, i.e., (n_t, τ_t)=(5,10), (10,10) and (10,5). To guarantee the fairness for all the tested algorithms, the total number of changes for problems $JY1-JY10$ was set to 20 during the evolution. Besides, 100 more generations were given to each algorithm before the first change to minimize the potential effect of static optimization. Thus, the total number of generations for running an algorithm was $100+20\tau_t$. For each problem, the number of decision variables was set to 10, and each tested algorithm, with a population size of 100, was executed 30 runs, and the experimental results were recorded. The experimental comparison employs $\overline{SP}$, $\overline{RMS}$, $\overline{IGD}$, and $R(PM)$ as the performance metrics, and the IGD metric is selected as the indicator PM for the robustness metric. That is, we use $R(IGD)$ to reflect the robustness performance of MOEAs for EDMO. Furthermore, for the computation of the IGD metric, 500 uniformly-distributed points (using the kth nearest neighbour truncation method [202]) were sampled from the true PF at each time step.

6.3.3 Experimental Results

The experimental results reported in Tables 6.2 to 6.5 are the average values, standard errors and individual ranks of four performance metrics, where the best results are highlighted in boldface. For each test instance and each algorithm, we conducted the Wilcoxon rank-sum test [79] at the 0.05 significance level to judge whether the results of a studied algorithm are significantly different from another compared algorithms. An algorithm that outperforms most of the competitors will rank the first, and the one that outperforms the least will be assigned the worst rank. In the case that several algorithms outperform the same number of other algorithms, they share the same rank. The rank of each algorithm for each test case is written in the tables in the superscript style. The last row of each table presents the average rank of all the algorithms over the entire test cases.

Table 6.2 presents the $\overline{S}$ values of the algorithms for instances $JY1 - JY10$. It can be clearly observed from the table that, in most cases, dCOEA and MOEA/D maintain the best population diversity. This performance may be attributed to the fact that, dCOEA employs a multi-population strategy to keep diversity during the search and introduces stochastic competitors to increase diversity when a change occurs, while MOEA/D always keeps a set of well-diversified subproblems regardless of environmental changes. Both schemes are able to maintain a high level of population diversity for handling dynamic environments. It is not surprising that DNSGA-II achieves good diversity performance for instances $JY5$ and $JY8$ because their PS remains stationary. In this case, the optimization task is to adjust the distribution of solutions on the time-varying PFs, and since this algorithm adopts the crowding distance to maintain a good distribution of solutions on the PF, it is able to achieve good $\overline{S}$ values on these two instances. Besides, DNSGA-II also performs the best in solving $JY3$, showing its effectiveness in handling strong dependencies between variables.

Table 6.2 also reveals the poor performance of DMOPSO and SPEA2 on the test cases. The following factors may be used to explain their ineffectiveness: (1) when the environment changes, DMOPSO only updates particles' memory and SPEA2 does nothing except restarting the optimization procedure, neither of them handles population diversity; (2) DMOPSO uses niche sharing to penalize crowded regions and SPEA2 use the k-th nearest neighbour density estimation technique to truncate redundant non-dominated solutions. However, if there are not enough solutions in the archive, these diversity maintenance techniques may not be effective to keep a set of uniformly distributed solutions; (3) the environmental changes may be too fast, which does not give DMOPSO and SPEA2 enough time to converge towards the moving PFs. The average rank in the last row of the table further confirms that dCOEA and MOEA/D outperform other algorithms in finding a diverse solution set.

Table 6.2 Mean $\overline{S}$ metric values, standard deviations and individual ranks for benchmarks $JY1 - JY10$

Prob.	(n_t, τ_t)	DMOPSO	DNSGA-II	PPS	dCOEA	SPEA2	MOEA/D
JY1	(5,10)	2.2115E-1(4.3181E-2)$^{(5)}$	3.0053E-1(8.4264E-2)$^{(6)}$	9.0481E-2(2.1140E-2)$^{(3)}$	**3.4435E-2(5.7330E-3)**$^{(1)}$	1.5994E-1(2.4313E-2)$^{(4)}$	3.5696E-2(1.4402E-2)$^{(2)}$
	(10,10)	1.1300E-1(1.3754E-2)$^{(5)}$	7.7553E-2(1.3524E-2)$^{(4)}$	6.9667E-2(3.7147E-2)$^{(3)}$	**3.5201E-2(3.3871E-3)**$^{(1)}$	1.5725E-1(2.7141E-2)$^{(6)}$	3.6545E-2(1.0309E-2)$^{(2)}$
	(10,5)	1.7068E-1(2.0702E-2)$^{(4)}$	1.6201E-1(2.5580E-2)$^{(3)}$	1.8589E-1(7.2232E-2)$^{(5)}$	7.3354E-2(2.3643E-2)$^{(2)}$	4.7084E-1(8.4109E-2)$^{(6)}$	**4.9500E-2(1.8202E-2)**$^{(1)}$
JY2	(5,10)	2.5414E-1(4.7194E-2)$^{(5)}$	2.8999E-1(6.2836E-2)$^{(6)}$	9.4642E-2(2.5067E-2)$^{(3)}$	**3.1899E-2(4.2768E-3)**$^{(1)}$	1.5662E-1(2.8437E-2)$^{(4)}$	7.6892E-2(2.9605E-2)$^{(2)}$
	(10,10)	1.0570E-1(1.6203E-2)$^{(5)}$	7.3577E-2(1.0402E-2)$^{(4)}$	5.8400E-2(3.7087E-2)$^{(3)}$	**3.0149E-2(7.0852E-3)**$^{(1)}$	1.5497E-1(1.8478E-2)$^{(6)}$	5.6367E-2(2.0744E-2)$^{(2)}$
	(10,5)	1.6693E-1(2.3111E-2)$^{(4)}$	1.4708E-1(2.6310E-2)$^{(3)}$	1.7542E-1(6.2242E-2)$^{(4)}$	6.0333E-2(2.0225E-2)$^{(2)}$	4.4891E-1(4.9997E-2)$^{(6)}$	**5.1339E-2(2.5281E-2)**$^{(1)}$
JY3	(5,10)	1.6121E+0(5.8397E-1)$^{(6)}$	**2.7795E-1(2.5869E-1)**$^{(1)}$	9.1244E-1(6.1568E-1)$^{(4)}$	3.4565E-1(2.7870E-1)$^{(2)}$	9.4053E-1(1.6043E-1)$^{(5)}$	3.5185E-1(1.3527E-1)$^{(2)}$
	(10,10)	1.5851E+0(5.3717E-1)$^{(6)}$	**1.2822E-1(1.4974E-1)**$^{(1)}$	1.1067E+0(6.2737E-1)$^{(5)}$	4.5379E-1(3.4603E-1)$^{(3)}$	9.1987E-1(1.4254E-1)$^{(4)}$	3.4869E-1(1.1914E-1)$^{(2)}$
	(10,5)	1.7713E+0(6.8890E-1)$^{(4)}$	**2.7044E-1(2.5630E-1)**$^{(1)}$	1.8411E+0(1.1272E+0)$^{(4)}$	5.8129E-1(3.9429E-1)$^{(3)}$	2.1558E+0(5.0641E-1)$^{(6)}$	5.5521E-1(2.4389E-1)$^{(2)}$
JY4	(5,10)	1.5992E+0(3.1996E-1)$^{(6)}$	1.9757E-1(3.7545E-2)$^{(4)}$	2.0596E-1(4.8453E-2)$^{(5)}$	1.8777E-1(7.5667E-2)$^{(3)}$	1.6214E-1(1.5659E-2)$^{(2)}$	**1.1217E-2(9.8391E-3)**$^{(1)}$
	(10,10)	8.9479E-1(2.8350E-1)$^{(6)}$	8.5118E-2(2.7838E-2)$^{(2)}$	1.2354E-1(3.8062E-2)$^{(3)}$	1.3199E-1(4.1564E-2)$^{(4)}$	1.8082E-1(1.8060E-2)$^{(5)}$	**9.8490E-3(3.9175E-3)**$^{(1)}$
	(10,5)	7.1210E-1(2.3538E-1)$^{(6)}$	1.8808E-1(5.9123E-2)$^{(3)}$	1.7271E-1(7.2708E-2)$^{(2)}$	2.1283E-1(7.1460E-2)$^{(4)}$	3.1718E-1(3.6567E-2)$^{(5)}$	**2.0509E-2(9.7608E-3)**$^{(1)}$
JY5	(5,10)	1.0915E-1(3.2658E-3)$^{(6)}$	**1.0189E-2(8.6466E-4)**$^{(1)}$	5.1189E-2(3.6838E-2)$^{(4)}$	1.6717E-2(1.4872E-3)$^{(2)}$	9.4649E-2(1.2576E-2)$^{(5)}$	3.9316E-2(1.1992E-2)$^{(3)}$
	(10,10)	7.4534E-2(2.5147E-2)$^{(5)}$	**1.1924E-2(4.4726E-3)**$^{(1)}$	5.2573E-2(3.0670E-2)$^{(4)}$	1.6566E-2(1.1316E-3)$^{(2)}$	8.8768E-2(1.5110E-2)$^{(6)}$	3.4718E-2(1.1565E-2)$^{(3)}$
	(10,5)	1.3024E-1(2.8187E-2)$^{(4)}$	**1.5551E-2(1.0621E-2)**$^{(1)}$	1.5845E-1(1.9475E-2)$^{(5)}$	3.2323E-2(4.5466E-3)$^{(2)}$	1.9331E-1(2.6183E-2)$^{(6)}$	5.3093E-2(2.0056E-2)$^{(3)}$
JY6	(5,10)	2.3786E+0(3.6653E-1)$^{(6)}$	2.2028E+0(4.8599E-1)$^{(5)}$	1.2026E+0(8.5695E-2)$^{(4)}$	3.7838E-1(2.0813E-1)$^{(2)}$	1.1728E+0(1.4320E-1)$^{(3)}$	**1.7892E-2(8.1106E-3)**$^{(1)}$
	(10,10)	1.5829E+0(2.3463E-1)$^{(5)}$	1.8823E+0(4.0778E-1)$^{(6)}$	1.2046E+0(1.8651E-1)$^{(3)}$	3.1709E-1(1.3053E-1)$^{(2)}$	1.2099E+0(1.5283E-1)$^{(4)}$	**2.1475E-2(1.6156E-2)**$^{(1)}$
	(10,5)	1.8280E+0(2.4787E-1)$^{(4)}$	2.3224E+0(6.4197E-1)$^{(5)}$	1.5387E+0(3.0404E-1)$^{(3)}$	9.2784E-1(4.8779E-1)$^{(2)}$	2.4235E+0(3.9681E-1)$^{(6)}$	**2.1836E-2(1.6212E-2)**$^{(1)}$
JY7	(5,10)	5.2149E+0(6.8540E-1)$^{(6)}$	3.7440E+0(6.1899E-1)$^{(5)}$	3.6775E+0(3.6788E-1)$^{(4)}$	**4.2160E-1(3.3697E-1)**$^{(1)}$	1.6964E+0(2.6702E-1)$^{(2)}$	2.0636E+0(5.6442E-1)$^{(3)}$
	(10,10)	5.3287E+0(7.7166E-1)$^{(6)}$	2.8484E+0(5.7687E-1)$^{(4)}$	3.6382E+0(3.0270E-1)$^{(5)}$	**2.6136E-1(1.2485E-1)**$^{(1)}$	1.6639E+0(2.5295E-1)$^{(3)}$	1.1806E+0(5.5606E-1)$^{(2)}$
	(10,5)	5.3969E+0(6.4479E-1)$^{(6)}$	3.6371E+0(6.7081E-1)$^{(4)}$	4.2909E+0(4.4701E-1)$^{(5)}$	7.1534E-1(3.0057E-1)$^{(2)}$	3.3671E+0(4.1572E-1)$^{(3)}$	**6.0389E-1(4.2520E-1)**$^{(1)}$
JY8	(5,10)	1.5464E-1(3.1713E-2)$^{(5)}$	**1.0843E-2(1.5233E-3)**$^{(1)}$	5.3281E-2(1.8849E-2)$^{(4)}$	1.8058E-2(2.6451E-3)$^{(2)}$	1.9474E-1(3.4346E-2)$^{(6)}$	3.8355E-2(1.1226E-2)$^{(3)}$
	(10,10)	1.4694E-1(2.8485E-2)$^{(5)}$	**1.4780E-2(8.9017E-3)**$^{(1)}$	3.8167E-2(2.5342E-2)$^{(4)}$	2.0867E-2(3.1221E-3)$^{(2)}$	1.8539E-1(3.4419E-2)$^{(6)}$	3.5295E-2(9.3868E-3)$^{(3)}$
	(10,5)	2.3745E-1(3.9448E-2)$^{(5)}$	**2.1189E-2(1.7770E-2)**$^{(1)}$	9.8063E-2(2.3370E-2)$^{(4)}$	3.4647E-2(5.9556E-3)$^{(2)}$	5.6375E-1(1.0371E-1)$^{(6)}$	6.8335E-2(2.0631E-2)$^{(3)}$
JY9	(5,10)	5.4384E-1(1.1455E-1)$^{(5)}$	6.9630E-1(1.6179E-1)$^{(6)}$	3.6902E-1(6.0664E-2)$^{(4)}$	6.0311E-2(1.4386E-2)$^{(2)}$	1.4644E-1(1.9297E-2)$^{(3)}$	**3.7047E-2(1.2077E-2)**$^{(1)}$
	(10,10)	3.7622E-1(7.4034E-2)$^{(5)}$	3.6938E-1(1.0053E-1)$^{(5)}$	2.4072E-1(3.6293E-2)$^{(4)}$	2.9119E-2(3.9191E-3)$^{(2)}$	1.5110E-1(2.4794E-2)$^{(3)}$	**3.4303E-2(1.1667E-2)**$^{(1)}$
	(10,5)	4.0103E-1(1.0485E-1)$^{(5)}$	3.1714E-1(5.6264E-2)$^{(3)}$	3.2785E-1(4.5412E-2)$^{(4)}$	1.0499E-1(7.0416E-2)$^{(2)}$	4.7488E-1(5.2222E-2)$^{(6)}$	**4.2714E-2(2.0723E-2)**$^{(1)}$
JY10	(5,10)	6.0150E-1(2.0020E-1)$^{(6)}$	3.2654E-1(1.3470E-1)$^{(3)}$	3.3913E-1(4.7806E-2)$^{(4)}$	**4.7730E-2(1.7151E-2)**$^{(1)}$	2.4936E-1(8.0549E-2)$^{(2)}$	3.9099E-1(1.8213E-1)$^{(5)}$
	(10,10)	4.1814E-1(2.2511E-1)$^{(6)}$	3.4747E-1(3.0096E-1)$^{(4)}$	2.6679E-1(4.4989E-2)$^{(2)}$	**4.1091E-2(1.2047E-2)**$^{(1)}$	2.6268E-1(1.1449E-1)$^{(2)}$	3.9584E-1(1.2693E-1)$^{(5)}$
	(10,5)	6.0456E-1(2.0783E-1)$^{(5)}$	4.8471E-1(2.3782E-1)$^{(4)}$	3.8023E-1(4.5400E-2)$^{(3)}$	**1.6459E-1(1.1568E-1)**$^{(1)}$	7.0212E-1(2.4336E-1)$^{(6)}$	3.2560E-1(2.4470E-1)$^{(2)}$
rank		6	3	4	1	5	2

Table 6.3 Mean $\overline{RMS}$ metric values, standard deviations and individual ranks for benchmarks $JY1 - JY10$

Prob.	(n_t, τ_t)	DMOPSO	DNSGA-II	PPS	dCOEA	SPEA2	MOEA/D
JY1	(5,10)	**1(0)**$^{(1)}$	9.9999E-1(3.2465E-7)$^{(2)}$	9.9912E-1(3.5339E-4)$^{(3)}$	9.5816E-1(9.8363E-3)$^{(6)}$	9.9909E-1(5.9294E-4)$^{(4)}$	9.7922E-1(8.1579E-3)$^{(5)}$
	(10,10)	**1(0)**$^{(1)}$	9.9987E-1(3.5809E-4)$^{(2)}$	9.9906E-1(4.1814E-4)$^{(4)}$	9.6271E-1(1.1780E-2)$^{(6)}$	9.9917E-1(5.0906E-4)$^{(3)}$	9.7800E-1(7.7631E-3)$^{(5)}$
	(10,5)	**1(0)**$^{(1)}$	9.9998E-1(5.3940E-4)$^{(2)}$	9.9566E-1(2.0018E-3)$^{(3)}$	9.5815E-1(1.2602E-2)$^{(5)}$	9.9303E-1(1.9295E-3)$^{(4)}$	9.5034E-1(1.3157E-3)$^{(6)}$
JY2	(5,10)	**1(0)**$^{(1)}$	9.9995E-1(1.6266E-4)$^{(2)}$	9.9900E-1(2.6726E-4)$^{(3)}$	9.5975E-1(1.0485E-2)$^{(6)}$	9.9811E-1(4.4254E-4)$^{(4)}$	9.7528E-1(9.6337E-3)$^{(5)}$
	(10,10)	**1(0)**$^{(1)}$	9.9978E-1(4.3952E-4)$^{(2)}$	9.9905E-1(2.4062E-4)$^{(3)}$	9.5680E-1(1.2611E-2)$^{(6)}$	9.9844E-1(2.9795E-4)$^{(4)}$	9.7494E-1(9.0494E-3)$^{(5)}$
	(10,5)	**1(0)**$^{(1)}$	9.9996E-1(1.5875E-4)$^{(2)}$	9.9629E-1(1.4581E-3)$^{(3)}$	9.4872E-1(1.7932E-2)$^{(6)}$	9.9116E-1(2.2180E-3)$^{(4)}$	9.5278E-1(1.0165E-2)$^{(5)}$
JY3	(5,10)	**1(0)**$^{(1)}$	9.2753E-1(1.6563E-1)$^{(3)}$	7.5296E-1(3.0290E-1)$^{(5)}$	2.6686E-1(1.3579E-1)$^{(6)}$	9.1774E-1(1.1188E-2)$^{(4)}$	9.3233E-1(4.3157E-2)$^{(2)}$
	(10,10)	**1(0)**$^{(1)}$	8.6286E-1(2.6734E-1)$^{(5)}$	8.7780E-1(2.1439E-1)$^{(4)}$	3.1828E-1(1.7272E-1)$^{(6)}$	9.1387E-1(1.5497E-2)$^{(3)}$	9.4710E-1(4.1344E-2)$^{(2)}$
	(10,5)	**1(0)**$^{(1)}$	9.1820E-1(1.7955E-1)$^{(3)}$	7.3352E-1(2.4975E-1)$^{(5)}$	2.9578E-1(1.1712E-1)$^{(6)}$	8.6372E-1(1.3534E-2)$^{(4)}$	9.7648E-1(3.7596E-2)$^{(2)}$
JY4	(5,10)	9.8302E-1(1.4597E-3)$^{(2)}$	9.7614E-1(1.2568E-2)$^{(4)}$	**9.8519E-1(3.0860E-4)**$^{(1)}$	9.2414E-1(2.1227E-2)$^{(6)}$	9.8278E-1(1.1903E-3)$^{(3)}$	9.2831E-1(1.5238E-2)$^{(5)}$
	(10,10)	9.8362E-1(1.3336E-3)$^{(3)}$	9.7826E-1(3.4126E-3)$^{(4)}$	9.8594E-1(5.3743E-4)$^{(2)}$	9.2796E-1(2.2971E-2)$^{(6)}$	**9.9777E-1(1.1098E-3)**$^{(1)}$	9.2799E-1(1.5013E-2)$^{(5)}$
	(10,5)	9.8000E-1(2.1633E-3)$^{(3)}$	9.7397E-1(9.7036E-3)$^{(4)}$	9.8478E-1(9.5697E-4)$^{(2)}$	9.1911E-1(2.5120E-2)$^{(5)}$	**9.9589E-1(1.8165E-3)**$^{(1)}$	9.1306E-1(1.4408E-2)$^{(6)}$
JY5	(5,10)	**1(0)**$^{(1)}$	9.9959E-1(3.6413E-4)$^{(2)}$	9.9916E-1(4.3684E-4)$^{(3)}$	9.4969E-1(1.0827E-2)$^{(6)}$	9.9873E-1(3.4385E-4)$^{(4)}$	9.8468E-1(2.8648E-2)$^{(5)}$
	(10,10)	**1(0)**$^{(1)}$	9.9916E-1(9.3236E-4)$^{(3)}$	9.9929E-1(2.4941E-4)$^{(2)}$	9.5120E-1(1.0035E-2)$^{(6)}$	9.9871E-1(3.1110E-4)$^{(4)}$	9.9850E-1(2.4088E-2)$^{(5)}$
	(10,5)	**1(0)**$^{(1)}$	9.9814E-1(1.9238E-3)$^{(2)}$	9.9675E-1(7.9841E-4)$^{(3)}$	9.3398E-1(1.3265E-2)$^{(6)}$	9.9449E-1(1.3927E-3)$^{(4)}$	9.9251E-1(2.2933E-2)$^{(5)}$
JY6	(5,10)	**1(0)**$^{(1)}$	1.0000E+0(1.2098E-7)$^{(2)}$	9.9184E-1(2.4440E-3)$^{(3)}$	8.6658E-1(3.6885E-2)$^{(6)}$	9.8531E-1(3.6202E-3)$^{(4)}$	9.7467E-1(8.5146E-3)$^{(5)}$
	(10,10)	**1(0)**$^{(1)}$	9.9997E-1(1.2127E-4)$^{(2)}$	9.9039E-1(4.4466E-3)$^{(3)}$	8.7947E-1(2.5634E-2)$^{(6)}$	9.8513E-1(2.9510E-3)$^{(4)}$	9.7641E-1(9.6097E-3)$^{(5)}$
	(10,5)	**1(0)**$^{(1)}$	9.9954E-1(2.0746E-3)$^{(2)}$	9.4255E-1(1.7664E-2)$^{(3)}$	8.3131E-1(2.8317E-2)$^{(6)}$	9.1314E-1(1.9895E-2)$^{(5)}$	9.2885E-1(2.4804E-2)$^{(4)}$
JY7	(5,10)	9.9178E-1(5.8574E-4)$^{(1)}$	**9.9182E-1(1.2689E-2)**$^{(1)}$	9.0733E-1(1.0826E-2)$^{(4)}$	8.7990E-1(3.0917E-2)$^{(5)}$	9.5959E-1(1.5128E-2)$^{(3)}$	5.3937E-1(5.5567E-1)$^{(6)}$
	(10,10)	**9.9488E-1(4.8072E-4)**$^{(1)}$	9.9286E-1(9.8146E-3)$^{(2)}$	9.4809E-1(1.3820E-2)$^{(4)}$	8.9970E-1(2.2385E-2)$^{(5)}$	9.5493E-1(1.9552E-2)$^{(3)}$	5.1396E-1(6.1510E-2)$^{(6)}$
	(10,5)	**9.9337E-1(8.4847E-4)**$^{(1)}$	9.9157E-1(1.2335E-2)$^{(2)}$	9.0219E-1(2.7365E-2)$^{(3)}$	8.4798E-1(3.3826E-2)$^{(5)}$	9.0482E-1(2.9825E-2)$^{(3)}$	3.7993E-1(7.2743E-2)$^{(6)}$
JY8	(5,10)	9.2233E-1(1.7493E-3)$^{(2)}$	9.0951E-1(4.2461E-4)$^{(4)}$	9.1733E-1(3.2631E-3)$^{(3)}$	8.6593E-1(5.8224E-4)$^{(6)}$	**9.3861E-1(1.0197E-3)**$^{(1)}$	8.9406E-1(9.1086E-4)$^{(5)}$
	(10,10)	**9.8218E-1(1.6495E-3)**$^{(1)}$	9.6507E-1(2.5432E-4)$^{(2)}$	8.7143E-1(4.0633E-3)$^{(5)}$	8.3318E-1(1.2091e-3)$^{(6)}$	9.2017E-2(1.3050E-3)$^{(3)}$	8.9786E-3(1.4708E-3)$^{(4)}$
	(10,5)	**9.7094E-1(2.2361E-3)**$^{(1)}$	9.1614E-1(2.8160E-3)$^{(2)}$	8.6154E-1(3.4112E-3)$^{(5)}$	8.7941E-1(7.1417E-4)$^{(4)}$	9.1323E-1(3.7025E-3)$^{(2)}$	8.1056E-1(2.1731E-3)$^{(6)}$
JY9	(5,10)	**1(0)**$^{(1)}$	9.9997E-1(1.1694E-4)$^{(2)}$	9.9799E-1(8.0809E-4)$^{(3)}$	9.5169E-1(1.3956E-2)$^{(6)}$	9.9893E-1(4.3958E-4)$^{(3)}$	9.7751E-1(7.8537E-3)$^{(5)}$
	(10,10)	**1(0)**$^{(1)}$	9.9999E-1(2.8008E-6)$^{(2)}$	9.9847E-1(6.7767E-4)$^{(3)}$	9.5247E-1(1.2076E-2)$^{(6)}$	9.9872E-1(4.9874E-4)$^{(3)}$	9.7946E-1(9.2575E-3)$^{(5)}$
	(10,5)	**1(0)**$^{(1)}$	9.9995E-1(1.6680E-4)$^{(2)}$	9.9255E-1(2.1887E-3)$^{(3)}$	9.4205E-1(1.5013E-2)$^{(6)}$	9.9282E-1(2.1178E-3)$^{(3)}$	9.4852E-1(1.2511E-2)$^{(5)}$
JY10	(5,10)	9.3853E-1(4.5985E-2)$^{(3)}$	**9.9324E-1(1.4542E-2)**$^{(1)}$	9.1482E-1(7.2486E-2)$^{(4)}$	8.4226E-1(8.2853E-2)$^{(6)}$	9.5205E-1(4.8964E-2)$^{(2)}$	8.7657E-1(8.6824E-2)$^{(5)}$
	(10,10)	9.6913E-1(3.4319E-2)$^{(3)}$	9.9614E-1(1.2884E-2)$^{(1)}$	9.2927E-1(1.0833E-1)$^{(4)}$	8.3554E-1(8.2731E-2)$^{(6)}$	**9.9894E-1(1.0032E-3)**$^{(1)}$	8.6999E-1(8.8566E-2)$^{(5)}$
	(10,5)	9.6782E-1(3.5337E-2)$^{(2)}$	9.9567E-1(1.9382E-2)$^{(3)}$	8.9672E-1(6.1693E-2)$^{(4)}$	8.5666E-1(6.1780E-2)$^{(5)}$	**9.9700E-1(2.4440E-3)**$^{(1)}$	8.2538E-1(9.8427E-2)$^{(6)}$
rank		1	2	3	6	4	5

Table 6.4 Mean $\overline{IGD}$ metric values, standard deviations and individual ranks for benchmarks $JY1 - JY10$

Prob.	(n_t, τ_t)	DMOPSO	DNSGA-II	PPS	dCOEA	SPEA2	MOEA/D
JY1	(5,10)	2.4576E-2(4.3005E-3)$^{(5)}$	4.5836E-2(5.9221E-3)$^{(6)}$	9.0564E-3(2.7019E-3)$^{(3)}$	3.4680E-3(2.5642E-4)$^{(1)}$	1.5922E-2(9.6796E-4)$^{(4)}$	**3.3283E-3(2.2835E-4)**$^{(1)}$
	(10,10)	9.9614E-3(1.0954E-3)$^{(4)}$	1.6546E-2(1.9206E-3)$^{(5)}$	7.6616E-3(5.2732E-3)$^{(3)}$	3.6956E-3(2.7778E-4)$^{(2)}$	1.5606E-2(1.2448E-3)$^{(5)}$	**3.4394E-3(2.9096E-4)**$^{(1)}$
	(10,5)	1.5794E-2(1.8954E-3)$^{(3)}$	4.0298E-2(7.0986E-3)$^{(5)}$	2.2916E-2(2.0018E-3)$^{(4)}$	**4.8348E-3(4.9517E-4)**$^{(1)}$	4.0683E-2(3.7164E-3)$^{(5)}$	8.8256E-3(5.8027E-4)$^{(2)}$
JY2	(5,10)	2.9474E-2(5.0288E-3)$^{(5)}$	4.8242E-2(6.3309E-3)$^{(6)}$	9.5494E-3(2.1188E-3)$^{(3)}$	**2.9893E-3(3.1355E-4)**$^{(1)}$	1.6285E-2(7.9016E-4)$^{(4)}$	4.8691E-3(1.7406E-4)$^{(2)}$
	(10,10)	1.0212E-2(1.3445E-3)$^{(4)}$	1.5445E-2(2.2259E-3)$^{(5)}$	7.4558E-3(6.2034E-3)$^{(3)}$	**3.0260E-3(3.4433E-4)**$^{(1)}$	1.6425E-2(1.2199E-3)$^{(6)}$	4.8188E-3(9.3727E-5)$^{(2)}$
	(10,5)	1.5611E-2(1.4170E-3)$^{(3)}$	4.2233E-2(8.0493E-3)$^{(5)}$	1.9521E-2(1.1678E-2)$^{(4)}$	**4.3761E-3(4.2792E-4)**$^{(1)}$	4.4076E-2(3.2921E-3)$^{(6)}$	8.8975E-3(4.8676E-4)$^{(2)}$
JY3	(5,10)	1.4818E-1(7.8041E-2)$^{(6)}$	**5.3791E-2(4.9115E-3)**$^{(1)}$	6.1207E-2(2.4299E-2)$^{(2)}$	6.1284E-2(7.8412E-3)$^{(2)}$	1.2502E-1(6.6890E-3)$^{(5)}$	6.2494E-2(4.5584E-3)$^{(4)}$
	(10,10)	1.7385E-1(8.7424E-2)$^{(6)}$	**5.2482E-2(8.9614E-3)**$^{(1)}$	7.1139E-2(1.9809E-2)$^{(4)}$	6.1889E-2(7.0232E-3)$^{(2)}$	1.2060E-1(6.6211E-3)$^{(5)}$	6.4126E-2(7.3189E-3)$^{(3)}$
	(10,5)	1.8013E-1(6.8710E-2)$^{(5)}$	**6.0098E-2(1.4420E-2)**$^{(1)}$	8.3533E-2(3.4515E-2)$^{(3)}$	6.7117E-2(1.1761E-2)$^{(2)}$	2.1226E-1(1.4031E-2)$^{(6)}$	9.8946E-2(8.8529E-3)$^{(4)}$
JY4	(5,10)	1.4186E-1(1.9564E-2)$^{(5)}$	1.3491E-1(2.1409E-2)$^{(5)}$	3.4723E-2(4.8741E-3)$^{(3)}$	2.0223E-2(1.2398E-2)$^{(2)}$	7.0597E-2(5.9566E-3)$^{(4)}$	**1.2860E-2(1.0222E-3)**$^{(1)}$
	(10,10)	6.1184E-2(1.6903E-2)$^{(5)}$	5.8454E-2(7.1745E-3)$^{(4)}$	2.6020E-2(1.5289E-2)$^{(3)}$	1.6022E-2(5.3674E-3)$^{(2)}$	9.1726E-2(7.6797E-3)$^{(6)}$	**1.2637E-2(1.0568E-3)**$^{(1)}$
	(10,5)	6.3319E-2(8.6966E-3)$^{(4)}$	1.2068E-1(1.8828E-2)$^{(6)}$	4.5829E-2(2.6106E-2)$^{(3)}$	2.4123E-2(1.5501E-2)$^{(1)}$	1.1440E-1(7.3326E-3)$^{(5)}$	**2.3951E-2(1.2593E-3)**$^{(1)}$
JY5	(5,10)	7.8916E-3(5.5703E-3)$^{(5)}$	**5.8490E-4(1.6311E-5)**$^{(1)}$	5.2662E-3(4.0159E-3)$^{(4)}$	2.5758E-3(1.8189E-4)$^{(2)}$	1.6455E-2(4.3054E-4)$^{(6)}$	4.8963E-3(4.9188E-4)$^{(3)}$
	(10,10)	5.0419E-3(1.5179E-3)$^{(5)}$	**5.7646E-4(1.5974E-5)**$^{(1)}$	4.8047E-3(3.0641E-3)$^{(4)}$	2.5858E-3(2.1189E-4)$^{(2)}$	9.2492E-3(5.2017E-4)$^{(6)}$	4.6571E-3(3.5615E-4)$^{(3)}$
	(10,5)	8.5293E-3(2.0189E-3)$^{(4)}$	**6.7683E-4(3.0593E-5)**$^{(1)}$	1.4583E-2(1.5940E-3)$^{(5)}$	3.7948E-3(4.0851E-4)$^{(2)}$	2.1131E-2(9.6332E-4)$^{(6)}$	4.8352E-3(4.1812E-4)$^{(3)}$
JY6	(5,10)	4.5312E-1(4.5705E-2)$^{(6)}$	3.7983E-1(3.8201E-2)$^{(5)}$	2.3759E-1(1.2809E-2)$^{(4)}$	**3.6824E-2(3.8123E-3)**$^{(1)}$	2.0250E-1(1.1152E-2)$^{(3)}$	4.3397E-2(1.8170E-3)$^{(2)}$
	(10,10)	2.9461E-1(2.4649E-2)$^{(2)}$	2.5650E-1(2.7947E-2)$^{(5)}$	2.3524E-1(3.2155E-2)$^{(4)}$	**2.9860E-2(3.7114E-3)**$^{(1)}$	2.0313E-1(1.0359E-2)$^{(3)}$	4.3589E-2(4.3421E-3)$^{(2)}$
	(10,5)	3.7529E-1(3.1560E-2)$^{(4)}$	3.8736E-1(4.8610E-2)$^{(5)}$	2.8746E-1(4.7924E-2)$^{(3)}$	**4.8382E-2(6.9547E-3)**$^{(1)}$	3.9579E-1(1.4055E-2)$^{(5)}$	1.0333E-1(8.8308E-3)$^{(2)}$
JY7	(5,10)	7.5488E-1(3.0699E-2)$^{(6)}$	3.6727E-1(2.4223E-2)$^{(4)}$	4.4931E-1(2.1406E-2)$^{(5)}$	**1.9092E-2(5.7554E-6)**$^{(1)}$	1.2992E-1(8.9316E-3)$^{(2)}$	3.1154E-1(1.8810E-2)$^{(3)}$
	(10,10)	7.5412E-1(3.0362E-2)$^{(6)}$	3.1583E-1(3.0571E-2)$^{(3)}$	4.4237E-1(1.9914E-2)$^{(5)}$	**1.6815E-2(2.9680E-3)**$^{(1)}$	1.3440E-1(9.7957E-3)$^{(2)}$	3.0834E-1(2.5724E-2)$^{(3)}$
	(10,5)	7.8714E-1(3.5140E-2)$^{(6)}$	4.3981E-1(3.6027E-2)$^{(4)}$	4.9185E-1(3.4660E-2)$^{(5)}$	**3.8650E-2(1.0235E-2)**$^{(1)}$	3.0073E-1(1.8860E-2)$^{(2)}$	3.6521E-1(2.3675E-2)$^{(3)}$
JY8	(5,10)	1.2233E-2(1.7493E-3)$^{(5)}$	**2.0951E-3(4.2461E-4)**$^{(1)}$	8.1733E-3(3.2631E-3)$^{(2)}$	8.6593E-3(5.8224E-4)$^{(2)}$	2.3861E-2(1.0197E-3)$^{(6)}$	8.9406E-3(9.1086E-4)$^{(2)}$
	(10,10)	1.2218E-2(1.6495E-3)$^{(5)}$	**1.6507E-3(2.5432E-4)**$^{(1)}$	6.7143E-3(4.0633E-3)$^{(3)}$	6.3318E-3(1.2091E-3)$^{(2)}$	2.2017E-2(1.3050E-3)$^{(6)}$	7.9786E-3(1.4708E-3)$^{(4)}$
	(10,5)	1.7094E-2(2.2361E-3)$^{(4)}$	**3.1614E-3(2.8160E-3)**$^{(1)}$	1.6154E-2(3.4112E-3)$^{(4)}$	9.7941E-3(7.1417E-4)$^{(2)}$	5.9323E-2(3.7025E-3)$^{(6)}$	1.1056E-2(2.1731E-3)$^{(3)}$
JY9	(5,10)	9.1544E-2(9.3585E-3)$^{(5)}$	1.4108E-1(1.2678E-2)$^{(6)}$	7.2722E-2(6.3619E-3)$^{(4)}$	3.7858E-2(1.0687E-3)$^{(3)}$	2.0666E-2(1.4999E-3)$^{(2)}$	**3.3934E-3(2.2449E-4)**$^{(1)}$
	(10,10)	4.8915E-2(1.0035E-2)$^{(5)}$	6.8419E-2(1.1182E-2)$^{(6)}$	4.3795E-2(4.3370E-3)$^{(4)}$	5.0184E-3(3.2546E-4)$^{(2)}$	2.1313E-2(1.6143E-3)$^{(3)}$	**3.3437E-3(1.7892E-4)**$^{(1)}$
	(10,5)	6.3014E-2(7.3838E-3)$^{(4)}$	9.6062E-2(9.4427E-3)$^{(6)}$	6.2169E-2(5.5242E-3)$^{(4)}$	**6.3257E-3(7.1891E-4)**$^{(1)}$	5.0164E-2(2.4089E-3)$^{(3)}$	8.8643E-3(5.9063E-4)$^{(2)}$
JY10	(5,10)	9.0868E-2(1.8480E-2)$^{(5)}$	7.6736E-2(2.2771E-2)$^{(4)}$	1.3154E-1(1.8979E-2)$^{(6)}$	**2.7285E-2(1.5762E-2)**$^{(1)}$	6.2069E-2(2.5791E-2)$^{(2)}$	6.8150E-2(4.9828E-2)$^{(2)}$
	(10,10)	6.8494E-2(2.6837E-2)$^{(3)}$	8.4256E-2(4.7116E-2)$^{(5)}$	9.8830E-2(3.3394E-2)$^{(6)}$	**2.8343E-2(1.7144E-2)**$^{(1)}$	3.0060E-2(1.0663E-2)$^{(2)}$	6.6837E-2(5.6151E-2)$^{(3)}$
	(10,5)	1.1941E-1(4.5839E-2)$^{(4)}$	1.3167E-1(5.6269E-2)$^{(5)}$	1.2362E-1(1.2736E-2)$^{(5)}$	**2.8149E-2(1.5885E-2)**$^{(1)}$	7.5354E-2(3.1506E-2)$^{(2)}$	8.4767E-2(5.5052E-2)$^{(3)}$
rank		6	3	4	1	5	2

Table 6.5 Mean $R(IGD)$ metric values, standard deviations and individual ranks for benchmarks $JY1-JY10$

Prob.	(n_t,τ_t)	DMOPSO	DNSGA-II	PPS	dCOEA	SPEA2	MOEA/D
JY1	(5,10)	2.2615E-2(6.3383E-3)$^{(5)}$	3.2366E-2(4.0559E-3)$^{(6)}$	7.9716E-3(3.6594E-3)$^{(3)}$	1.5553E-3(3.3643E-4)$^{(2)}$	9.7472E-3(1.3009E-3)$^{(4)}$	**1.2275E-3(1.8507E-4)**$^{(1)}$
	(10,10)	5.7792E-3(1.4972E-3)$^{(3)}$	1.4496E-2(3.5270E-3)$^{(6)}$	8.7222E-3(8.0187E-3)$^{(4)}$	1.6116E-3(3.6058E-4)$^{(2)}$	1.0752E-2(1.8823E-3)$^{(5)}$	**1.3288E-3(3.3161E-4)**$^{(1)}$
	(10,5)	9.6013E-3(2.2318E-3)$^{(3)}$	3.3601E-2(8.4655E-3)$^{(6)}$	2.4806E-2(1.8720E-2)$^{(4)}$	**2.3275E-3(5.7419E-4)**$^{(1)}$	2.9593E-2(3.6770E-3)$^{(5)}$	3.8537E-3(4.9485E-4)$^{(2)}$
JY2	(5,10)	2.7095E-2(7.3523E-3)$^{(5)}$	3.4828E-2(4.8267E-3)$^{(6)}$	8.2951E-3(2.5605E-3)$^{(3)}$	**1.2066E-3(2.7053E-4)**$^{(1)}$	1.0338E-2(1.5988E-3)$^{(4)}$	1.4403E-3(1.1767E-4)$^{(2)}$
	(10,10)	5.7417E-3(1.8539E-3)$^{(4)}$	1.4973E-2(3.0940E-3)$^{(6)}$	8.4245E-3(9.0147E-3)$^{(3)}$	**1.4200E-3(4.6466E-4)**$^{(1)}$	1.1386E-2(1.9980E-3)$^{(5)}$	1.4287E-3(8.5421E-5)$^{(2)}$
	(10,5)	1.0072E-2(1.8844E-3)$^{(4)}$	3.5567E-2(9.2729E-3)$^{(6)}$	2.0801E-3(1.5323E-2)$^{(2)}$	**2.2158E-3(4.8969E-4)**$^{(1)}$	2.9810E-2(4.4291E-3)$^{(5)}$	3.6835E-3(4.2791E-4)$^{(2)}$
JY3	(5,10)	8.9665E-2(7.0631E-2)$^{(6)}$	**3.2106E-3(3.8492E-3)**$^{(1)}$	2.3475E-2(2.0171E-2)$^{(2)}$	1.7667E-2(1.73823-2)$^{(2)}$	3.6688E-2(4.9832E-3)$^{(5)}$	2.2579E-2(1.2988E-2)$^{(2)}$
	(10,10)	1.0214E-1(6.9523E-2)$^{(6)}$	**2.1080E-3(2.5141E-3)**$^{(1)}$	3.1261E-2(1.4477E-2)$^{(4)}$	2.1629E-2(1.5351E-2)$^{(2)}$	3.8599E-2(6.4409E-3)$^{(5)}$	2.2736E-2(1.9780E-2)$^{(2)}$
	(10,5)	1.0670E-1(7.3419E-2)$^{(6)}$	**9.3054E-3(2.3429E-3)**$^{(1)}$	4.0127E-2(2.3248E-2)$^{(3)}$	2.8927E-2(2.1303E-2)$^{(2)}$	8.5434E-2(7.9086E-3)$^{(5)}$	5.4749E-2(1.1690E-2)$^{(4)}$
JY4	(5,10)	1.3523E-1(2.1105E-2)$^{(6)}$	7.9687E-2(1.5238E-2)$^{(5)}$	2.2896E-2(5.2030E-3)$^{(2)}$	3.2373E-2(4.9928E-2)$^{(3)}$	5.6955E-2(8.7403E-3)$^{(4)}$	**5.3642E-3(1.1881E-3)**$^{(1)}$
	(10,10)	6.6900E-2(2.8756E-2)$^{(6)}$	4.0311E-2(9.2533E-3)$^{(4)}$	2.4936E-2(2.2417E-2)$^{(3)}$	1.3133E-2(1.1959E-2)$^{(2)}$	5.6951E-2(7.7050E-3)$^{(5)}$	**5.0281E-3(1.3546E-3)**$^{(1)}$
	(10,5)	5.7279E-2(1.5664E-2)$^{(4)}$	7.8597E-2(1.8298E-2)$^{(6)}$	4.2132E-2(2.9441E-2)$^{(3)}$	3.1484E-2(4.7990E-2)$^{(2)}$	6.7847E-2(9.0503E-3)$^{(5)}$	**1.0339E-2(1.2577E-3)**$^{(1)}$
JY5	(5,10)	3.5353E-3(2.0233E-3)$^{(3)}$	**6.3615E-5(2.2761E-5)**$^{(1)}$	4.8849E-3(4.2601E-3)$^{(5)}$	1.1699E-3(2.4770E-4)$^{(2)}$	1.5169E-2(6.0481E-4)$^{(6)}$	4.2292E-3(3.2633E-4)$^{(4)}$
	(10,10)	2.5308E-3(1.0910E-3)$^{(3)}$	**4.8411E-5(2.2339E-5)**$^{(1)}$	4.0963E-3(3.2071E-3)$^{(5)}$	1.2138E-3(2.9585E-4)$^{(2)}$	5.4911E-3(5.4656E-4)$^{(6)}$	3.9497E-3(2.5087E-4)$^{(4)}$
	(10,5)	4.8782E-3(1.4629E-3)$^{(4)}$	**8.3150E-5(2.5970E-5)**$^{(1)}$	1.1418E-2(1.2487E-3)$^{(5)}$	1.9677E-3(5.5120E-4)$^{(2)}$	1.3018E-2(1.1218E-3)$^{(6)}$	4.1912E-3(5.4263E-4)$^{(3)}$
JY6	(5,10)	2.3263E-1(2.5679E-2)$^{(5)}$	2.4236E-1(3.1782E-2)$^{(5)}$	1.1376E-1(8.6201E-3)$^{(3)}$	**2.9592E-2(3.8997E-3)**$^{(1)}$	1.0582E-1(1.0477E-2)$^{(3)}$	2.9767E-3(3.8069E-3)$^{(1)}$
	(10,10)	1.5257E-1(2.0086E-2)$^{(6)}$	1.3778E-1(2.2134E-2)$^{(5)}$	1.0945E-1(2.9413E-2)$^{(3)}$	**2.2309E-2(4.5449E-3)**$^{(1)}$	1.0755E-1(9.8136E-3)$^{(3)}$	2.7556E-2(4.4324E-3)$^{(2)}$
	(10,5)	2.0739E-1(2.4211E-2)$^{(5)}$	2.4350E-1(4.7537E-2)$^{(6)}$	1.6667E-1(5.0356E-2)$^{(3)}$	**3.8780E-2(9.9061E-3)**$^{(1)}$	1.8844E-1(1.3488E-2)$^{(4)}$	4.3816E-2(4.4208E-3)$^{(2)}$
JY7	(5,10)	7.5110E-1(4.0029E-2)$^{(6)}$	3.0353E-1(3.3898E-2)$^{(4)}$	3.5572E-1(3.0933E-2)$^{(5)}$	**2.0455E-2(1.1561E-2)**$^{(1)}$	1.2772E-1(1.4917E-2)$^{(3)}$	1.0706E-1(1.2660E-2)$^{(2)}$
	(10,10)	7.0620E-1(3.9224E-2)$^{(6)}$	2.8812E-1(5.3380E-2)$^{(4)}$	3.3067E-1(3.1074E-2)$^{(5)}$	**1.7892E-2(5.6820E-3)**$^{(1)}$	1.1946E-1(1.5077E-2)$^{(3)}$	1.1207E-1(1.9473E-2)$^{(2)}$
	(10,5)	7.4648E-1(4.9574E-2)$^{(6)}$	4.1139E-1(3.4148E-2)$^{(5)}$	3.3996E-1(4.0839E-2)$^{(4)}$	**4.5925E-2(2.0791E-2)**$^{(1)}$	2.5676E-1(2.6204E-2)$^{(3)}$	1.1669E-1(1.2414E-2)$^{(2)}$
JY8	(5,10)	6.6964E-3(1.5910E-3)$^{(2)}$	**1.3074E-3(5.9419E-1)**$^{(1)}$	7.0432E-3(3.9987E-3)$^{(2)}$	1.0061E-2(1.3951E-3)$^{(6)}$	8.6673E-3(1.0495E-3)$^{(4)}$	9.8245E-3(2.0371E-3)$^{(5)}$
	(10,10)	5.7629E-3(1.8542E-3)$^{(2)}$	**8.8632E-4(4.4199E-4)**$^{(1)}$	5.7303E-3(3.9658E-3)$^{(2)}$	8.8843E-3(1.3185E-3)$^{(4)}$	1.0925E-2(1.8064E-3)$^{(5)}$	1.0160E-2(2.1520E-3)$^{(5)}$
	(10,5)	8.3506E-3(1.9443E-3)$^{(2)}$	**2.7611E-3(5.8350E-3)**$^{(1)}$	1.3616E-2(2.5396E-3)$^{(4)}$	9.8583E-3(1.7486E-3)$^{(3)}$	3.2476E-2(7.0159E-3)$^{(6)}$	1.8244E-2(2.7461E-3)$^{(5)}$
JY9	(5,10)	1.3915E-1(1.8556E-2)$^{(5)}$	2.1850E-1(2.0509E-2)$^{(6)}$	8.9592E-2(7.8177E-3)$^{(3)}$	1.0089E-1(3.7162E-3)$^{(4)}$	2.0488E-2(2.7729E-3)$^{(2)}$	**1.3142E-3(2.6369E-4)**$^{(1)}$
	(10,10)	4.2105E-2(8.7283E-3)$^{(4)}$	7.7891E-2(1.3282E-2)$^{(6)}$	6.8291E-2(7.9244E-3)$^{(5)}$	3.5673E-3(2.2443E-4)$^{(2)}$	2.1633E-2(3.6923E-3)$^{(3)}$	**1.3256E-3(2.5234E-4)**$^{(1)}$
	(10,5)	5.2803E-2(8.4317E-3)$^{(4)}$	1.0599E-1(1.2518E-2)$^{(5)}$	9.2289E-2(1.0223E-2)$^{(5)}$	**3.9474E-3(7.5653E-4)**$^{(1)}$	4.6159E-2(5.5109E-3)$^{(3)}$	4.0057E-3(5.3761E-4)$^{(1)}$
JY10	(5,10)	7.7704E-2(2.5122E-2)$^{(2)}$	9.9982E-2(3.4665E-2)$^{(4)}$	1.5444E-1(2.6838E-2)$^{(6)}$	**3.3228E-2(1.4127E-2)**$^{(1)}$	8.3292E-2(3.8794E-2)$^{(3)}$	1.1386E-1(7.3905E-2)$^{(5)}$
	(10,10)	6.1549E-2(2.9701E-2)$^{(3)}$	1.2052E-1(6.1734E-2)$^{(4)}$	1.1090E-1(2.4337E-2)$^{(4)}$	**3.2948E-2(1.6474E-2)**$^{(1)}$	3.8099E-2(2.1085E-2)$^{(2)}$	1.0000E-1(8.6969E-2)$^{(4)}$
	(10,5)	1.1303E-1(5.4679E-2)$^{(3)}$	1.5847E-1(6.7176E-2)$^{(5)}$	1.4157E-1(1.3747E-2)$^{(5)}$	**3.2730E-2(1.7318E-2)**$^{(1)}$	8.4470E-2(4.4207E-2)$^{(2)}$	1.2146E-1(8.1829E-2)$^{(4)}$
rank		6	4	3	1	5	2

The results of the $\overline{RMS}$ metric obtained by the algorithms are given in Table 6.3. Clearly, DMOPSO produces significantly better $\overline{RMS}$ results than the rest of the compared algorithms. DMOPSO obtains better maximum spread values in 22 out of the 30 test cases. This can be explained by the fact that, in DMOPSO, boundary solutions are more likely to be chosen as leaders because they are located at less crowded areas, and they will naturally drive more solutions towards boundary regions. Therefore, DMOPSO is expected to cover extreme or boundary regions of the PF and tends to give a better $\overline{RMS}$ value. The similar reason can be also used to explain the good performance of the second best performer, i.e., DNSGA-II, on the $\overline{RMS}$ metric. In DNSGA-II, the crowding distance technique will assign a high value to boundary solutions, so these solutions have more opportunity to survive to the next generation. It is understandable that the performances of dCOEA and MOEA/D in the $\overline{RMS}$ comparison are compromised because neither of them rewards boundary solutions.

Table 6.4 presents the $\overline{IGD}$ metric values obtained by six algorithms. The results of $\overline{IGD}$ are also almost consistent with those of $\overline{S}$ displayed in Table 6.2. This is reasonable because the $\overline{IGD}$ metric measures not only the proximity of approximate solutions but also their diversity or distribution. If the diversity performance of an algorithm is not good enough, then its $\overline{IGD}$ value will be affected. This also implies that maintaining population diversity is one of the main goals of handling dynamic environments. It can be seen from Table 6.4 that DNSGA-II, dCOEA, and MOEA/D are the three best performers in terms of the test problems considered in this work, and achieve better $\overline{IGD}$ metric values in 9, 14, and 7 out of the 30 test cases, respectively. Note that, although DNSGA-II outperforms the other algorithms for $JY3$, $JY5$ and $JY8$, it struggles to handle multi-modality (see the large values for $JY6$ and $JY7$). In addition, the mixed types of changes in $JY9$ and $JY10$ also challenge the performance of DNSGA-II, whose results are not as good as those of the others. If we take a close look at the $\overline{IGD}$ metric values provided by the six algorithms for $JY9$ and $JY10$, we can observe that, the randomness of the orders of types clearly increases the difficulty of Mixed-Type DMOPs, thus affecting the performance of all the algorithms. On the other hand, the $\overline{IGD}$ metric values of PPS on the 30 test cases are not as good as expected in its original work. One possible explanation for its poor performance is that, the population prediction strategy works well only under the assumption that the PFs (PSs) of consecutive MOPs are similar to each other in most cases. However, the PFs (PSs) of consecutive MOPs considered in this work are less similar, which causes difficulties for PPS to predict exact or promising population locations in new environments.

The results in Table 6.4 also show the effects of the severity of change n_t and the frequency of change τ_t. In most cases, compared with severe landscape changes ($n_t = 5$), moderate changes ($n_t = 10$) are less challenging for the algorithms, leading to roughly better $\overline{IGD}$ metric values for the six algorithms. The influence of the frequency of change can be observed from the different $\overline{IGD}$ metric values of n_t and τ_t combinations between

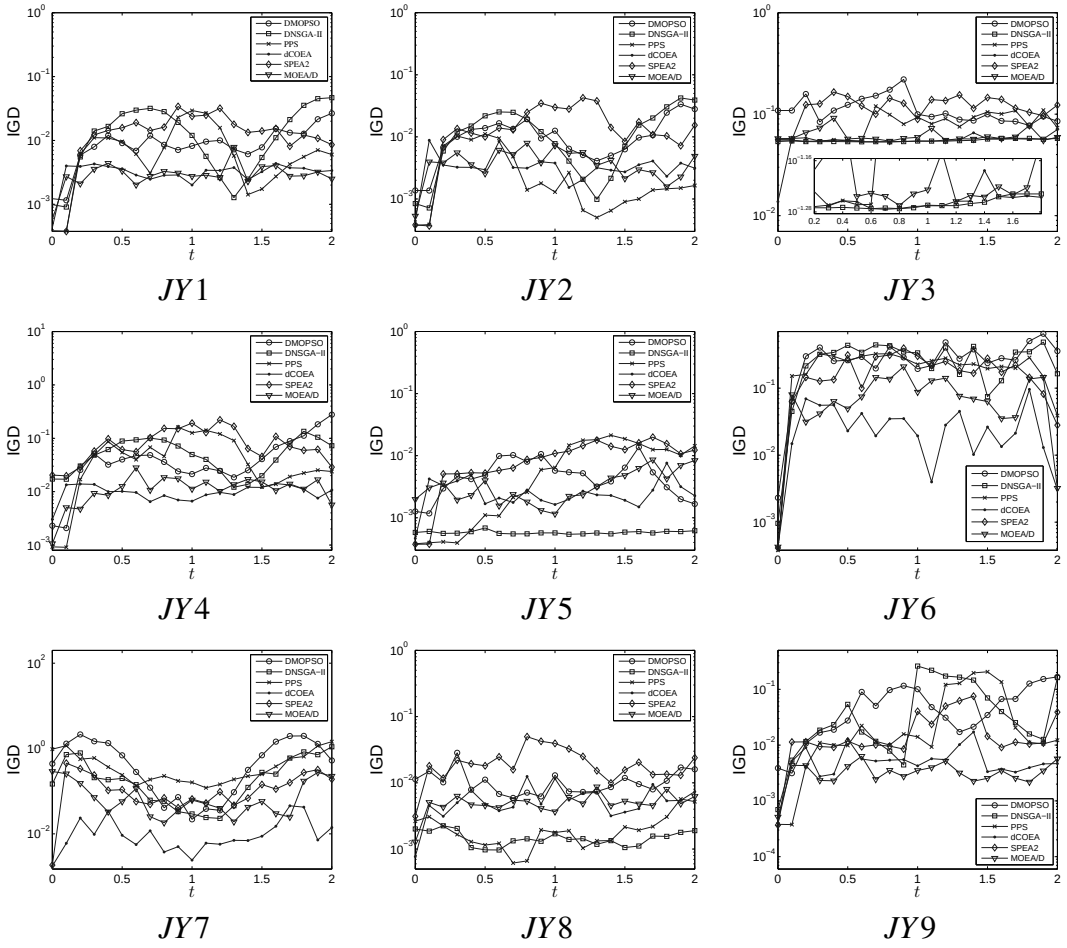


Fig. 6.10 The tracking of the IGD values obtained by six algorithms for time t from 0 to 2.

(10,5) and (10,10). The majority of the algorithms in fast landscape changes ($\tau_t = 5$) cannot provide as good results as they do in less frequent environmental changes ($\tau_t = 10$). Compared with the effect of severity n_t, the effect of frequency τ_t is much more obvious and challenging and the $\overline{IGD}$ metric values obtained by the algorithms become much worse when the environment changes faster, i.e., the frequency of change τ_t becomes smaller. Notably, MOEA/D commonly works well in each combination of n_t and τ_t for all the test problems except $JY8$. This good performance can be attributed to the fast convergence of MOEA/D, which helps the algorithm to track the changing PFs as quickly as possible.

By comparing the algorithms' performance for $JY1$ and $JY6$ on $\overline{IGD}$, we can observe that, while these algorithms can easily handle the static unimodality of $JY1$, they face difficulties in handling the time-varying multimodality of $JY6$ as the time-varying multimodality distinctly aggravates these algorithms' convergence, leading to massive challenges to their tracking ability.

The robustness performance of the six algorithms are presented in Table 6.5. The $R(IGD)$ values of these algorithms in this table are roughly consistent with their $\overline{IGD}$ metric values shown in Table 6.4. dCOEA and MOEA/D are the two most robust MOEAs on the 30 test cases, which further illustrates their ability to deal with various dynamisms. It is interesting that although PPS never wins on the 30 test cases, its average rank value is better than that of DNSGA-II, which wins over other algorithms in 9 out of the 30 cases in terms of the $R(IGD)$ metric. This means PPS is more robust than DNSGA-II in all the test cases. If we compare the average ranks in both Table 6.4 and Table 6.5, it is also easy to see that, in the event of an environmental change, PPS is on average expected to be stabler than DNSGA-II although the $\overline{IGD}$ metric values of the former are not as good as those of the latter.

In addition to showing some consistency with the $\overline{IGD}$ metric, the $R(IGD)$ metric values in Table 6.5 can also help to further compare the performance of the tested algorithms, especially when two or more algorithms achieve similar $\overline{IGD}$ values on a test case. To be more specific, for $JY1$, dCOEA and MOEA/D obtain similar $\overline{IGD}$ values shown in Table 6.4 for the setting of $n_t = 5$ and $\tau_t = 10$. In this case, it is hard to identify which one is better on this test case in terms of the $\overline{IGD}$ metric. With the use of the robustness metric, nevertheless, MOEA/D is shown to be more robust than dCOEA for the test case, which can be observed from Table 6.5. In Table 6.4, PPS, dCOEA, and MOEA/D also achieve similar $\overline{IGD}$ values for the test case $JY8$ with $n_t = 5$ and $\tau_t = 10$, and the Wilcoxon rank-sum test indicates there is no statistically significant difference among these three algorithms in terms of the $\overline{IGD}$ metric. Again, the corresponding $R(IGD)$ metric values in Table 6.5 provide additional information to judge their performance, illustrating that PPS is stabler than MOEA/D, which is in turn stabler than dCOEA with regard to the IGD metric on this test case. When (n_t, τ_t) is set to (10,5), the $\overline{IGD}$ values obtained by DMOPSO and PPS are extremely close to each other for both $JY8$ and $JY9$, but their $R(IGD)$ metric values are very dissimilar, which can clearly distinguish their robustness performance and hence rank DMOPSO as the better one between these two algorithms in terms of IGD in this case. In addition, the effectiveness of the proposed robustness performance measure can be observed by further checking the $\overline{IGD}$ and $R(IGD)$ values of DNSGA-II and SPEA2 for $JY6$ with $(n_t, \tau_t) = (10, 5)$, and DMOPSO and MOEA/D for $JY10$ with the setting of $(n_t, \tau_t) = (10, 10)$.

For a nicer readability of algorithm comparisons, the final rank of each algorithm under each performance metric for every test problem is presented in Table 6.6, where ordering in the rank column is purely alphabetical. Fig. 6.10 shows the tracking of IGD values obtained by the six algorithms over the period from $t = 0$ to $t = 2$ for some test problems.

To have a better understanding of the tracking ability of these algorithms, we also plot the PF approximations of $JY2$, $JY5$ and $JY8$ over a number of time steps in Figs. 6.11,

Table 6.6 Performance rankings on four metrics for benchmarks $JY1 - JY10$

Prob.	Rank	Ranking by $\overline{S}$	Ranking by $\overline{RMS}$	Ranking by $\overline{IGD}$	Ranking by $R(IGD)$
JY1	1th	dCOEA	DMOPSO	MOEA/D dCOEA	MOEA/D
	2nd	MOEA/D	DNSGA-II	PPS	dCOEA
	3rd	PPS	PPS	DMOPSO	PPS
	4th	DNSGA-II	SPEA2	SPEA2	DMOPSO
	5th	DMOPSO	MOEA/D	DNSGA-II	SPEA2
	6th	SPEA2	dCOEA		DNSGA-II
JY2	1th	dCOEA	DMOPSO	dCOEA	dCOEA
	2nd	MOEA/D	DNSGA-II	MOEA/D	MOEA/D
	3rd	PPS	PPS	PPS	PPS
	4th	DNSGA-II	SPEA2	DMOPSO	DMOPSO
	5th	DMOPSO	MOEA/D	DNSGA-II	SPEA2
	6th	SPEA2	dCOEA	SPEA2	DNSGA-II
JY3	1th	DNSGA-II	DMOPSO	DNSGA-II	DNSGA-II
	2nd	MOEA/D	MOEA/D	dCOEA	dCOEA
	3rd	dCOEA	DNSGA-II SPEA2	PPS	MOEA/D
	4th	PPS	PPS	MOEA/D	PPS
	5th	SPEA2	dCOEA	SPEA2	SPEA2
	6th	DMOPSO		DMOPSO	DMOPSO
JY4	1th	MOEA/D	PPS SPEA2	MOEA/D	MOEA/D
	2nd	DNSGA-II	DMOPSO	dCOEA	dCOEA
	3rd	PPS	DNSGA-II	PPS	PPS
	4th	dCOEA	MOEA/D	DMOPSO	SPEA2
	5th	SPEA2	dCOEA	DNSGA-II SPEA2	DNSGA-II
	6th	DMOPSO			DMOPSO
JY5	1th	DNSGA-II	DMOPSO	DNSGA-II	DNSGA-II
	2nd	dCOEA	DNSGA-II	dCOEA	dCOEA
	3rd	MOEA/D	PPS	MOEA/D	DMOPSO
	4th	PPS	SPEA2	PPS	MOEA/D
	5th	DMOPSO	MOEA/D	DMOPSO	PPS
	6th	SPEA2	dCOEA	SPEA2	SPEA2
JY6	1th	MOEA/D	DMOPSO	dCOEA	DNSGA-II
	2nd	dCOEA	DNSGA-II	MOEA/D	dCOEA
	3rd	PPS	PPS	PPS SPEA2	DMOPSO
	4th	SPEA2	SPEA2	DMOPSO	MOEA/D
	5th	DMOPSO	MOEA/D	DNSGA-II	PPS
	6th	DNSGA-II	dCOEA		SPEA2
JY7	1th	dCOEA	DMOPSO	dCOEA	dCOEA
	2nd	MOEA/D	DNSGA-II	SPEA2	MOEA/D
	3rd	SPEA2	SPEA2	MOEA/D	SPEA2
	4th	DNSGA-II	PPS	DNSGA-II	DNSGA-II
	5th	PPS	dCOEA	PPS	PPS
	6th	DMOPSO	MOEA/D	DMOPSO	DMOPSO
JY8	1th	DNSGA-II	DMOPSO	DNSGA-II	DNSGA-II
	2nd	dCOEA	SPEA2	dCOEA	DMOPSO
	3rd	MOEA/D	DNSGA-II	PPS MOEA/D	PPS
	4th	PPS	PPS	DMOPSO	dCOEA
	5th	DMOPSO	MOEA/D	SPEA2	SPEA2 MOEA/D
	6th	SPEA2	dCOEA		
JY9	1th	MOEA/D	DMOPSO	MOEA/D	MOEA/D
	2nd	dCOEA	DNSGA-II	dCOEA	dCOEA
	3rd	PPS SPEA2	PPS	SPEA2	SPEA2
	4th	DNSGA-II	SPEA2	PPS	DMOPSO PPS
	5th	DMOPSO	MOEA/D	DMOPSO	DNSGA-II
	6th		dCOEA	DNGA-II	
JY10	1th	dCOEA	SPEA2	dCOEA	dCOEA
	2nd	PPS	DNSGA-II	SPEA2	SPEA2
	3rd	SPEA2	DMOPSO	MOEA/D	DMOPSO
	4th	DNSGA-II	PPS	DMOPSO	DNSGA-II MOEA/D
	5th	MOEA/D	MOEA/D	DNSGA-II	PPS
	6th	DMOPSO	dCOEA	PPS	

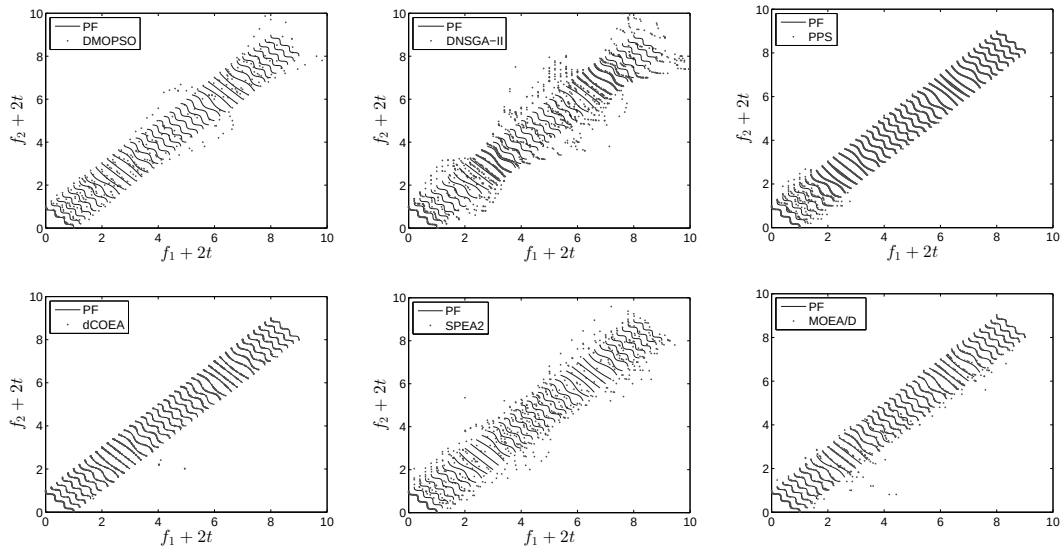


Fig. 6.11 PFs of $JY2$ with lowest $\overline{IGD}$ values obtained by six algorithms for time t from 0 to 4.

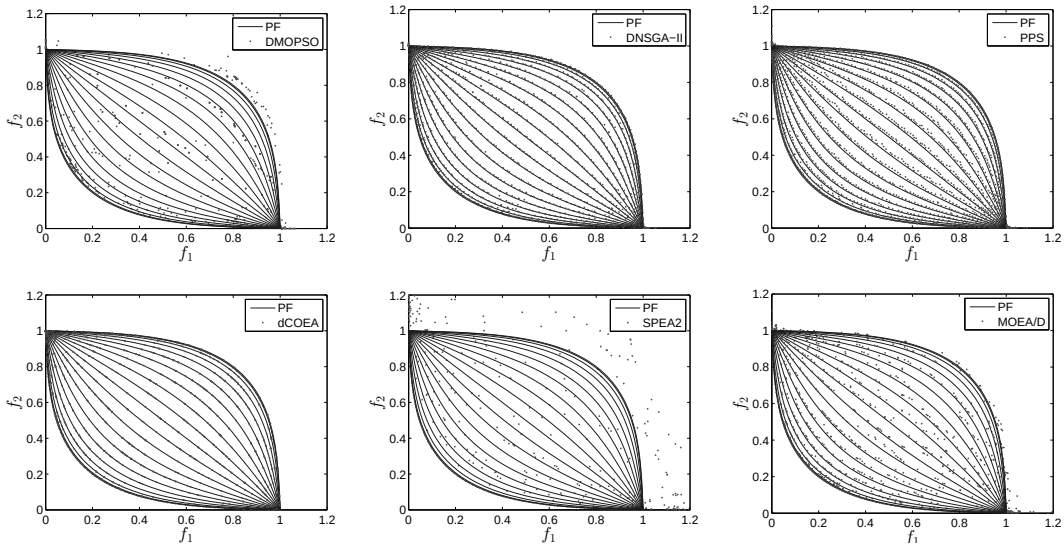


Fig. 6.12 PFs of $JY5$ with lowest $\overline{IGD}$ values obtained by six algorithms for time t from 0 to 2.

6.12 and 6.13, respectively. In Fig. 6.11, dCOEA can approximate the PFs of $JY2$ very well, and at the beginning time steps PPS does not track the changing PFs well because the quality of history information stored by PPS is not high. Figs 6.12 and 6.13 further show the good tracking performance of several algorithms (i.e., DNSGA-II, PPS and dCOEA for $JY5$ and $JY8$).

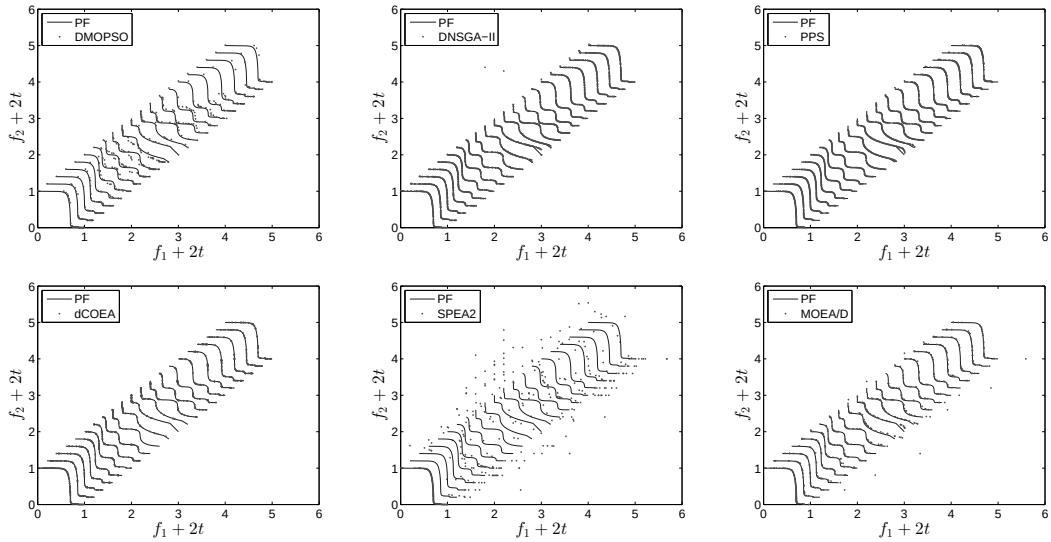


Fig. 6.13 PFs of $JY8$ with lowest $\overline{IGD}$ values obtained by six algorithms for time t from 0 to 2.

6.3.4 Influence of Variable Linkages

Since our work introduces a non-monotonic variable-linkage into $JY3$, one would wonder whether there is a clear difference between this variable-linkage and the monotonic variable-linkage existing in ZJZ.

ZJZ is originally defined as:

$$\begin{cases} min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\ f_1(\mathbf{x},t) = x_1 \\ f_2(\mathbf{x},t) = g(\mathbf{x_{II}},t)(1-(\frac{f_1(\mathbf{x},t)}{g(\mathbf{x_{II}},t)})^H) \\ g(\mathbf{x_{II}},t) = 1+\sum_{x_i \in \mathbf{x_{II}}}(x_i+G(t)-x_1^H)^2 \\ H = 1.5+G(t), G(t) = sin(0.5\pi t) \\ \mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2,...,x_n) \in [-1,2]^{n-1}. \end{cases} \tag{6.20}$$

In order to deeply compare the difference between non-monotonic and monotonic variable linkages, we replace the $g(\mathbf{x_{II}},t)$ with the one used in $JY3$. This can minimize the disturbance of the use of different objective functions.

Table 6.7 presents the comparison between different variable-linkage properties, where "ZJZ+monotonic" and "ZJZ+non-monotonic" represent the original ZJZ and ZJZ with non-monotonic variable-linkage of $JY3$, respectively, and values in bold-face highlight the best obtained results for each problem on each metric. It can be seen from the table that, for the original ZJZ, PPS obtains best values in terms on $\overline{MS}$ and $\overline{IGD}$ whereas MOEA/D performs the best on $\overline{S}$ and $R(IGD)$. However, when non-monotonic variable

Table 6.7 Influence of variable linkages on algorithms' performance

Algorithm	Metric	JY3	ZJZ+monotonic	ZJZ+non-monotonic
DMOPSO	S	1.5851E+00(5.3717E-01)	2.9111E-02(1.9070E-03)	3.3344E-02(8.0712E-03)
	MS	**1(0)**	7.0478E-01(8.3516E-03)	**6.6106E-01(5.3750E-02)**
	IGD	1.7385E-01(8.7424E-02)	3.0126E-02(5.0483E-04)	3.5934E-02(1.0169E-03)
	R(IGD)	1.0214E-01(6.9523E-02)	3.1006E-03(5.2539E-04)	5.4186E-03(5.9193E-04)
DNSGA-II	S	1.2822E-01(1.4974E-01)	1.8522E-02(1.9048E-03)	**5.3331E-03(1.5832E-03)**
	MS	8.6286E-01(2.6734E-01)	6.8429E-01(5.4611E-02)	2.7616E-01(1.9868E-02)
	IGD	**5.2482E-02(8.9614E-03)**	1.5003E-02(2.3446E-03)	3.4607E-02(1.2575E-03)
	R(IGD)	**2.1080E-03(2.5141E-03)**	9.8109E-03(1.5760E-03)	5.2191E-03(1.5803E-03)
PPS	S	1.1067E+00(6.2737E-01)	2.0776E-02(6.7486E-03)	7.6649E-02(7.7798E-03)
	MS	8.7780E-01(2.1439E-01)	**8.7880E-01(5.7405E-02)**	4.6647E-01(1.7124E-02)
	IGD	7.1139E-02(1.9809E-02)	**5.1615E-03(3.3355E-03)**	3.3944E-02(5.6485E-04)
	R(IGD)	3.1261E-02(1.4477E-02)	3.9624E-03(2.8232E-03)	**4.2723E-03(5.9101E-04)**
dCOEA	S	**4.5379E-01(3.4603E-01)**	2.3489E-02(2.4378E-03)	5.3304E-02(5.9985E-03)
	MS	3.1828E-01(1.7272E-01)	4.6602E-01(1.7903E-02)	5.7387E-01(2.4312E-02)
	IGD	6.1889E-02(7.0232E-03)	2.2929E-02(8.9555E-04)	2.2566E-02(1.7229E-03)
	R(IGD)	2.1629E-02(1.5351E-02)	6.4051E-03(9.5174E-04)	7.7948E-03(7.8941E-04)
SPEA2	S	9.1987E-01(1.4254E-01)	3.9290E-02(2.2268E-03)	4.6528E-02(4.6503E-03)
	MS	9.1387E-01(1.5497E-02)	5.9360E-01(1.6754E-02)	7.9907E-01(2.2773E-02)
	IGD	1.2060E-01(6.6211E-03)	2.2893E-02(4.8886E-04)	**2.4095E-02(6.2133E-04)**
	R(IGD)	3.8599E-02(6.4409E-03)	3.8357E-03(3.7128E-04)	6.0995E-03(6.5995E-04)
MOEA/D	S	3.4869E-01(1.1914E-01)	**1.0234E-02(8.3466E-04)**	7.1600E-03(1.7420E-03)
	MS	9.4710E-01(4.1344E-02)	8.2433E-01(1.2889E-02)	5.9811E-01(2.6839E-02)
	IGD	6.4126E-02(7.3189E-03)	7.9385E-03(3.7371E-04)	2.5316E-02(1.7813E-03)
	R(IGD)	2.2736E-02(1.9780E-02)	**2.9948E-03(4.9946E-04)**	6.5317E-03(9.6642E-04)

linkages are introduced to ZJZ, the four metric values obtained by most of the algorithms experience a significant deterioration. Taking the $\overline{S}$ metric for example, DNSGA-II, PPS and MOEA/D undergo massive changes when the non-monotonic variable-linkage is used. Likewise, their increasing $\overline{IGD}$ values suggest the non-monotonic variable-linkage makes the algorithms more difficult to converge than the monotonic variable-linkage.

The comparison between $JY3$ and ZJZ also indicates that the former has a harder convergence-resistant property than the latter, but the latter challenges algorithms' coverage performance. Besides, the non-monotonic variable-linkage of $JY3$ also causes some difficulties for dCOEA to cover the PF well, as indicated by the poor MS value of dCOEA.

6.3.5 Limitations

In the EDMO community, the mean of performance measure over particular time steps is widely adopted as it can present the performance of algorithms in a compact form. Despite of its significance and success, the mean value for experimental analysis might have limitations if the considered performance measure contains outliers at a time step [74]. For this reason, the performance measures, i.e., $\overline{S}$, $\overline{RMS}$ and $\overline{IGD}$, may be inaccurate, and other approaches, if exist, are needed to further verify conclusions based on these measures. However, besides measuring algorithms' stability over a number of changes, the proposed robustness metric can be also used as a tool to embody the impact of outliers on a performance measure, because it takes into account the variability of the performance measure values over different time steps. If there are outliers at a specific time step influencing the performance measure, the robustness value will be large. This way, robustness provides additional information to justify the performance of algorithms, increasing the reliability of performance analysis. On the other hand, no performance measure is perfect, and standard performance metrics are urgently needed in the field of EDMO.

6.4 Summary

An extensive study of the current DMOPs used to assess the performance of algorithms showed that there is a lack of standard test functions that could occur in real-world life for the EDMO domain. The commonly used DMOPs are so simple that some characteristics are excluded, such as mixed convex/concave PF components, a more powerful diversity-resistant structure in the problem, a non-monotonic correlation between variables and a mixed type of problem that can jump between different types in a regular or random pattern.

To address the above shortcomings, this chapter presents a generic scheme to generate desired benchmark functions that can compare the performance of different MOEAs. Furthermore, six representative MOEAs were tested on 10 test instances generated by the

benchmark generator and the results were evaluated by several performance metrics, including two new performance metrics proposed in this work. The comparison among these algorithms shows that the proposed test instances are effective and can help to clearly distinguish the performance of each algorithm through proper statistical testing.

The key findings from the empirical study are summarized as follows:

1. Fast converging algorithms, i.e., MOEA/D and dCOEA, can adapt quickly to changing environments, thus they may have advantages in dealing with DMOPs.

2. The lack of diversity maintenance brings about severe consequences to algorithms like DMOPSO and SPEA2 in dynamic environments. DNSGA-II, dCOEA, and MOEA/D performs very well regarding the spacing metric because they successfully maintain good population diversity during the evolution.

3. The modified maximum spread helps assess algorithms' coverage over the PF. Niche sharing (DMOPSO) and crowding distance (DNSGA-II) tend to reward boundary solutions, thus the maximum spread of algorithms with those schemes is roughly good.

4. Non-monotonic variable linkages are significantly harder than monotonic ones. Time-changing multimodality ($JY6$) is more challenging than static multimodality. Besides, the mixed type of change ($JY9$) complicates dynamic environments and randomness ($JY10$) in type of change further challenges algorithms' performance.

5. Robustness is another important performance indicator for EDMO, which helps to have a comprehensive assessment of algorithms' performance. The experimental results have shown that, when several algorithms achieve similar values on a performance metric and there is no statistically significant difference among them, the proposed robustness performance measure can provide additional information to distinguish their performance. This is really helpful for a better understanding and comparison of algorithms' performance. The significance of this new indicator can be even clearer when a stable and high-performance transient response to an environmental change is pursued.

Despite that the proposed benchmark generator can produce a series of features that are rarely tested in the literature, further research is needed regarding how to extend the generator to many-objective problems where the number of objectives is easy to scale up. Besides, the current work focuses mainly on comparing the performance of existing different metaheuristics for EDMO. It is greatly needed to design new and robust MOEAs that can handle various environmental changes. These issues will be left for further discussions in our future work.

Chapter 7

EAs for Dynamic MOPs

Many real-world optimization problems are multiobjective and dynamic in nature. Due to multiobjectivity and dynamisms, the generic optimization goal of DMOPs is to approximate a sequence of PSs/PFs that change over time. To achieve this goal, MOEAs should be able to track any environmental change accurately, react to it quickly, and provide good approximations. However, there is little work relating to algorithm design in the field of EDMO. This is not only because DMO is a very new research area but also because DMOPs are hard to handle. Despite that, new algorithms are desperately needed to move forward the research in EDMO.

Inspired by the fast convergence of steady-state MOEAs and the good diversity maintenance of generational MOEAs, this chapter is devoted to exploring the possibility of taking these two advantages to deal with DMOPs.

This chapter is organized as follows. In Section 7.1, a brief literature review about EDMO and incentives behind this work are presented. Section 7.2 describes the framework of the proposed SGEA, together with detailed descriptions of each component of the algorithm. Section 7.3 is devoted to presenting experimental settings for comparison. Section 7.4 provides experimental results and comparison on tested algorithms. A further discussion of the algorithm is offered in Section 7.5. Section 7.6 concludes the chapter with discussions on future work.

7.1 Introduction

In the past few years, there has been an increasing amount of research interest in the field of EMO as many real-world applications, like thermal scheduling [163] and circular antenna design [9], can be seen as MOPs. When an MOP involves time-dependent components, it can be regarded as a DMOP. Many real-life problems in nature are DMOPs, such as planning [20], scheduling [44, 134], and control [55, 187]. There have been a number of contributions made to several important aspects of this field, including dy-

namism classification [55, 160], test problems [11, 55, 76, 92, 93, 96], performance metrics [24, 55, 60, 61, 74, 160, 197], and algorithm design [24, 44, 55, 61, 75, 105, 196, 197]. Among these, algorithm design is the most important issue as it is the problem-solving tool for DMOPs.

Due to the presence of dynamisms, the design of an MOEA for dynamic MOPs is different from that for static MOPs. Specifically, dynamic MOEAs (DMOEAs) should not only have a fast convergence performance (which is crucial to their tracking ability), but also be able to address diversity loss whenever there is an environmental change in order to explore the new search space. Besides, if changes are not assumed to be knowable, DMOEAs should be able to detect them in order not to mislead the optimization process. This is because, when a change occurs, the previously discovered PS may not remain optimal for the new environment.

In principle, a change can be detected by re-evaluating dedicated detectors [44, 61, 182, 196, 197] or assessing algorithm behaviours [55, 125, 140]. The former is a easy-to-use mechanism and allows "robust detection" [140] if a sufficient number of detectors are used, but it may require additional cost since detectors have to be re-evaluated at every generation, and it may not be accurate when there is noise in function evaluations. The latter does not need additional function evaluations, but it may cause false positives and thus make algorithms overreacting when no change occurs. Both of them cannot guarantee that changes are detected [140].

On the other hand, whenever a change is detected, it is often inefficient to restart the optimization process from scratch, although the restart strategy may be a good choice if the environmental change is considerably severe [14]. In the literature, various approaches have been proposed to handle environmental changes, and they can be mainly categorized into diversity-based approaches and convergence-based approaches, according to their algorithm behaviors. Diversity-based approaches focus on maintaining population diversity whereas convergence-based ones aim to achieve a fast convergence performance so that algorithms' tracking ability is guaranteed. Generally, population diversity can be handled by increasing diversity using mutation of selected old solutions or random generation of some new solutions upon the detection of environmental changes [44, 61, 197], maintaining diversity throughout the optimization process [3, 5, 10], or employing multi-population schemes [61, 152]. Proper diversity is helpful for exploring promising search regions, but too much diversity may cause evolutionary stagnation [12].

Convergence-based approaches try to exploit past information for better tracking performance [14], especially when the new PS is somewhat similar to the previous one or environmental changes exhibit regular patterns. Accordingly, recording relevant past information to be reused at a later stage may be helpful for tracking the new PF as quickly as possible. The reuse of past information is closely related to the type of environmental change and hence can be helpful for different purposes [10]. If the environment changes

periodically, relevant information of the current PS can be stored in a memory and can be directly re-introduced into the evolving population when needed. This kind of strategy is often called memory-based approaches and has been extensively studied in dynamic multiobjective optimization [14, 20, 61, 71, 191]. In contrast, if the environment change follows a regular pattern, past information can be collected and used to model the movement of the changing PF/PS. As a result, the location of the new PS can be predicted, helping the evolving population quickly track the moving PF. Prediction-based approaches have received massive attention because most existing benchmark DMOPs (e.g., the FDA test suite [55]) involve predictable characteristics, and studies along this direction can be referred to [71, 105, 124, 125, 136, 182, 196, 197].

Aside from the above-mentioned approaches, some studies concentrate on finding an insensitive robust PF instead of closely tracking the moving PF [57, 97, 146]. Robustness-based approaches assume that when the environment changes, the old obtained solution can still be used in the new environment as long as its quality is acceptable [97]. However, the criterion for an acceptable optimal solution is quite problem-specific, which may hinder the wide application of these approaches.

Although a number of approaches have been proposed for solving DMOPs, the development of DMOEAs is a relatively young field and more studies are greatly needed. In this chapter, a new algorithm, called steady-state and generational EA (SGEA), is proposed for efficiently handling DMOPs. SGEA makes most of the advantages of steady-state EAs in dynamic environments [166] for environmental change detection and response. If a change is detected, SGEA reuses a portion of old solutions with good diversity and exploits information collected from both previous environments and the new environment to relocate a part of its evolving population. At the end of every generation, like conventional generational EAs [45, 202], SGEA performs environmental selection to preserve good individuals for the next generation. By mixing the steady-state and generational manners, SGEA can adapt to dynamic environments quickly whenever a change occurs, providing very promising tracking ability for DMOPs.

7.2 Proposed SGEA Method

The basic framework of the proposed SGEA is presented in Algorithm 7.1. SGEA starts with an initial population P and the initialization of an elitist population $\overline{P}$ and an archive A through environmental selection. In every generational cycle, SGEA detects possible environmental changes and evolves the population in a steady-state manner. If a change is detected by a population member, then a change response mechanism is adopted to handle the detected change. After that, genetic operation is applied to produce one offspring solution for the population member, which is then used to update the parent population P and

Algorithm 7.1: Framework of SGEA

Input: N (population size)
Output: a series of approximated PFs

```
Create an initial parent population P := {x_1, ..., x_N};
(A, P̄) := EnvironmentSelection(P);
while stopping criterion not met do
    for i ← 1 to N do
        if change detected and not responded then
            ChangeResponse();
        end
        y := GenerateOffspring(P, A);
        (P, A) := UpdatePopulation(y);
    end
    (A, P̄) := EnvironmentSelection(P ∪ P̄);
    Set P := P̄;
end
```

archive A. At the end of each generation, P and $\overline{P}$ are combined. Similar to generational EAs [45, 202] or speciation techniques used in niching [12, 107], a generational environmental selection is conducted on the combined population to preserve a population of good solutions for the next generation. This way, SGEA can be regarded as a steady-state and generational MOEA. For a better understanding of SGEA, a graphical flowchart of the algorithm is presented in Fig. 7.1. In the following subsections, the implementation of each component of SGEA will be detailed step by step.

7.2.1 Environmental Selection

The environmental selection procedure (Algorithm 7.2), which aims to preserve a fixed number of elitists from a solution set Q after every generational cycle, starts with fitness assignment. Each individual i of Q is assigned a fitness value $F(i)$, which is defined as the number of individuals that dominate [202] it, as follows:

$$F(i) = |\{j \in Q | j \preceq i, i \in Q\}|, \tag{7.1}$$

where $|\cdot|$ denotes the cardinality of a set. It should be noted that, various fine-grained methods proposed in the literature [53, 172, 202] can be used to assign fitness values for individuals. However, the fitness assignment method used in this work is relatively simple and computationally efficient. Most importantly, when an external individual e enters the set Q, the update of $F(i)$ needs only one dominance comparison between individuals e and i. The easy-to-update property of this method will be clearly embodied in the population update procedure presented in Algorithm 7.4.

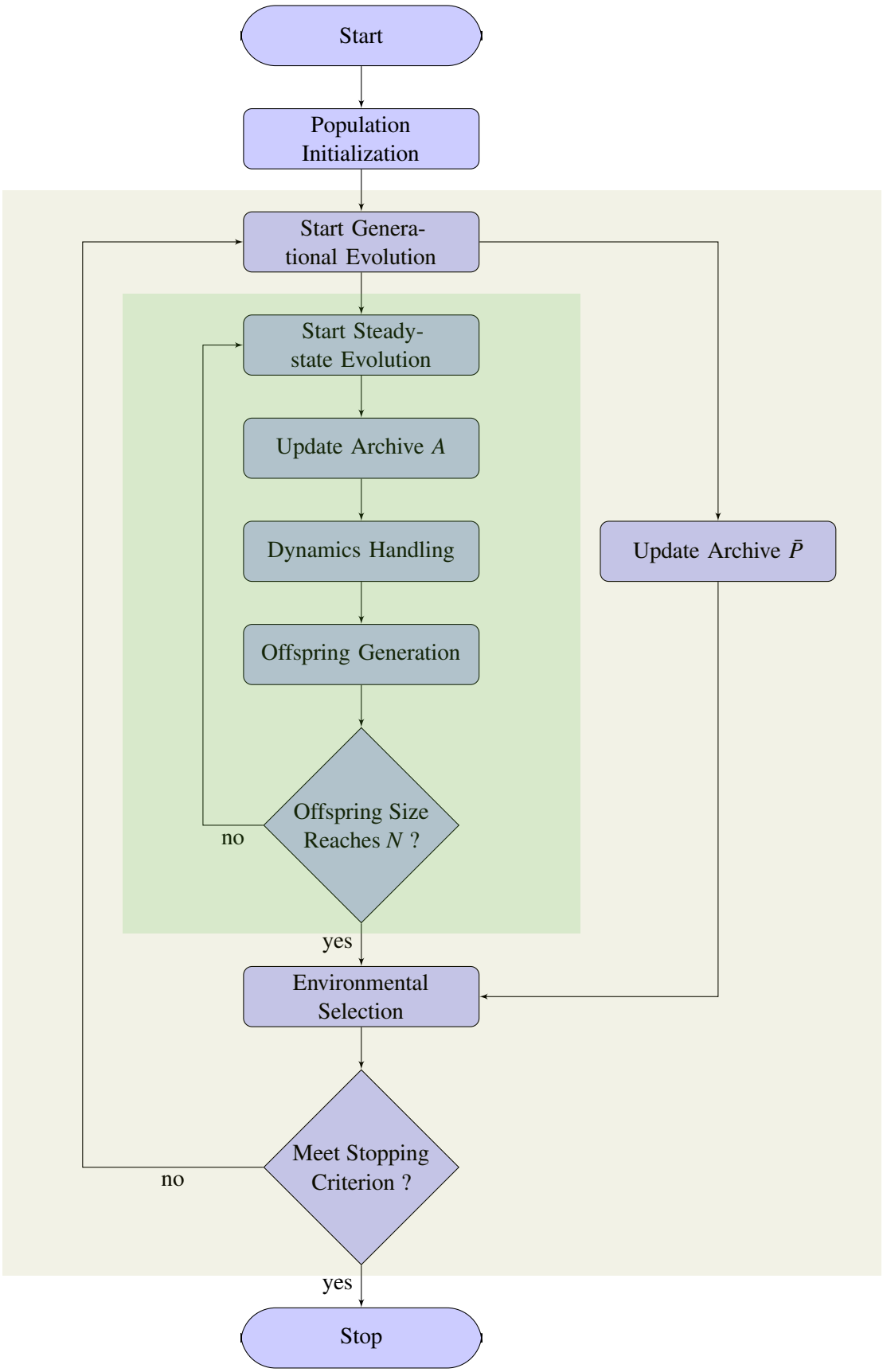


Fig. 7.1 The flowchart of SGEA with the steady-state procedure (in light green) and generational procedure (in light grey).

Algorithm 7.2: `EnvironmentSelection`(Q)

Input: Q (a set of solutions)
Output: A (archive), $\overline{P}$ (N elitists preserved)

```
Set A := ∅ and P̄ := ∅;
Assign a fitness value to each member in Q;
for i ← 1 to |Q| do
    if F(i) < 1 then
        Copy x_i from Q to A;
    end
end
if |A| < N then
    Copy the best N individuals in terms of their fitness values from Q to P̄;
end
else
    if |A| == N then
        Set P̄ := A;
    end
    else
        Prune A to a set of N individuals by any truncation operator and copy the
        truncated A to P̄;
    end
end
```

Afterwards, individuals having a fitness value of zero are identified as nondominated solutions and then copied to an archive A. If $|A|$ is smaller than the population size N, the best N individuals (including both dominated and nondominated ones) in terms of their fitness values are preserved in an elitist population $\overline{P}$. Otherwise, there can be two situations: either the number of nondominated solutions fits exactly the population size, or there are too many nondominated solutions. In the first case, all nondominated solutions are copied to $\overline{P}$. In the second case, a truncation technique is needed to reduce A to a population of N nondominated solutions such that the truncated A have the best diversity possible. In SGEA, the k-th nearest neighbour truncation technique proposed in the strength Pareto EA 2 (SPEA2) [202] is used to perform the truncation operation, although we recognize there are other options, e.g., the farthest first method [26, 27], which can also serve this purpose. After that, solutions in the truncated A are copied to $\overline{P}$.

Note that, like classic generational MOEAs, such as the nondominated sorting genetic algorithm II (NSGA-II) [45] and SPEA2 [202], SGEA performs environmental selection at the end of each generation. Thus, SGEA can be generally categorized into generational MOEAs.

Algorithm 7.3: `GenerateOffspring`(P,A)

Input: P (parent population), A (archive population)
Output: y (offspring solution)

```
1 if rnd < 0.5 then
2 |   Perform binary tournament selection on P to select two distinct individuals as
  |   the mating parents;
3 end
4 else
5 |   Randomly pick an individual from A and perform binary tournament selection
  |   on P to select another distinct individual as the mating parents;
6 end
7 Apply genetic operators to generate a new solution y;
```

7.2.2 Mating Selection and Genetic Operators

Mating selection is an important operation before the production of new offspring (line 8 of Algorithm 7.1). In this work, mating parents can be selected either from the parent population P or the archive population A. The benefit of such a mating selection method has been extensively investigated on static MOPs in a number of studies [109, 128, 162, 203]. While selecting mating parents from P can maintain good population diversity, selecting parents from A can significantly improve the convergence speed of the population, which is considerably desirable in fast-changing environments. If a mating parent is to be selected from P, SGEA performs a binary tournament selection according to individuals' fitness values. If not, the mating parent can be randomly selected from the archive population A.

Following the mating selection, genetic operators are applied on the mating parents to generate a new offspring solution. In SGEA, the simulation binary crossover and polynomial mutation are chosen as the recombination and mutation operators, respectively. The reproduction procedure is presented in Algorithm 7.3.

7.2.3 Population Update

In SGEA, population update (line 9 of Algorithm 7.1) is conducted on both the parent population P and archive population A, which is detailed in Algorithm 7.4. The update operation on P is in fact replacing the worst solution of P with the newly generated solution y while the update on A is using y to update the archived nondominated set. First, if y is not a duplicate solution, it will be compared with each member x_i of P for the dominance relation (lines 6 to 8 of Algorithm 7.4). If y dominates x_i ($y \preceq x_i$), the fitness value of x_i is increased by one. If y is dominated by x_i ($x_i \preceq y$), the fitness value of y is increased by one. Then, the worst individual in P with the highest fitness value is identified, and if

Algorithm 7.4: `UpdatePopulation`(y)

Input: y (offspring solution)
Output: P (updated parent population), A (updated archive population)

1 Set the fitness value of y as zero: $F(y) \leftarrow 0$;
2 **for** $i \leftarrow 1$ **to** $|P|$ **do**
3 **if** $y == x_i$ **then**
4 Return;
5 **end**
6 **if** $y \prec x_i$ **then**
7 Add one to the fitness value of x_i: $F(i) := F(i) + 1$;
8 **end**
9 **if** $y \succ x_i$ **then**
10 Add one to the fitness value of y: $F(y) := F(y) + 1$;
11 **end**
12 **end**
13 Compute the individual in P having the highest fitness value: $\hat{i} := i : \text{argmax}_{\{1 \leq i \leq |P|\}} F(i)$;
14 **if** $F(y) \leq F(\hat{i})$ **then**
15 Set $x_{\hat{i}} := y$ and $F(\hat{i}) := F(y)$;
16 **if** $F(y) < 1$ **then**
17 Remove all solutions in A that are dominated by y, and add y to A if A is not full;
18 **end**
19 **end**

there are two or more such individuals, a random one is selected. If y is not worse than the identified individual $x_{\hat{i}}$ in terms of the fitness value, the solution replacement takes place, as shown in line 15 of Algorithm 7.4. Besides, if y is not dominated by any member in P (which means its fitness value is zero), it should be further considered to update the archive population A if A is not full. This means, the archive update occurs only when y successfully enters the parent population. It can be observed that, the fitness assignment method used here is easy to update an individual's fitness value, which helps SGEA conduct solution replacement in the parent population and archive update in an efficient manner.

7.2.4 Dynamism Handling

This section discusses two main aspects of dynamism handling. One is change detection, a step to detect whether a change has occurs during the evolutionary process. The other is known as change response or change reaction, which takes actions to quickly react to environmental changes so that the population adapts to new environments rapidly.

7.2.4.1 Change Detection

Change detection can be performed by either re-evaluating a portion of existing solutions [44, 61, 182, 196, 197] or assessing some statistical information of some selected population members [55, 125, 140]. Since both methods choose a small proportion of population members as detectors, detection may fail if changes occur on non-detectors. On the contrary, it will be computationally expensive if the whole population members are chosen as detectors. Therefore, a good detection method should strike a balance between the detection ability and efficiency.

The proposed algorithm detects changes in a steady-state manner, as shown in line 6 of Algorithm 7.1. In every generation, population members (in random order) are checked one by one for discrepancy between their previous objective values and re-evaluated ones. If a discrepancy exists in a population member, we assume a change is successfully detected and there is no need to do further checks for the rest of population members. When a change is detected, SGEA immediately reacts to it in a steady-state manner. The detection method is beneficial to prompt and steady change reaction at the cost of high complexity. For efficiency, the number of individuals re-evaluated for change detection is restricted to a small percentage of the population size. It is worth noting that, re-evaluation based change detection methods assume that there is no noise in function evaluations, i.e., they are not robust. Thus, the proposed method may not be suitable for detecting changes in noisy environments.

7.2.4.2 Change Response

If a change is successfully detected, some actions should be taken to react to the environmental change. A good change response mechanism must be able to maintain a good level of population diversity and relocate the population in promising areas that are close to the new PS. Simply discarding old solutions and randomly reinitializing the population is beneficial to population diversity but may be time-consuming for algorithms to converge. Likewise, fully reusing old solutions for the new environment might be misleading if the landscapes of two consecutive changes are significantly different. Also, this may cause the loss of population diversity. As a consequence, algorithms may get trapped into local minima or cannot find all PF regions for the new environment. For these reasons, in this work the population for the new environment consists of half of old solutions and half of reinitialized solutions. The half old solutions are selected by the farthest first selection method [26, 164], which was originally proposed to reduce an approximation set to the maximum allowable size. The farthest first selection method has been reported to provide better approximation than NSGA-II's crowding distance [45] for unconstrained and constrained static MOPs [26, 27]. This method selects half of old solutions that maximize the diversity in the objective space (line 3 of Algorithm 7.5). The other half reinitialized solu-

Algorithm 7.5: `ChangeResponse()`

Input: y (offspring solution)
Output: P (parent population), A (archive population)

```
R := farthest_first_selection(P) [27];
Compute the centroid C_t of A at time step t using Eq. (7.2);
Set A := ∅;
Re-evaluate solutions in R and copy nondominated solutions of R to A;
Compute search direction D using Eq. (7.4);
for each x_t ∈ P \ R do
    Reinitialize x_t using Eq. (7.5) and re-evaluate the new solution x_{t+1};
    Remove all solutions in A that are dominated by x_{t+1}, and add x_{t+1} to A;
end
Set C_{t-1} := C_t;
```

tions in the new population are produced by a guess of the new location of the changed PS. To make a correct or at least reasonable guess, one must know two things, i.e., moving direction and movement step-size. The following paragraphs contribute to how to compute them.

Let C_t be the centroid of PS and A_t be the obtained approximation set at time step t, then C_t can be computed by:

$$C_t = \frac{1}{|A_t|} \sum_{x \in A_t} x. \tag{7.2}$$

The movement step-size S_t to the new location of the changed PS at time step $t+1$ can be estimated by:

$$S_t = \|C_t - C_{t-1}\|, \tag{7.3}$$

where S_t is actually the Euclidean distance between centroids C_t and C_{t-1}.

The moving direction should be carefully elaborated to guide the population toward promising search regions. Otherwise, a completely wrong guess of the moving direction will mislead the population and make it hard to converge. Bearing this in mind, we make use of half of the old solution set R preselected by the farthest first selection [26] to compute the moving direction. First, The solutions in R are re-evaluated, and nondominated solutions are saved in the pre-emptied archive A. Then, the moving direction can be calculated by:

$$D = \frac{C_A - C_R}{\|C_A - C_R\|}, \tag{7.4}$$

where C_A and C_R are centroids of A and R in the decision space, respectively.

Having obtained the moving direction and movement step-size, the other half population can be easily reinitialized. For each member x_t in $P \setminus R$, its new location in the

decision space is generated as follows:

$$x_{t+1} = x_t + S_t D + \varepsilon_t, \tag{7.5}$$

where $\varepsilon_t \sim N(0, I\delta_t)$ is a Gaussian noise, added to increase the probability of the reinitialized population to cover the PS in the new environment. I is an identity matrix and δ_t is the standard deviation in the Gaussian distribution. δ_t is defined by:

$$\delta_t = \frac{S_t}{2\sqrt{n}}, \tag{7.6}$$

where S_t is the step-size defined in Eq. (7.3), and n is the number of decision variables.

The overall change response procedure is presented in Algorithm 7.5. It is worth noting that when the first environment change occurs, the computation of C_{t-1} is not applicable. In this situation, random reinitialization is employed for the generation of solutions in $P \setminus R$. As long as the centroids of the approximation sets of two consecutive environments are available, the above reinitialization method can be adopted.

It should be mentioned that, the proposed reinitialization method is somewhat predictive but in some sense beyond prediction. Prediction approaches usually collect only history information to predict future events. However, our method exploits both the information of previous environments and that of the new environment to reinitialize a portion of solutions, which we would like to call "guided" solutions because their relocation is guided by an estimate of the performance of the reused old solutions in the new environment. Therefore, this method may be helpful for quickly tracking the changing environment if the estimate of the new environment is reliable. It is worth mentioning that, the guided reinitialization method implicitly assumes that a change does not affect too much the relative positions between solutions in the PS. It may fail in case of a notable violation of the assumption. In this situation, the proposed method may need to work with other population reinitialization techniques in order to produce good tracking performance.

7.2.5 Computational Complexity of One Generation of SGEA

In the for loop (lines 5 to 9 in Algorithm 7.1) of each generation, computational resources are mainly consumed by the offspring generation, population update and environmental selection procedures, and other procedures need less computational cost. The generation of an offspring solution (line 8 of Algorithm 7.1) requires $O(M)$ computations, where M is the number of objectives. The population update procedure (line 9 of Algorithm 7.1) takes $O(MN)$, where N is the population size. Thus, the whole steady-state evolution part takes $O(MN^2)$ computations. The environmental selection procedure (line 11 of Algorithm 7.1) spends $O(MN^2)$ computations on fitness assignment and on average $O(N^2 \log N)$

computations [202] on elitist preservation. Therefore, the overall computational complexity of SGEA for one generational cycle is $O(MN^2)$ or $O(N^2 \log N)$, whichever is larger. It should be noted that, in fast-changing environments, the run-time complexity of environmental selection might rarely reach $O(N^2 \log N)$ as individuals usually are unlikely well-converged (obtaining excessive nondominated solutions) within very limited response time.

7.3 Experimental Design

7.3.1 Test Problems

Twenty-one test problems, including five FDA [55] problems, three dMOP [61] problems, six ZJZ problems (F5-F10) [196], and seven UDF [11] problems, are used to assess the proposed algorithm in comparison with other algorithms. The time instance t involved in these problems is defined as $t = \frac{1}{n_t}\lfloor \frac{\tau}{\tau_t} \rfloor$ (where n_t, τ_t, and τ represent the severity of change, the frequency of change, and the iteration counter, respectively). Note that, most of the test problems have periodical changes.

7.3.2 Compared Algorithms

Four popular DMOEAs are used for comparison in our empirical studies. They are the MOEA based on decomposition (MOEA/D) [188], dynamic version of NSGA-II (DNSGA-II) [44], dCOEA [61], and PPS [196], representing different classes of metaheuristics. The following gives a brief description of each compared algorithm.

1. MOEA/D: as a representative of decomposition-based algorithms, MOEA/D [188] converts a mutiobjective problem by aggregation functions into a number of single-objective subproblems and optimizes them simultaneously. MOEA/D maintains population diversity by the diversity of subproblems, and a fast convergence can be achieved by defining a neighbourhood for each subproblem and performing mating selection and solution update within this neighbourhood. Due to these features, MOEA/D has gained increasing popularity in recent years and has become a benchmark algorithm in static multiobjective optimization. In this work, the modified version of the weighted Tchebycheff approach used in [185] is adopted as the aggregation function for MOEA/D because it has been recently proved to provide better distribution than its original version. Also, a limited number n_r of solutions will be replaced by any new solution, as suggested in [120].

2. DNSGA-II: it is a dynamic version of the popular NSGA-II algorithm [45], which is a representative of Pareto-dominance based MOEAs. To make it suitable for han-

dling dynamic optimization problems, Deb et al. [44] adapted NSGA-II by replacing some population members with either randomly created solutions or mutated solutions of existing solutions if a change occurs. While the former may perform better in environments with severe changes, the latter may work well on DMOPs with moderate changes. In our experiment, the latter method is adopted as it shows slightly better performance than the former in the study of [44].

3. dCOEA: it hybridizes competitive and cooperative mechanisms observed in nature to solve static MOPs and to track the changing PF in a dynamic environment [61]. dCOEA uses a fixed number of archived solutions to detect changes, and if detected, its competitive mechanism will be started to assess the potential of existing information of various subpopulations. To increase diversity after a change, dCOEA also introduces stochastic solutions into the competitive pool. Besides, dCOEA uses an additional external population to store useful but outdated archived solutions, hoping to help the evolving population quickly adapt to the new environment by exploiting these pieces of history information. It has been shown that dCOEA is very promising for handling dynamic environments [61, 91].

4. PPS: it is a representative of prediction-based methods that model the movement track of the PF or PS in dynamic environments and then use this model to predict the new location of PS. In PPS [196], the PS information is divided into two parts: the population centre and manifold. Based on the archived population centres over a number of continuous time steps, PPS employs a univariate autoregression model to predict the next population centre. Likewise, previous manifolds are used to predict the next manifold. When a change occurs, the initial population for the new environment is created from the predicted centre and manifold. PPS has been proved to be very competitive for dynamic optimization when it is incorporated with an estimation of distribution algorithm [192], and it outperforms other predictive models [196].

7.3.3 Performance Metrics

In our experimental studies, we adopt some performance metrics introduced in Chapter 6, i.e., the averaged spacing (S), averaged maximum spread (MS), and averaged IGD. These indicators can help deeply investigate algorithms' performance regarding convergence, distribution, and diversity. Besides, we adopt the averaged HV difference (HVD) [197] for performance assessment. The reference point for the computation of hypervolume is $(z_1+0.5, z_2+0.5, \cdots, z_M+0.5)$, where z_j is the maximum value of the j-th objective of the true PF and M is the number of objectives.

7.3.4 Parameter Settings

The parameters of the MOEAs considered in the experiment were referenced from their original papers. Some key parameters in these algorithms were set as follows:

1. Population size: The population size (N) for all the test problems was set to 100. To make MOEA/D have 100 subproblems for three-objective FDA4 and FDA5, we first uniformly generate around 1000 weight vectors using the simplex-lattice design [188], then prune them to 100 using the farthest first method [26, 27].

2. Parameter settings for SGEA: These parameters were set to the same values in all the compared algorithms. Specifically, the crossover probability was $p_c = 1.0$ and its distribution index was $\eta_c = 20$. The mutation probability was $p_m = 1/n$ and its distribution $\eta_m = 20$. The archive size was the same as the population size.

3. Stopping criterion and the number of executions: Each algorithm terminates after a pre-specified number of generations and should cover all possible changes. To minimize the effect of static optimization, we gave 50 generations for each algorithm before the first change occurs. The total number of generations was set to $3n_t\tau_t + 50$, which ensures there are $3n_t$ changes during the evolution. Additionally, each algorithm was executed 30 independent times on each test instance.

4. The neighbourhood size and the number n_r of solutions allowed to replace in MOEA/D were set to 20 and 2, respectively.

5. For all the algorithms, the maximum 10% population members were chosen for change detection. For the steady-state MOEA/D, it used the same change detection mechanism as SGEA, and population re-evaluation for change response.

6. The number of uniformly sampled points on the true PF was set to 500 and 990 for the computation of IGD for bi- and three-objective problems, respectively.

7.4 Experimental Results

7.4.1 Results on FDA and dMOP Problems

To study the impact of change frequency on algorithms' ability in dynamic environments, the severity of change (n_t) was fixed to 10, and the frequency of change (τ_t) was set to 5, 10, and 20, respectively. The obtained average S, MS, IGD, and HVD results over a series of time windows and their standard deviation values are presented in Tables 7.1, 7.2, 7.3, and 7.4, respectively, where the best values obtained by one of five algorithms

Table 7.1 Mean and standard deviation values of S metric obtained by five algorithms

Prob.	(τ_t, n_t)	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
FDA1	(5,10)	2.5421E-2(2.5497E-3)‡	3.3966E-2(2.6330E-3)‡	6.1386E-2(1.6514E-2)‡	4.6542E-1(1.4472E-1)‡	**1.3267E-2(1.1095E-3)**
	(10,10)	1.0136E-2(7.4361E-3)‡	1.8316E-2(1.4011E-3)‡	1.7072E-2(6.5312E-3)‡	4.8939E-1(1.9408E-1)‡	**7.5411E-3(5.8178E-4)**
	(20,10)	6.7495E-3(7.3732E-4)‡	8.9615E-3(7.8094E-4)‡	5.7913E-2(1.6129E-2)‡	3.5391E-1(1.6524E-1)‡	**3.9986E-3(2.5969E-4)**
FDA2	(5,10)	**7.6448E-3(3.1834E-4)**	2.7693E-2(3.9466E-3)‡	2.4594E-2(6.0101E-3)‡	1.8142E-2(2.8950E-2)‡	9.4054E-3(1.6736E-3)
	(10,10)	**5.3715E-3(3.3796E-4)**	1.5614E-2(2.8655E-3)‡	1.7122E-2(3.9192E-3)‡	1.5625E-2(2.4152E-2)‡	6.5871E-3(8.7753E-4)
	(20,10)	5.0340E-3(1.3246E-4)†	8.0937E-3(2.0835E-3)‡	1.8392E-2(4.0463E-3)‡	1.0903E-2(4.8363E-3)‡	**4.9516E-3(4.9187E-4)**
FDA3	(5,10)	**1.7052E-2(2.3120E-3)**	3.3698E-2(1.6310E-2)†	5.2045E-2(9.7887E-3)‡	8.3517E-2(4.6837E-2)‡	3.1669E-2(4.1347E-3)
	(10,10)	**1.1167E-2(1.9011E-3)**	1.7698E-2(9.1874E-3)	1.6536E-2(4.1971E-3)	4.6011E-2(1.8288E-2)‡	2.4160E-2(1.8298E-3)
	(20,10)	**8.2268E-3(1.7859E-3)**	1.2049E-2(6.1286E-3)	9.0478E-3(2.0861E-3)	2.9416E-2(8.6135E-3)‡	2.2741E-2(9.9650E-4)
FDA4	(5,10)	1.2706E-1(5.5003E-3)‡	**5.9217E-2(4.6346E-3)**	1.0232E-1(9.7961E-3)†	1.8035E-1(3.2800E-2)‡	8.7427E-2(7.6848E-3)
	(10,10)	9.1659E-2(3.8467E-3)‡	**3.8658E-2(3.2771E-3)**	6.0989E-2(1.0643E-2)‡	1.6494E-1(2.9433E-2)‡	4.1252E-2(2.9737E-3)
	(20,10)	5.5146E-2(2.1395E-3)‡	2.7830E-2(1.5839E-3)†	4.8519E-2(2.9057E-3)‡	1.6572E-1(2.5986E-2)‡	**2.5354E-2(2.8502E-3)**
FDA5	(5,10)	1.5306E-1(5.0947E-3)‡	9.9019E-2(8.8149E-3)‡	1.4717E-1(1.1045E-2)‡	1.5505E-1(1.4762E-2)‡	**8.2228E-2(4.2364E-3)**
	(10,10)	1.1245E-1(3.9588E-3)‡	6.3211E-2(4.8740E-3)‡	1.0820E-1(8.7265E-3)‡	1.2839E-1(1.5067E-2)‡	**4.5009E-2(2.6441E-3)**
	(20,10)	8.0300E-2(2.3006E-3)‡	4.9950E-2(3.1582E-3)‡	8.6349E-2(4.1808E-3)‡	1.0497E-1(7.8394E-3)‡	**3.0379E-2(6.7640E-4)**
dMOP1	(5,10)	5.3389E-3(7.8416E-4)‡	8.4983E-2(5.2562E-3)‡	1.0375E-1(7.8713E-2)‡	4.1207E-2(1.1779E-1)‡	**3.4712E-3(5.4488E-4)**
	(10,10)	5.5311E-3(1.3101E-3)‡	1.5696E-2(9.5712E-3)‡	2.5068E-2(2.4719E-2)‡	5.6413E-2(2.0924E-1)‡	**2.7029E-3(3.0835E-4)**
	(20,10)	5.2961E-3(2.7514E-4)‡	6.3031E-3(6.6072E-4)‡	1.4722E-2(2.0239E-2)‡	2.6844E-2(8.1479E-2)‡	**2.5010E-3(2.5768E-4)**
dMOP2	(5,10)	1.6538E-2(1.7941E-3)‡	6.0455E-2(2.1579E-3)‡	2.7767E-2(4.5722E-3)‡	1.4701E-1(5.3676E-2)‡	**1.3177E-2(1.4569E-3)**
	(10,10)	1.0690E-2(5.3335E-4)‡	3.0587E-2(3.9867E-3)‡	1.1608E-2(2.7373E-3)‡	1.4459E-1(5.3516E-2)‡	**6.6710E-3(5.8584E-4)**
	(20,10)	6.2086E-3(1.9806E-4)‡	1.4253E-2(1.7038E-3)‡	6.2807E-3(1.1104E-3)‡	1.4322E-1(6.6231E-2)‡	**3.9175E-3(2.9561E-4)**
dMOP3	(5,10)	1.4393E-2(1.2499E-3)‡	3.3786E-2(5.5519E-3)‡	2.7518E-2(4.8871E-3)‡	2.7281E-2(2.2967E-2)‡	**9.5664E-3(9.9353E-4)**
	(10,10)	8.1655E-3(6.5231E-4)‡	1.5418E-2(1.0978E-3)‡	1.6453E-2(2.3904E-3)‡	1.2555E-2(2.0652E-3)‡	**5.4336E-3(6.0751E-4)**
	(20,10)	5.3930E-3(5.5912E-4)‡	7.3129E-3(3.9782E-4)‡	1.1264E-2(1.7604E-3)‡	9.9081E-3(1.4603E-3)‡	**4.2793E-3(5.3812E-4)**

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

Table 7.2 Mean and standard deviation values of MS metric obtained by five algorithms

Prob.	(τ_t, n_t)	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
FDA1	(5,10)	6.8875E-1(6.9604E-2)‡	8.6361E-1(2.5899E-2)‡	8.7571E-1(3.3122E-2)‡	8.2378E-1(2.2483E-2)‡	**9.3411E-1(3.2794E-2)**
	(10,10)	9.2689E-1(1.9129E-2)‡	8.9378E-1(2.2115E-2)‡	9.6555E-1(1.2319E-2)‡	9.2142E-1(1.6053E-2)‡	**9.7277E-1(1.0854E-2)**
	(20,10)	9.8453E-1(2.1657E-3)†	9.2981E-1(1.2003E-2)‡	9.8426E-1(4.8155E-3)†	9.6140E-1(8.4959E-3)‡	**9.8810E-1(6.2816E-3)**
FDA2	(5,10)	**9.9649E-1(4.2818E-3)**†	8.1389E-1(4.8855E-2)‡	9.0733E-1(5.3057E-2)‡	9.4951E-1(3.7796E-2)‡	9.9231E-1(5.2065E-3)
	(10,10)	**9.9730E-1(2.6637E-3)**†	8.7511E-1(2.9208E-2)‡	9.3410E-1(1.2746E-2)‡	9.6362E-1(2.5629E-2)‡	9.9308E-1(3.3464E-3)
	(20,10)	**9.9786E-1(1.9825E-3)**	9.1688E-1(3.2152E-2)‡	9.3897E-1(7.2423E-3)‡	9.7535E-1(1.8048E-2)‡	9.9342E-1(2.6409E-3)
FDA3	(5,10)	6.3387E-1(1.1045E-1)‡	5.0510E-1(4.5498E-2)‡	6.0036E-1(3.4102E-2)‡	7.3593E-1(9.3637E-2)‡	**8.8834E-1(8.9085E-2)**
	(10,10)	7.6418E-1(7.9082E-2)‡	5.7869E-1(3.6421E-2)‡	6.0893E-1(2.6990E-2)‡	8.2943E-1(8.4314E-2)‡	**9.3342E-1(7.1125E-2)**
	(20,10)	7.8775E-1(7.2659E-2)‡	6.8023E-1(4.3336E-2)‡	6.0760E-1(2.4411E-2)‡	8.8984E-1(2.1886E-2)‡	**9.4731E-1(7.2987E-2)**
FDA4	(5,10)	**9.9999E-1(3.2759E-6)**†	9.6390E-1(7.4777E-3)‡	9.9823E-1(7.5711E-4)‡	9.9999E-1(2.1721E-6)†	9.9997E-1(1.9039E-5)
	(10,10)	**1.0000E+0(7.8284E-7)**	9.7421E-1(6.0289E-3)‡	9.9903E-1(1.2185E-4)‡	9.9999E-1(8.5330E-7)	9.9995E-1(2.6230E-5)
	(20,10)	1.0000E+0(3.0455E-7)	9.8552E-1(2.3528E-3)‡	9.9904E-1(9.8111E-5)‡	**1.0000E+0(2.6739E-7)**	9.9992E-1(2.5034E-5)
FDA5	(5,10)	**9.9999E-1(2.0403E-6)**	9.3043E-1(3.7021E-2)‡	9.9758E-1(2.6961E-3)	9.9866E-1(3.2365E-3)	9.9442E-1(8.0786E-3)
	(10,10)	**1.0000E+0(4.3629E-7)**	9.5871E-1(3.5891E-2)‡	9.9781E-1(3.8432E-3)‡	9.9995E-1(1.4197E-4)	9.9949E-1(7.9814E-4)
	(20,10)	**1.0000E+0(7.6916E-8)**	9.7908E-1(1.9611E-2)‡	9.9955E-1(1.7863E-4)†	9.9999E-1(7.9466E-7)	9.9993E-1(5.9215E-5)
dMOP1	(5,10)	9.5971E-1(4.5522E-2)†	8.2629E-1(4.1500E-2)‡	9.3007E-1(6.7780E-2)‡	**9.6544E-1(3.8454E-2)**†	9.5950E-1(3.3426E-2)
	(10,10)	9.8083E-1(2.0385E-2)†	8.8318E-1(2.5097E-2)‡	9.7105E-1(3.3827E-2)‡	9.8276E-1(1.5980E-2)†	**9.8351E-1(1.3118E-2)**
	(20,10)	9.8836E-1(1.1924E-2)†	9.3962E-1(1.0940E-2)‡	9.8192E-1(1.8910E-2)†	**9.8869E-1(1.0211E-2)**‡	9.8534E-1(1.2710E-2)
dMOP2	(5,10)	7.1985E-1(9.8981E-2)‡	7.4615E-1(5.4804E-2)‡	8.5360E-1(1.3935E-2)‡	7.9673E-1(1.2783E-2)‡	**9.4952E-1(1.3091E-2)**
	(10,10)	8.8398E-1(1.0456E-2)‡	8.1368E-1(2.5334E-2)‡	9.5016E-1(1.6218E-2)‡	8.8264E-1(1.4109E-2)‡	**9.8099E-1(4.5689E-3)**
	(20,10)	9.8039E-1(3.2935E-2)‡	9.0203E-1(1.6144E-2)‡	9.7464E-1(2.6993E-3)‡	9.5552E-1(5.9188E-3)‡	**9.9251E-1(1.4628E-3)**
dMOP3	(5,10)	4.3016E-1(2.2614E-2)‡	**8.7837E-1(2.1444E-2)**	8.5479E-1(1.3831E-2)	5.0950E-1(3.1263E-2)†	4.9760E-1(2.2063E-2)
	(10,10)	5.3193E-1(2.1894E-2)†	**9.1097E-1(1.1716E-2)**	8.8793E-1(9.6772E-3)	6.3606E-1(1.8266E-2)	5.7573E-1(2.9590E-2)
	(20,10)	6.2492E-1(1.9883E-2)‡	**9.4844E-1(1.1052E-2)**	9.0666E-1(9.4326E-3)	7.7993E-1(1.9421E-2)	6.8486E-1(2.9571E-2)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

Table 7.3 Mean and standard deviation values of IGD metric obtained by five algorithms

Prob.	(τ_t, n_t)	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
FDA1	(5,10)	6.4053E-1(9.8895E-2)‡	6.3686E-2(1.1610E-2)‡	2.0885E-1(8.4104E-2)‡	3.5649E-1(4.9023E-2)‡	**3.4182E-2(8.0969E-3)**
	(10,10)	5.8213E-2(3.8909E-3)‡	4.1342E-2(6.5605E-3)‡	4.2736E-2(1.9486E-2)‡	1.2112E-1(1.1879E-2)‡	**1.4809E-2(2.0621E-3)**
	(20,10)	4.1464E-2(4.2405E-3)‡	2.3984E-2(2.2878E-3)‡	1.6218E-2(7.9450E-3)‡	4.0424E-2(2.2617E-3)‡	**7.5500E-3(1.4897E-3)**
FDA2	(5,10)	2.8517E-2(2.4351E-3)‡	7.2853E-2(3.8658E-2)‡	8.1301E-2(3.0399E-2)‡	8.4088E-2(1.3585E-2)‡	**1.5004E-2(1.6826E-3)**
	(10,10)	1.0805E-2(9.0279E-4)‡	4.7325E-2(3.3605E-2)‡	6.3561E-2(1.0647E-2)‡	3.3894E-2(8.8878E-3)‡	**9.1174E-3(6.3334E-4)**
	(20,10)	6.5124E-3(5.2611E-4)†	3.2472E-2(4.6061E-2)‡	6.2768E-2(9.0724E-3)‡	1.6459E-2(4.9937E-3)‡	**6.3268E-3(4.0710E-4)**
FDA3	(5,10)	2.6346E-1(6.0463E-2)‡	2.6371E-1(3.5505E-2)‡	4.4378E-1(1.1102E-1)‡	2.4764E-1(2.3050E-2)‡	**6.2525E-2(3.8414E-2)**
	(10,10)	1.0821E-1(3.3153E-2)‡	1.9526E-1(3.2807E-2)‡	2.1946E-1(1.8132E-2)‡	1.3090E-1(2.5891E-2)‡	**4.0371E-2(2.9061E-2)**
	(20,10)	9.0365E-2(2.8703E-3)‡	1.2625E-1(3.1398E-2)‡	1.9259E-1(2.4153E-2)‡	5.4535E-2(8.3567E-3)‡	**3.5293E-2(2.9668E-2)**
FDA4	(5,10)	1.4906E+0(1.2669E-1)‡	**1.6224E-1(6.1969E-3)**	3.0719E-1(1.9145E-2)‡	1.3602E+0(1.6118E-1)‡	4.6085E-1(6.6670E-2)
	(10,10)	7.6342E-1(4.4885E-2)‡	**1.2450E-1(4.5799E-3)**	2.1151E-1(2.0215E-2)‡	5.7713E-1(5.4877E-2)‡	1.8302E-1(6.6613E-3)
	(20,10)	2.6255E-1(1.6817E-2)‡	**1.0303E-1(1.7584E-3)**	1.7909E-1(3.0438E-3)‡	2.2277E-1(1.3352E-2)‡	1.2684E-1(1.5029E-3)
FDA5	(5,10)	1.7611E+0(1.0707E-1)‡	**4.3378E-1(4.6953E-2)**	6.5562E-1(3.1705E-2)‡	1.5704E+0(1.3189E-1)‡	5.2338E-1(3.3442E-2)
	(10,10)	1.0239E+0(5.4901E-2)‡	3.6283E-1(4.0631E-2)†	4.8031E-1(3.5207E-2)‡	8.1980E-1(6.0501E-2)‡	**3.6260E-1(8.5854E-3)**
	(20,10)	4.8890E-1(1.2544E-2)‡	3.1016E-1(2.7499E-2)†	3.7195E-1(1.2431E-2)‡	4.0732E-1(1.4768E-2)‡	**3.0953E-1(2.2283E-3)**
dMOP1	(5,10)	1.3135E-1(1.1037E-2)‡	6.9595E-2(1.4007E-2)‡	4.1528E-1(7.4997E-1)‡	1.3604E-2(9.0549E-3)‡	**1.1207E-2(8.1627E-3)**
	(10,10)	8.8338E-3(5.0638E-3)‡	3.9362E-2(6.2467E-3)‡	5.0918E-2(9.3741E-2)‡	9.3916E-3(4.3151E-3)‡	**8.2424E-3(5.3626E-3)**
	(20,10)	7.3907E-3(3.2736E-3)‡	1.8848E-2(2.3214E-3)‡	4.3965E-2(8.4779E-2)‡	7.1797E-3(2.7117E-3)‡	**6.5411E-3(3.0256E-3)**
dMOP2	(5,10)	6.8741E-1(7.5422E-2)‡	1.2043E-1(2.0546E-2)‡	1.5635E-1(1.8877E-2)‡	4.9102E-1(4.1828E-2)‡	**3.0254E-2(3.4200E-3)**
	(10,10)	1.1864E-1(9.4674E-3)‡	7.3299E-2(8.9931E-3)‡	4.2819E-1(1.7367E-2)‡	1.8898E-1(1.9146E-2)‡	**1.2148E-2(5.7205E-4)**
	(20,10)	1.5741E-1(6.7003E-4)‡	3.4622E-2(4.3234E-3)‡	2.0207E-2(2.4955E-3)‡	5.6301E-2(3.9135E-3)‡	**6.3230E-3(1.7401E-4)**
dMOP3	(5,10)	5.6244E-1(3.9864E-2)‡	**4.9556E-2(4.8079E-3)**	1.7617E-1(8.0705E-2)†	3.4211E-1(1.9264E-2)‡	1.8143E-1(9.6531E-2)
	(10,10)	2.0009E-1(1.5091E-2)‡	**2.9589E-2(2.4806E-3)**	1.1367E-1(1.2092E-2)	1.6853E-1(1.0496E-2)‡	1.3248E-1(1.3627E-2)
	(20,10)	1.0780E-1(8.5053E-3)‡	**1.6366E-2(1.7152E-3)**	8.9901E-2(6.7418E-3)†	6.2795E-2(4.3764E-3)	8.1563E-2(1.2540E-2)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

Table 7.4 Mean and standard deviation values of HVD metric obtained by five algorithms

Prob.	(τ_t, n_t)	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
FDA1	(5,10)	8.7093E-1(7.5592E-2)‡	1.2585E-1(2.4080E-2)‡	3.8772E-1(1.0116E-1)‡	7.7026E-1(9.4376E-2)‡	**8.1493E-2(2.0911E-2)**
	(10,10)	1.3610E-1(1.7463E-2)‡	8.5252E-2(2.0248E-2)‡	2.9712E-1(1.6596E-2)‡	2.8825E-1(2.9076E-2)‡	**3.8112E-2(1.4430E-2)**
	(20,10)	3.5539E-2(1.3774E-2)‡	5.4656E-2(1.6343E-2)‡	2.8401E-1(1.5220E-2)‡	1.3478E-1(9.2613E-3)‡	**2.0270E-2(1.2830E-2)**
FDA2	(5,10)	4.7185E-2(1.4726E-2)‡	1.8564E-1(6.4420E-2)‡	3.2184E-1(6.7336E-2)‡	1.3022E-1(2.5946E-2)‡	**2.5498E-2(1.3466E-2)**
	(10,10)	2.0598E-2(1.4744E-2)‡	1.2486E-1(4.6708E-2)‡	2.6663E-1(1.4716E-2)‡	6.2906E-2(1.8881E-2)‡	**1.6745E-2(1.4126E-2)**
	(20,10)	1.3369E-2(1.4735E-2)‡	8.6455E-2(7.0196E-2)‡	2.5527E-1(9.4754E-3)‡	3.2497E-2(1.4799E-2)‡	**1.2377E-2(1.4101E-2)**
FDA3	(5,10)	1.5478E+0(1.6485E-1)‡	1.4594E+0(8.5119E-2)‡	1.7549E+0(1.8461E-1)‡	1.6606E+0(7.8359E-2)‡	**9.8045E-1(1.0710E-1)**
	(10,10)	1.0970E+0(9.9053E-2)‡	1.3223E+0(7.7970E-2)‡	1.1626E+0(4.6925E-2)‡	1.1225E+0(9.3170E-2)‡	**9.2413E-1(8.2603E-2)**
	(20,10)	1.0419E+0(7.9293E-2)‡	1.1578E+0(6.6326E-2)‡	1.0365E+0(7.4288E-2)‡	9.4755E-1(2.2550E-2)‡	**9.1188E-1(8.1889E-2)**
FDA4	(5,10)	2.0595E+0(2.0121E-1)‡	**3.8011E-1(2.6939E-2)**	7.7744E-1(6.8856E-2)‡	3.9719E+0(1.6351E+0)‡	1.0379E+0(1.3381E-1)
	(10,10)	1.5893E+0(6.6551E-2)‡	**2.7081E-1(3.5707E-2)**†	4.3455E-1(7.2947E-2)‡	1.2458E+0(1.3708E-1)‡	2.7403E-1(2.4337E-2)
	(20,10)	5.4876E-1(5.7277E-2)‡	1.8048E-1(2.4395E-2)‡	3.3435E-1(8.3758E-3)‡	4.3462E-1(5.0662E-2)‡	**1.4480E-1(2.0339E-2)**
FDA5	(5,10)	6.7506E+0(1.9848E-1)‡	2.7667E+0(2.8579E-1)‡	3.8869E+0(3.1382E-1)‡	7.0876E+0(1.0641E+0)‡	**2.7020E+0(2.2339E-1)**
	(10,10)	5.4164E+0(1.6223E-1)‡	2.3772E+0(2.7316E-1)‡	2.1995E+0(3.9061E-1)‡	4.8043E+0(2.6911E-1)‡	**1.8867E+0(9.3801E-2)**
	(20,10)	2.6454E+0(1.1158E-1)‡	2.0207E+0(1.8696E-1)‡	1.0481E+0(1.1778E-1)	2.1517E+0(1.0853E-1)‡	**1.7874E+0(7.1890E-2)**
DMOP1	(5,10)	3.9375E-2(3.8881E-2)†	1.7307E-1(3.3448E-2)‡	2.8629E-1(3.6238E-1)‡	4.6453E-2(3.6865E-2)‡	**3.7523E-2(2.5376E-2)**
	(10,10)	2.2844E-2(2.0365E-2)‡	1.1236E-1(2.0863E-2)‡	9.2754E-2(1.3906E-1)‡	2.5712E-2(1.5458E-2)‡	**1.9048E-2(1.4569E-2)**
	(20,10)	1.7194E-2(1.4792E-2)†	5.6555E-2(8.1366E-3)‡	6.0241E-2(8.1387E-2)‡	**1.5920E-2(7.9781E-3)**	1.8012E-2(1.3010E-2)
DMOP2	(5,10)	8.0662E-1(1.1259E-1)‡	3.0338E-1(4.9212E-2)‡	3.9550E-1(3.9842E-2)‡	9.0438E-1(7.3270E-2)‡	**8.7174E-2(1.9234E-2)**
	(10,10)	2.9084E-1(2.5036E-2)‡	2.0782E-1(2.4976E-2)‡	1.1778E-1(4.3469E-2)‡	4.4678E-1(4.2568E-2)‡	**3.5928E-2(1.1177E-2)**
	(20,10)	4.5002E-2(1.2356E-2)‡	1.0906E-1(1.5524E-2)‡	5.6596E-2(6.2322E-3)‡	1.9824E-1(1.4847E-2)‡	**1.8517E-2(1.1142E-2)**
DMOP3	(5,10)	9.5131E-1(3.4052E-2)‡	**1.0526E-1(1.6998E-2)**	4.2264E-1(1.5786E-2)†	7.6163E-1(5.3913E-2)†	4.0715E-1(2.4743E-2)
	(10,10)	4.7415E-1(2.8497E-2)‡	**6.5770E-2(1.3652E-2)**	2.7970E-1(2.7200E-2)	4.5433E-1(2.8194E-2)‡	3.1866E-1(2.9555E-2)
	(20,10)	2.7629E-1(2.5543E-2)‡	**3.6360E-2(1.3138E-2)**	2.2118E-1(1.5326E-2)†	2.8764E-1(2.0740E-2)‡	2.1541E-1(3.0679E-2)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

are highlighted in bold face. The Wilcoxon rank-sum test [179] is carried out to indicate significance between different results at the 0.05 significance level.

It can be observed from Table 7.1 that SGEA obtains the best results on the majority of the tested FDA and dMOP instances, implying that it maintains better distribution of its approximations over changes than the other compared algorithms in most cases. However, it performs slightly worse than DNSGA-II for FDA2 and FDA3, and dCOEA for FDA4 with fast changes (i.e., $\tau_t = 5$ and 10). For all the test instances, both PPS and MOEA/D fail to show encouraging performance on the S metric, and MOEA/D seems struggling for maintaining a uniform distribution of its PF approximations for dynamic optimization, as indicated by the large S values in Table 7.1.

As shown in Table 7.2, the results on the MS metric are quite divergent. DNSGA-II and SGEA obtain a spread coverage for FDA2, FDA4, and FDA5, although DNSGA-II provides slightly better MS values than SGEA. For problems FDA1, FDA3, and dMOP2, SGEA significantly outperforms the other algorithms by a clear margin in terms of the MS metric. PPS and MOEA/D cover the PF very well for two three-objective problems, i.e., FDA4 and FDA5, and all the algorithms perform similarly on dMOP1 except dCOEA, whose MS values are not very competitive in this case. To have a better understanding of how algorithms' MS performance can be affected by different dynamisms, we discuss a little bit more on FDA3 and dMOP3. FDA3 is a problem in which environmental changes shift the PS and affect the density of points on the PF whereas dMOP3 is a problem where the population diversity can decrease dramatically. The results of MS show that, for FDA3, SGEA can maintain a good coverage of the PF when the other algorithms perform poorly. However, this is not the case for dMOP3, where only dCOEA and PPS are able to distribute their obtained solutions widely on the PF. This means that the change response mechanisms in DNSGA-II, MOEA/D, and SGEA may face big challenges when dynamisms drastically aggravate population diversity.

Since the IGD metric mainly depends on the closeness, distribution, and coverage of an approximation to the true PF, we can use IGD together with SP and MS to deeply and extensively reveal the algorithms' performance on the test instances. Table 7.3 clearly shows that, SGEA performs the best on the majority of the test instances and mainly loses on FDA4 and dMOP3, where dCOEA is the best performer, in terms of the IGD metric. Clearly, the uncompetitive distribution (i.e., slightly large S metric) and poor coverage (i.e., relatively small MS metric) of obtained approximations are the main reasons for the low performance of SGEA on FDA4 and dMOP3, respectively. However, good S and MS values do not necessarily result in satisfying IGD metric, and this can be particularly observed from the case of DNSGA-II on FDA2, suggesting that DNSGA-II converges worse than SGEA although it provides the best S and MS metrics on this problem. For PPS and MOEA/D, the IGD performance is not competitive in spite of their good spread

performance for most of the test instances, and this may be caused by their poor solution distribution, as indicated by their large S values.

Table 7.4 presents the HVD metric obtained by five algorithms on the FDA and dMOP problems. The obtained HVD values are roughly consistent with the IGD ones illustrated in Table 7.3. Clearly, SGEA is more promising than the other algorithms to solve most FDA and dMOP instances, but it is outperformed by dCOEA on FDA4 and DMOP3. Besides, the steady-state MOEA/D also shows some appealing results on FDA3 and DMOP1 when τ_t equals 20, implying that its steady-state update method may be helpful for handling slow-changing environments.

It can also be observed from the results of the three used metrics that, the frequency of change has a significant effect on algorithms' performance, and the effect decreases when environmental changes become slow. For two three-objective problems, i.e., FDA4 and FDA5, DNSGA-II and MOEA/D are most influenced by frequent changes and struggle to push their populations toward the PF, as indicated by their large IGD and HVD values in Tables 7.3 and 7.4, respectively. Overall, dCOEA and SGEA seem less sensitive to the frequency of change, as can be seen from their gradual improvement on three metrics when τ_t increases from 5 to 20. On the other hand, with the variation of frequency, there are drastic improvements on DNSGA-II, PPS, and MOEA/D in most of the test instances.

Apart from tabular presentation, we provide evolution curves of the average IGD values on the test instances in Fig. 7.2. It can be clearly seen that, compared with the other algorithms, SGEA responds to changes more stably and recovers faster for most of the test problems, thereby obtaining higher convergence performance. The only exception is dMOP3, where dCOEA performs the best, and due to the lack of population diversity (indicated by poor MS values) when a change occurs, the IGD values obtained by SGEA fluctuate widely on this problem. Despite that, SGEA performs similarly to PPS and better than DNSGA-II and MOEA/D on dMOP3. For a graphical view of the algorithms' tracking ability, we also plot their obtained PFs of FDA1, FDA2, FDA3 and dMOP3 over 31 time windows, which are shown in Fig. 7.3. Fig. 7.3 evidently shows that SGEA is very capable of tracking environmental changes, but may be of limited coverage if there is a significant diversity loss (e.g., on dMOP3) in dynamic environments.

7.4.2 Results on ZJZ and UDF Problems

Unlike the FDA and dMOP test suites, the ZJZ (F5-F10) [196] and UDF [11] test problems have nonlinear linkages between decision variables. Also, the ZJZ and UDF test suites introduce a number of new dynamic features which are not included in FDA and dMOP. Table 7.5 reports the HVD values obtained by five algorithms for these challenging problems with $(\tau_t, n_t) = (10, 10)$.

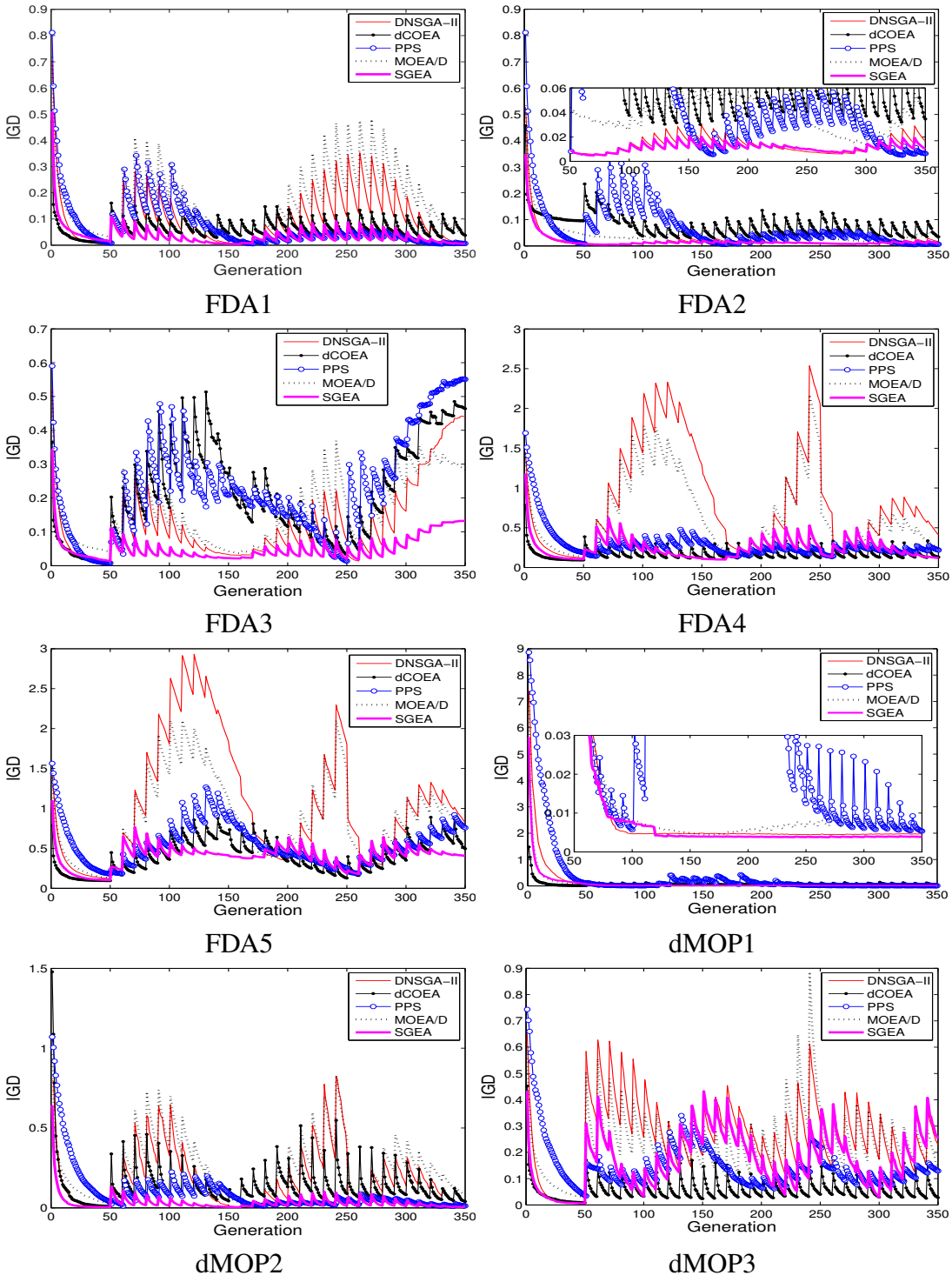
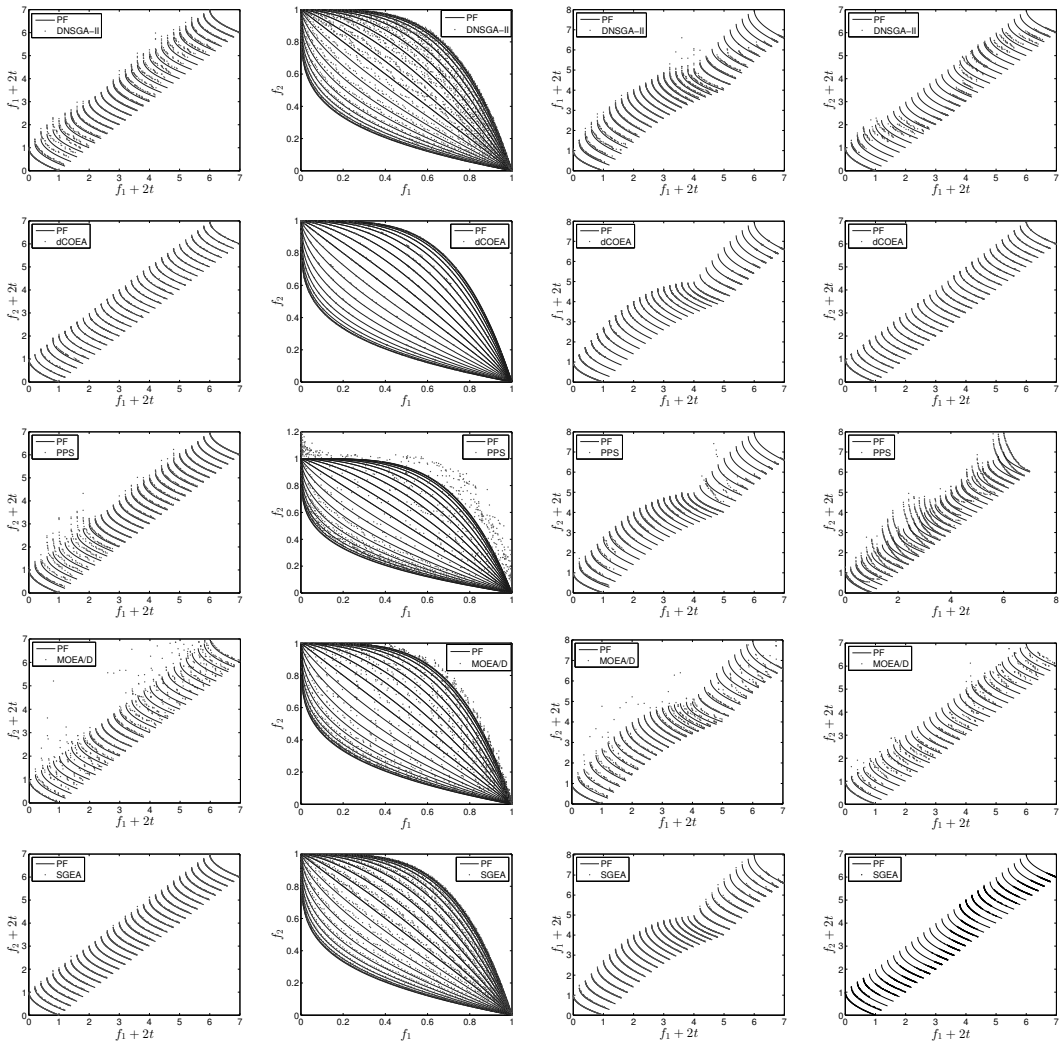


Fig. 7.2 Evolution curves of average *IGD* values for eight problems with $\tau_t = 10$ and $n_t = 10$.

Compared with the average HVD values on FDA and dMOP problems given in Section 7.4.1, the average HVD values obtained on ZJZ and UDF problems are generally much higher, implying that the optimization difficulties are increased in the ZJZ and UDF



FDA1 FDA2 FDA3 dMOP3

Fig. 7.3 Obtained PFs for four problems with $\tau_t = 10$ and $n_t = 10$.

problems. Table 7.5 clearly shows that SGEA and PPS are top performers on these difficult problems. SGEA obtains the best HVD values on some problems while PPS wins on others. SGEA performs significantly better than DNSGA-II on problems F5-F10, but this superiority disappears when they are compared on the UDF problems, and there is no much difference between them. This means SGEA has no much advantage in dealing with difficult variable-linkage UDF problems. PPS, which is not impressive for solving FDA and dMOP problems, shows very promising performance on some ZJZ and UDF problems. This is because PPS employs an estimation of distribution algorithm [192] as its reproduction operator. This operator can exploit problem-specific knowledge and hence is very helpful for solving variable-linkage problems. With the aid of such a powerful operator, it is natural that PPS can obtain competitive results on these variable-linkage DMOPs. In contrast to PPS, dCOEA faces dramatic difficulties to handle the ZJZ and UDF problems, although it has previously shown good performance on FDA and dMOP

Table 7.5 Mean and standard deviation values of HVD metric obtained by five algorithms on ZJZ and UDF problems

Prob.	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
F5	1.2584E+0(2.5806E-2)‡	1.1019E+0(1.6678E-1)‡	**4.0198E-1(9.9177E-2)**	1.1908E+0(2.9956E-2)‡	7.1648E-1(8.2355E-2)
F6	4.7654E-1(3.7611E-2)‡	9.2223E-1(1.0246E-1)‡	4.9294E-1(1.5074E-1)‡	5.7587E-1(7.5659E-2)‡	**3.6068E-1(2.5674E-2)**
F7	6.4963E-1(1.0867E-2)‡	1.2297E+0(1.5928E-1)‡	**4.4905E-1(1.4280E-1)**	6.5075E-1(2.8591E-2)‡	6.0586E-1(1.5195E-2)
F8	1.0626E+0(4.6244E-2)‡	8.8580E-1(1.2482E-1)‡	1.3462E+0(1.0652E-1)‡	1.0615E+0(6.6784E-2)‡	**4.5728E-1(3.2881E-2)**
F9	8.8751E-1(3.4535E-2)‡	1.0741E+0(1.9861E-1)‡	6.8857E-1(7.7943E-2)‡	8.5809E-1(4.6913E-2)‡	**5.7634E-1(7.0349E-2)**
F10	1.2217E+0(5.0091E-2)‡	8.5883E-1(8.8251E-2)‡	**5.3839E-1(1.2028E-1)**†	1.0590E+0(5.9197E-2)‡	5.7721E-1(2.3204E-2)
UDF1	**5.1409E-1(3.2724E-2)**†	7.4761E-1(3.8905E-2)‡	7.9775E-1(5.2094E-2)‡	6.1209E-1(9.4226E-2)‡	5.1825E-1(5.0120E-2)
UDF2	5.5156E-1(2.4931E-2)‡	6.1354E-1(2.8689E-2)‡	**4.3230E-1(1.9124E-2)**	5.4236E-1(1.7627E-2)‡	5.1049E-1(2.5728E-2)
UDF3	1.2217E+0(1.9063E-3)†	1.2314E+0(7.0157E-2)†	1.7374E+0(3.1733E-4)‡	1.2266E+0(2.4696E-3)†	**1.2212E+0(2.4181E-3)**
UDF4	3.4766E-1(8.3674E-2)†	5.0624E-1(3.7884E-2)‡	3.7727E-1(2.1791E-2)‡	6.4101E-1(1.9436E-1)‡	**3.3216E-1(7.1516E-2)**
UDF5	2.7870E-1(2.5461E-2)†	3.9877E-1(3.3025E-2)‡	**2.7052E-1(1.5772E-2)**†	3.6585E-1(2.7331E-2)‡	2.7251E-1(1.8914E-2)
UDF6	**9.3426E-1(1.5483E-1)**	1.2681E+0(7.2900E-2)‡	1.8374E+0(1.0066E-2)‡	1.2118E+0(1.4935E-1)	9.7707E-1(2.0394E-1)
UDF7	2.4041E+0(7.4722E-2)‡	**1.9125E+0(1.7349E-1)**†	2.0607E+0(5.4338E-2)†	2.3287E+0(2.4253E-1)‡	2.0625E+0(1.2304E-1)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

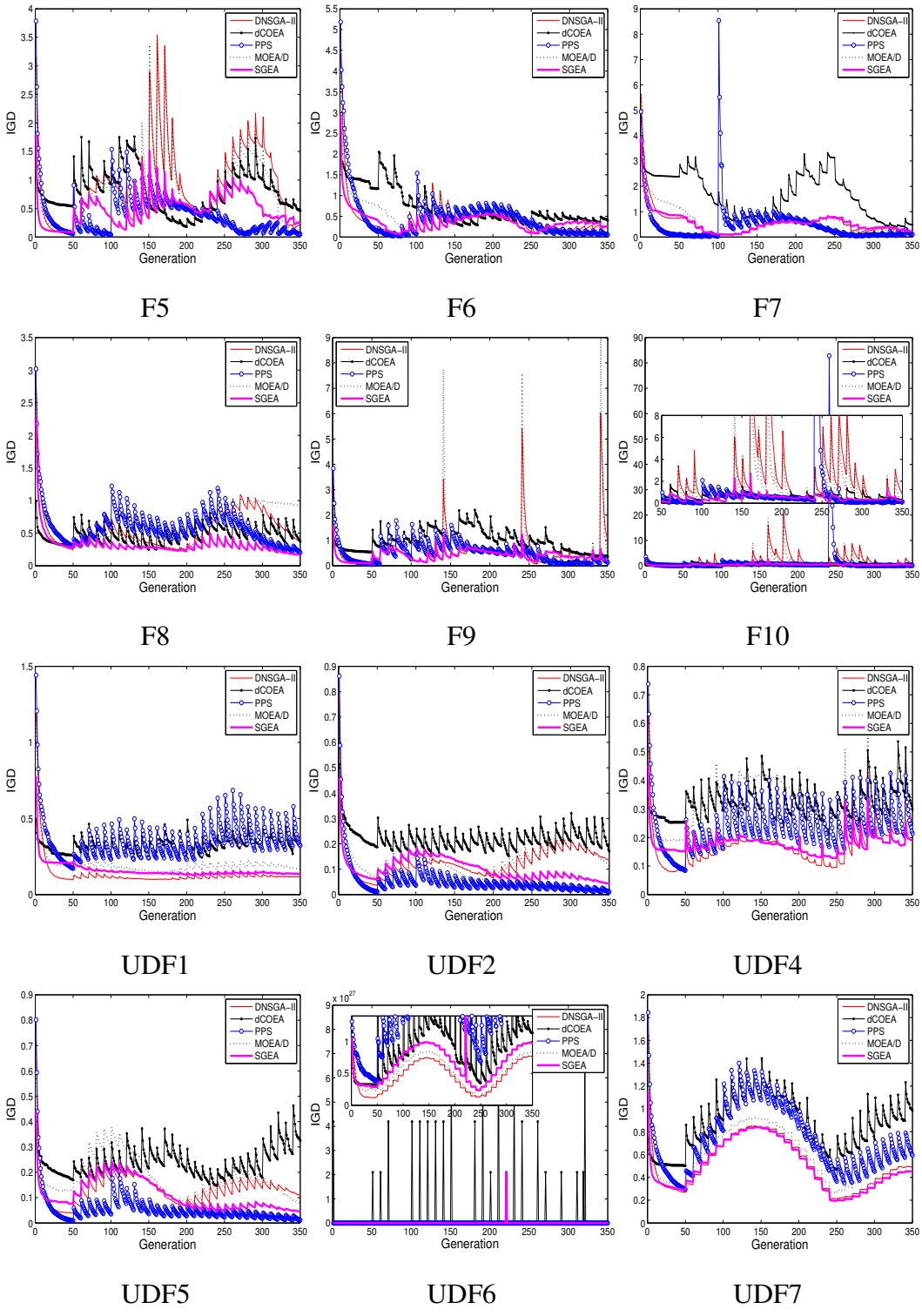


Fig. 7.4 Evolution curves of average *IGD* values for twelve variable-linkage problems with $\tau_t = 10$ and $n_t = 10$.

problems.

Table 7.5 also shows that almost all the tested algorithms are struggling for three-objective problems, i.e., F8 and UDF7, and disconnected problems, i.e., UDF3 and UDF6,

as indicated by their relatively high HVD values. This is understandable because the increase of the number of objectives and disconnectivity are themselves very challenging in static optimization, let alone in dynamic optimization.

To show the evolution performance, Fig. 7.4 plots the evolution curve of the average IGD metric values over 30 independent runs. We can see from the figure that SGEA is able to respond to environmental changes fast and stably in most cases. DNSGA-II and MOEA/D roughly have similar evolution curves on the majority of cases. PPS recovers from environmental changes fast on some problems, e.g., F6, F9, UDF2, and UDF5, but recovers slowly on other problems like F8 and UDF1. dCOEA seems struggling on these variable-linkage DMOPs.

It is worth noting that, the tested algorithms do not react to changes stably on a few problems, e.g., F5, F9, and F10. The IGD values vary widely on these problems because they involves more severe changes in PS than the other ZJZ problems. Clearly, the severe PS movement in F5 degrades the performance of SGEA, hence it is outperformed by PPS.

7.5 Discussions

7.5.1 Influence of Severity of Change

To examine the effect of severity levels on algorithms' performance, experiments were carried out on FDA and dMOP problems with τ_t fixed to 10, and n_t set to 5, 10, and 20, which represent severe, moderate, and slight environmental changes, respectively. Experimental results of five algorithms on the HVD metric are given in Table 7.6.

It can be observed from the table that all the algorithms are very sensitive to the severity of change, as can be seen from the improvement of HVD when increasing the value of n_t. For different severity levels, SGEA is able to produce impressive performance and wins on the majority of the instances, and this algorithm is mainly exceeded by dCOEA on only two problems, i.e., FDA4 and dMOP3. However, for the problem dMOP3, the HVD metric of SGEA deteriorates with the decrease of the severity level. One possible explanation is that, on dMOP3, the amount of diversity loss is roughly the same for different severity levels, but for different severity levels, SGEA reacts to changes differently, with a large movement step-size for severe changes ($n_t = 5$) and a small movement step-size for slight ones ($n_t = 20$). A larger movement step-size is likely to increase more population diversity than a smaller one. Therefore, the increase of n_t may negatively affect population diversity, which in turn leads to the deterioration of the HVD metric. Such impact suggests that SGEA may need diversity increase techniques to deal with problems like dMOP3.

Table 7.6 Mean and standard deviation values of HVD metric obtained by five algorithms with different values of n_t

Prob.	(n_t, τ_t)	DNSGA-II	dCOEA	PPS	MOEA/D	SGEA
FDA1	(5,10)	8.7093e-1(7.5592e-2)‡	9.7456E-2(2.9072E-2)‡	3.2998E-1(7.1818E-2)‡	6.8904E-1(8.8855E-2)‡	**7.3337E-2(3.3442E-2)**
	(10,10)	1.3610e-1(1.7463e-2)‡	8.5252E-2(2.0248E-2)‡	2.9712E-1(1.6596E-2)‡	2.8825E-1(2.9076E-2)‡	**3.8112E-2(1.4430E-2)**
	(20,10)	3.5539e-2(1.3774e-2)‡	7.7223E-2(1.1557E-2)‡	3.0045E-1(4.5258E-2)‡	1.2551E-1(1.1056E-2)‡	**2.6418E-2(6.6551E-3)**
FDA2	(5,10)	4.7185e-2(1.4726e-2)‡	1.4908E-1(1.0125E-1)‡	6.3330E-1(2.7223E-2)‡	1.3071E-1(3.4644E-2)‡	**2.6401E-2(2.7317E-2)**
	(10,10)	2.0598e-2(1.4744e-2)‡	1.2486E-1(4.6708E-2)‡	2.6663E-1(1.4716E-2)‡	6.2906E-2(1.8881E-2)‡	**1.6745E-2(1.4126E-2)**
	(20,10)	1.3369e-2(1.4735e-2)‡	1.2103E-1(3.8214E-2)‡	1.9674E-1(4.0100E-3)‡	3.1918E-2(9.5535E-3)‡	**1.1804E-2(7.2197E-3)**
FDA3	(5,10)	1.5478e+0(1.6485e-1)‡	1.7059E+0(1.1325E-1)‡	1.4126E+0(1.3048E-1)‡	1.5956E+0(9.7437E-2)‡	**1.3476E+0(5.1086E-2)**
	(10,10)	1.0970e+0(9.9053e-2)‡	1.3223E+0(7.7970E-2)‡	1.1626E+0(4.6925E-2)‡	1.1225E+0(9.3170E-2)‡	**9.2413E-1(8.2603E-2)**
	(20,10)	1.0419e+0(7.9293e-2)‡	1.1057E+0(7.6375E-2)‡	1.1209E+0(5.0710E-2)‡	8.0963E-1(2.0865E-2)‡	**7.1753E-1(6.8361E-2)**
FDA4	(5,10)	2.0595e+0(2.0121e-1)‡	**2.6980E-1(4.9426E-2)**	8.2319E-1(1.7066E-1)†	2.9513E+0(1.2324E+0)‡	7.9666E-1(1.7675E-1)
	(10,10)	1.5893e+0(6.6551e-2)‡	**2.7081E-1(3.5707E-2)**†	4.3455E-1(7.2947E-2)‡	1.2458E+0(1.3708E-1)‡	2.7403E-1(2.4337E-2)
	(20,10)	5.4876e-1(5.7277e-2)‡	2.6163E-1(2.4904E-2)‡	3.2629E-1(1.2176E-2)‡	5.2499E-1(6.0127E-2)‡	**1.7814E-1(1.1872E-2)**
FDA5	(5,10)	6.7506e+0(1.9848e-1)‡	3.8546E+0(3.7632E-1)‡	3.6047E+0(5.8770E-1)‡	7.1636E+0(1.0447E+0)‡	**3.5028E+0(2.8444E-1)**
	(10,10)	5.4164e+0(1.6223e-1)‡	2.3772E+0(2.7316E-1)‡	2.1995E+0(3.9061E-1)‡	4.8043E+0(2.6911E-1)‡	**1.8867E+0(9.3801E-2)**
	(20,10)	2.6454e+0(1.1158e-1)‡	1.5328E+0(1.3437E-1)‡	1.1066E+0(1.9178E-1)†	1.9665E+0(1.2036E-1)‡	**1.0914E+0(3.2851E-2)**
dMOP1	(5,10)	3.9375e-2(3.8881e-2)†	1.0840E-1(3.4407E-2)‡	5.4358E-1(4.5618E-1)‡	4.0618E-2(3.0140E-2)†	**3.9348E-2(3.0606E-2)**
	(10,10)	2.2844e-2(2.0365e-2)‡	1.1236E-1(2.0863E-2)‡	9.2754E-2(1.3906E-1)‡	2.5712E-2(1.5458E-2)‡	**1.9048E-2(1.4569E-2)**
	(20,10)	1.7194e-2(1.4792e-2)‡	1.1161E-1(1.5244E-2)‡	2.5362E-2(4.4676E-2)‡	2.0475E-2(1.4607E-2)‡	**1.3659E-2(7.8804E-3)**
dMOP2	(5,10)	8.0662e-1(1.1259e-1)‡	1.9770E-1(4.3839E-2)‡	3.9550E-1(3.9842E-2)‡	9.4022E-1(7.3354E-2)‡	**7.2866E-2(2.5493E-2)**
	(10,10)	2.9084e-1(2.5036e-2)‡	2.0782E-1(2.4976E-2)‡	1.1778E-1(4.3469E-2)‡	4.4678E-1(4.2568E-2)‡	**3.5928E-2(1.1177E-2)**
	(20,10)	4.5002e-2(1.2356e-2)‡	1.7868E-1(1.9365E-2)‡	4.8491E-2(1.0317E-2)‡	1.9085E-1(1.9080E-2)‡	**2.5743E-2(5.5744E-3)**
dMOP3	(5,10)	9.5131e-1(3.4052e-2)‡	**6.6570E-2(2.8299E-2)**	4.0187E-1(5.2945E-2)‡	7.6718E-1(4.0803E-2)‡	2.4586E-1(3.9623E-2)
	(10,10)	4.7415e-1(2.8497e-2)‡	**6.5770E-2(1.3652E-2)**	2.7970E-1(2.7200E-2)‡	4.5433E-1(2.8194E-2)‡	3.1866E-1(2.9555E-2)
	(20,10)	2.7629e-1(2.5543e-2)‡	**6.2846E-2(7.6769E-3)**	2.1883E-1(1.4532E-2)‡	3.4128E-1(1.9083E-2)‡	3.6274E-1(1.9356E-2)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

7.5.2 Study of Different Components of SGEA

This subsection is devoted to studying the effect of different components of SGEA. SGEA has three key components, i.e., the "guided" reinitialization for change response, the steady-state population update, and the generational environmental selection. To deeply examine the role that each component plays in dynamic optimization, we adapt the original SGEA into three variants. The first variant (SGEA-S1) does not use the the part of "guided" change response. Instead, it re-evaluates all current population members in the event of environmental changes. The second variant (SGEA-S2) discards the steady-state upadate part of SGEA. In other words, SGEA-S2 generationally detects and reacts to changes, and reproduces offspring. SGEA-S3 is another modification of SGEA, in which the environmental selection at the end of every generation is conducted by preserving a population of individuals with good fitness. This means, SGEA-S3 prefers well-converged solutions regardless of their diversity. These three variants are compared with the original SGEA on four problems with the setting of $(\tau_t, n_t) = (10, 10)$. Table 7.7 presents the average and standard deviation values of four metrics obtained by different SGEA variants. The Wilcoxon signed-rank test [179] is carried out at the 0.05 significance level to indicate statistically significant difference between SGEA and the three variants.

In Table 7.7, SGEA performs significantly better than the three variants on FDA1 in terms of four metrics, implying all the three key components are crucial to the high performance of SGEA on this problem. For dMOP1, SGEA-S1, SGEA-S2, and SGEA obtain considerably small IGD and HVD values, indicating they can solve this problem very well. In contrast, SGEA-S3 seems incapable of solving dMOP1, as indicated by the inferior four metrics. The poor performance of SGEA-S3 on dMOP1 is mainly due to the lack of diversity maintenance, particularly when excessive nondominated solutions are obtained. This case clearly illustrates the importance of generational environmental selection to SGEA. For F5, there is notable difference between SGEA-S2 and the other algorithms in terms of the metrics. SGEA-S2 obtains the worst S, IGD, and HVD values, although it has better coverage (MS) than the others. The results of SGEA-S2 on F5 obviously suggest that the use of steady-state population update can significantly improve the performance of SGEA. Besides, the difference between SGEA-S1 and SGEA on F5, in terms of the IGD and HVD metrics, also validates the effectiveness of the proposed "guided" population reinitialization for handling environmental changes. The results of four algorithms on UDF1 show that SGEA is significantly better than SGEA-S1 and SGEA-S3. This observation further confirms the benefit of the "guided" population reinitialization and generational selection used in SGEA for dynamic optimization.

It is not difficult to understand that, as a combination of three key components, SGEA generally outperforms the other compared variants. The above observations clearly ex-

Table 7.7 Performance Comparison of SGEA Variants

Problem	Metric	SGEA-S1	SGEA-S2	SGEA-S3	SGEA
FDA1	S	1.0573E-2(1.4425E-3)‡	9.0396E-3(5.2292E-4)‡	9.7488E-3(1.1191E-3)‡	**7.5411E-3(5.8178E-4)**
	MS	9.6158E-1(1.0782E-2)‡	9.6332E-1(1.2436E-2)‡	9.6631E-1(9.1630E-3)‡	**9.7277E-1(1.0854E-2)**
	IGD	1.9931E-2(1.6198E-3)‡	2.6038E-2(2.9232E-3)‡	1.6973E-2(1.8987E-3)‡	**1.4809E-2(2.0621E-3)**
	HVD	3.5270E-1(1.4660E-2)‡	4.9284E-1(1.5196E-2)‡	3.1760E-1(1.2533E-2)‡	**3.8112E-2(1.4430E-2)**
dMOP1	S	**2.4045E-3(1.3972E-4)**	2.7215E-3(3.0493E-4)†	8.5790E-3(1.0189E-3)‡	2.7029E-3(3.0835E-4)
	MS	9.8107E-1(1.8150E-2)†	9.8097E-1(2.4778E-2)‡	8.1045E-1(1.0711E-1)‡	**9.8351E-1(1.3118E-2)**
	IGD	9.0188E-3(4.3106E-3)‡	8.5146E-3(6.9219E-3)‡	4.9847E-2(3.1290E-2)‡	**6.5411E-3(3.0256E-3)**
	HVD	2.0653E-2(1.8715E-2)‡	2.1863E-2(2.3243E-2)‡	1.5789E-1(8.5491E-2)‡	**1.9048E-2(1.4569E-2)**
F5	S	**3.3094E-2(7.3229E-3)**†	9.1399E-2(1.9653E-2)‡	3.5606E-2(9.4640E-3)‡	3.8765E-2(7.1231E-3)
	MS	4.0412E-1(4.2687E-2)‡	**5.5816E-1(4.4388E-2)**	4.7777E-1(4.8569E-2)†	5.0748E-1(4.2309E-2)
	IGD	5.4434E-1(4.3308E-2)‡	6.4214E-1(4.7136E-2)‡	4.5958E-1(3.3362E-2)†	**4.4195E-1(4.5046E-2)**
	HVD	9.2472E-1(6.3984E-2)‡	1.1087E+0(4.2539E-2)‡	7.8997E-1(5.9973E-2)‡	**7.1648E-1(8.2355E-2)**
UDF1	S	2.6490E-2(2.8379E-2)‡	6.8987E-2(3.6611E-2)‡	2.5998E-2(2.0151E-2)‡	**2.1084E-2(1.8674E-2)**
	MS	6.5907E-1(1.8926E-1)‡	**8.5675E-1(8.3575E-2)**	7.1284E-1(1.7661E-1)†	7.2501E-1(1.1595E-1)
	IGD	1.6684E-1(6.5977E-2)‡	**9.9619E-2(1.0661E-2)**†	1.4393E-1(4.6269E-2)‡	1.2449E-1(3.3093E-2)
	HVD	5.7070E-1(8.8933E-2)‡	**4.9834E-1(2.2682E-2)**†	5.3806E-1(6.0746E-2)‡	5.1825E-1(5.0120E-2)

‡ and † indicate SGEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

hibit the importance of each component in dealing with dynamic environments. Here, we would like to give more explanations for the role of each component. The "guided" population reinitialization exploits the information of new environments to coarsely relocate some population members close to the new PS, which is beneficial to rapidly track the changing PS. The steady-state update strategy can speed up the convergence process of the population. This is because, within every generation, when an offspring individual is generated, it is immediately used to update the evolving population and the external archive. Thus, the offspring, if very promising, has opportunities to be chosen as a parent for producing new offspring. This way,the steady-state update strategy offers a fast convergence speed and a steady reaction to changes as well. Although the steady-state update strategy is helpful for convergence, it does not consider population diversity. For this reason, the generational selection strategy is introduced to mainly maintain population diversity. One particular situation is that, when plenty of nondominated individuals are available, the generational selection can prune them so as to preserve a fixed-size population with good diversity for next generation. As a result, the balance between convergence and diversity can be properly struck during the evolution. In a nutshell, all these three components of SGEA play an important role in reacting steadily and adapting rapidly to environmental changes.

7.5.3 Influence of Introducing Mutated Solutions

In the previous section, empirical studies indicate that SGEA is very competitive for handling dynamic environments, but it does not work as well as DNSGA-II for a good distribution and coverage on problems like FDA2. Similar to DNSGA-II, we can introduce $\eta\%$ mutated solutions of existing solutions into the new population of SGEA after a change, and we call this version of SGEA as SGEA-v1. This means, the new population consists of 50% of old solutions, $\eta\%$ ($0 \leq \eta \leq 50$) mutated solutions, and $(50-\eta)\%$ guided solutions. Unlike SGEA, SGEA-v1 computes the moving direction (as shown in Eq. (7.4)) in a different way. To be specific, the 50% old solutions and $\eta\%$ mutated solutions are regarded as the set R in Eq. (7.4), and nondominated solutions from R after re-evaluation are copied to A. This way, $(50-\eta)\%$ of the new population to be re-initialized can benefit from the reused old solutions as well as the mutated solutions, especially when these solutions have a high level of diversity.

The effect of mutated solutions is studied on FDA1 and FDA2 with the setting of $\tau_t = 10$ and $n_t = 10$, and η varied from 0 to 50. In the case of $\eta = 0$, SGEA-v1 is actually the original SGEA, and $\eta = 50$ means there are no guided solutions in the new population. The mutation probability and the distribution index for making mutated solutions were set the same as in DNSGA-II [44].

Table 7.8 S, MS and IGD values of SGEA-v1 for FDA1 and FDA2

Prob.	η	S	MS	HVD
FDA1	0	**7.5411E-3(5.8178E-4)**	**9.7277E-1(1.0854E-2)**	**3.8112E-2(1.4430E-2)**
	10	9.4194E-3(5.2569E-4)	9.7253E-1(9.4358E-3)	3.9221E-2(1.3781E-2)
	20	9.4802E-3(4.1391E-4)	9.7125E-1(9.1182E-3)	4.0204E-2(1.6126E-2)
	30	9.8393E-3(6.5970E-4)	9.6881E-1(1.0522E-2)	4.2156E-2(1.4579E-2)
	40	9.9007E-3(4.8708E-4)	9.6982E-1(7.8856E-3)	4.2984E-2(1.3953E-2)
	50	1.1850E-2(1.4381E-3)	9.6134E-1(1.3593E-2)	5.0541E-2(1.3661E-2)
FDA2	0	**6.5871E-3(8.7753E-4)**	**9.9308E-1(3.3464E-3)**	**1.6745E-2(1.4126E-2)**
	10	7.2513E-3(8.1050E-4)	9.9302E-1(3.1580E-3)	1.7192E-2(1.4852E-2)
	20	7.1854E-3(8.2471E-4)	9.9275E-1(3.9994E-3)	1.7308E-2(1.4810E-2)
	30	6.8743E-3(6.3021E-4)	9.9268E-1(4.8573E-3)	1.7439E-2(1.4599E-2)
	40	7.0167E-3(1.9724E-4)	9.9165E-1(5.3058E-3)	1.7284E-2(1.4777E-2)
	50	7.2156E-3(3.9059E-4)	9.9175E-1(5.0043E-3)	1.7209E-2(1.4652E-2)

Table 7.8 presents the results of SGEA-v1 for the two test problems. For FDA1, the performance of SGEA-v1 on three metrics notably deteriorates with the increase in the number of mutation solutions in population. The similar trend can be observed from the results of FDA2, in which all the metric values are negatively influenced when η increases. The negative effect of introducing mutated solutions can be explained by the fact that, mutated solutions are more random than well-planned guided solutions used in SGEA, and may take more time to be directed toward the true PF. In other words, such mechanism seems not suitable for SGEA when handling dynamic environments.

7.5.4 Influence of Introducing Random Solutions

As illustrated in the previous experimental study, SGEA is quite vulnerable to severe diversity loss and thus cannot compete with dCOEA on dMOP3. For this reason, we devise another version of SGEA, denoted SGEA-v2, which is inspired by the use of stochastic competitors for diversity increase in dCOEA. SGEA-v2 has the similar change response framework to SGEA-v1 except that it replaces $\eta\%$ of the population with randomly created solutions. The influence of introducing random solutions is studied on dMOP3, where η varies from 0 to 50.

The results of SGEA-v2 on dMOP3 with $\tau_t = 10$ and $n_t = 10$ are given in Table 7.9. Clearly, the introduction of random solutions significantly improves the coverage perfor-

Table 7.9 S, MS and IGD values of SGEA-v2 for dMOP3

η	S	MS	IGD
0	**5.4336E-3(6.0751E-4)**	5.7573E-1(2.9590E-2)	1.3248E-1(1.3627E-2)
10	7.8473E-3(1.1449E-3)	8.8779E-1(2.3199E-2)	3.6223E-2(9.4230E-3)
20	8.7294E-3(3.0487E-3)	9.1631E-1(1.1874E-2)	3.0153E-2(4.1528E-3)
30	8.8099E-3(2.8361E-3)	9.3661E-1(1.5004E-2)	2.5424E-2(4.8542E-3)
40	9.3054E-3(2.6714E-3)	**9.4759E-1(1.3064E-2)**	**2.2199E-2(4.1150E-3)**
50	9.4439E-3(2.8387E-3)	9.4656E-1(1.2554E-2)	2.4757E-2(3.3081E-3)

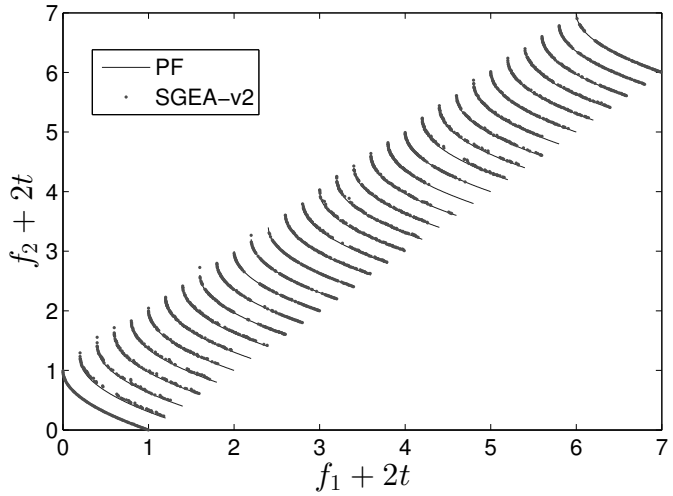


Fig. 7.5 PFs of dMOP3 ($\tau_t = 10$ and $n_t = 10$) obtained by SGEA-v2 over 31 time steps.

mance of SGEA, which in turn decreases the IGD values. Such benefit is maximized when 40% random solutions are adopted, and the corresponding approximations of 31 time steps are illustrated in Fig. 7.5, showing that SGEA with the use of random solutions is very capable of tracking the changing PF on dMOP3. On the other hand, the S metric is negatively affected by random solutions, with a notable decline when η increases. This is because the use of random solutions drastically increases population diversity, leading to a wide spread of the population along the PF so that the uniformity of the obtained approximation is not easy to keep. Thus, for dMOP3, the S metric is inconsistent with MS and IGD.

Since the use of random solutions considerably help SGEA cope with diversity loss, we wonder whether SGEA-v2 can win against the other compared algorithms on dMOP3. SGEA-v2 with $\eta = 40$ is compared with the previous best performer, i.e., dCOEA, on different dynamic scenarios of dMOP3. Table 7.10 and Fig. 7.6 present the comparison results, clearly showing that SGEA-v2 significantly outperforms dCOEA in terms of the

Table 7.10 Comparison between dCOEA and SGEA-v2 on dMOP3

Metric	(τ_t, n_t)	dCOEA	SGEA-v2
S	(5,10)	3.3786E-2(5.5519E-3)	**1.9335E-2(4.5045E-3)**
	(10,10)	1.5418E-2(1.0978E-3)	**9.3054E-3(2.6714E-3)**
	(20,10)	7.3129E-3(3.9782E-4)	**4.8211E-3(1.0737E-3)**
	(10,5)	1.6060E-2(1.9712E-3)	**1.0110E-2(2.6696E-3)**
	(10,20)	1.4903E-2(7.3451E-3)	**7.7303E-3(1.2510E-3)**
MS	(5,10)	8.7837E-1(2.1444E-2)	**9.1029E-1(1.2201E-2)**
	(10,10)	9.1097E-1(1.1716E-2)	**9.4759E-1(1.3064E-2)**
	(20,10)	9.4844E-1(1.1052E-2)	**9.6644E-1(8.0110E-3)**
	(10,5)	9.1484E-1(1.4420E-2)	**9.3430E-1(1.4221E-2)**
	(10,20)	9.1736E-1(8.7132E-3)	**9.4923E-1(1.2537E-2)**
IGD	(5,10)	4.9556E-2(4.8079E-3)	**4.7222E-2(5.2525E-3)**
	(10,10)	2.9589E-2(2.4806E-3)	**2.2199E-2(4.1150E-3)**
	(20,10)	1.6366E-2(1.7152E-3)	**1.1448E-2(2.5416E-3)**
	(10,5)	2.9953E-2(3.5438E-3)	**2.5428E-2(4.8462E-3)**
	(10,20)	3.1686E-2(2.6402E-3)	**2.2085E-2(4.5406E-3)**

three performance metrics. This further confirms the potential of SGEA for handling dynamic environments if the population diversity is properly maintained.

7.5.5 Further Discussion

The previous experimental comparison and analysis have shown that SGEA is capable of solving a wide range of DMOPs. Specifically, SGEA works well on simple DMOPs without strong variable linkages, like most of the FDA and dMOP problems. In some patterns of changes, such as, the geometric shapes of two consecutive PFs/PSs are similar, changes are slight or do not cause diversity loss, and changes are relatively smooth, SGEA is able to track the moving PFs/PSs effectively and efficiently. Therefore, SGEA provides better performance than the other compared algorithms in these cases. The fact that most of the test problems have periodical changes suggests SGEA is particularly applicable to periodical environments.

However, like other algorithms, SGEA has some drawbacks too. One drawback is that SGEA struggles to deal with changes that bring about severe diversity loss, which

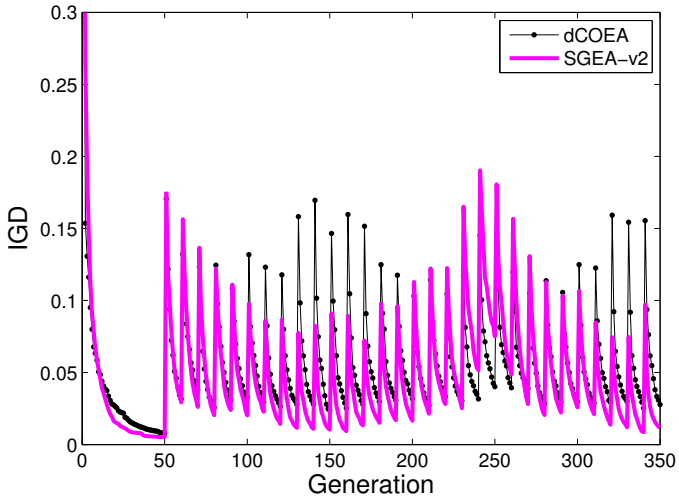


Fig. 7.6 Comparison of IGD curves between dCOEA and SGEA-v2 for dMOP3 with $\tau_t = 10$ and $n_t = 10$.

has been illustrated by dMOP3. In practice, SGEA does not increase diversity when changes occur, so it is vulnerable to the loss of diversity. However, as have shown in our study, this drawback can be alleviated by introducing some randomly created individuals when a change is detected. Another drawback comes from the inefficiency of SGEA for handling severe movements in PS. As verified by F5, such a severe change can significantly degrade the performance of SGEA. Besides, the dissimilar geometric shapes between two consecutive PFs/PSs (see results on F10) may challenge the performance of SGEA. In case that a change affects too much the relative positions between solutions in the PS, the guided reinitialization method of SGEA may not work well due to its linear property. On the other hand, SGEA also suffers from optimization difficulties caused by variable linkages. Experimental comparisons on the UDF problems evidently show that SGEA and the other algorithms all have difficulty in solving strong variable-linkage problems. A possible way to solve variable linkages may be borrowing similar idea from the optimizer of PPS or incorporating with new operators [163] to evolve the population.

7.6 Summary

This chapter has presented a steady-state and generational EAs, i.e., SGEA, for handling multiobjective problems with time-varying characteristics. Different from existing dynamism handling approaches in the literature, SGEA detects and reacts to changes in a steady-state manner. If a change is detected, SGEA reuses a portion of old solutions with good diversity and re-evaluates them, providing the algorithm with some basic understanding of the landscape of the new environment. As a result, SGEA exploits useful information extracted from the new environment, i.e., the moving direction, to relocate

the remaining portion of population to regions near the new PF. Otherwise, a generational cycle of static steady-state optimization is executed, in which the evolving population progressively interacts with an external archive, promoting the convergence speed of SGEA. At the end of each generation, the previous and current populations are combined, and the environmental selection is performed on the combined population to preserve elitists for the next generation.

SGEA has been compared with other several popular DMOEAs on a number of DMOPs, including bi- and three-objective problems, with different dynamic characteristics and difficulties. Experimental studies have shown that, on the majority of the considered problems, SGEA is capable of tracking their changing PFs efficiently, but may struggle to recover if the problem has strong variable linkages or changes cause a significant diversity loss.

The main components of SGEA have been studied and their roles in handling dynamic environments have been deeply illustrated. Besides, the influence of the introduction of mutated and randomly created solutions for change reaction has been investigated, showing that mutated solutions may have a negative effect on the elaborated SGEA, and the use of random solutions can considerably alleviate the diversity loss caused by environmental changes, thereby offering significant improvement on the performance of SGEA.

Although SGEA has provided encouraging performance on the test problems considered in this work, it needs to be examined on a wider range of dynamic environments, such as changes that are hard to be detected or do not vary regularly. Our future work includes the incorporation of new constraint handling techniques to deal with dynamic constrained problems, new operators like [163] to evolve population, new detectors and response mechanisms to handle environmental changes. Besides, new dynamic benchmarks and performance metrics are needed to facilitate the analysis of DMOEAs.

Chapter 8

Conclusions and Future Work

In this chapter, the general results obtained from the previous chapters are summarized. The contributions of this thesis are mentioned concerning every presented research work. Moreover, different directions of future work that could extend on what this thesis presents are summarized in the last part of the chapter.

8.1 Conclusions

Multiobjective problems occur frequently in practice and EAs have been applied successfully in many application areas to approximate Pareto-optimal solutions. However, EAs' performance depends on problem properties and optimization environments. For example, if an MOP has many objecitves, most state-of-the-art EAs will have difficulties to find good approximations of the PF. This thesis has made a significant step towards understanding what causes optimization difficulties for EAs and figuring out how to help EAs overcome these difficulties. The general results and contributions are summarized as follows.

In Chapter 3, an improved decomposition-based EA (MOEA/D-TPN) equipped with some new strategies is presented for solving MOPs with complex PFs. The proposed MOEA/D-TPN is examined on a number of complex MOPs, including both existing and newly-developed ones. Experimental studies show that, compared with peer algorithms and several state-of-the-arts, MOEA/D-TPN is more capable of handling MOPs with complex PF properties. They also indicate that the suggested strategies help improve population diversity, avoid evolutionary stagnation during the search, and achieve a set of well-distributed solutions close to the PF. The main contribution of the proposed MOEA/D-TPN is the use of two new strategies that enhance the search in underexplored regions due to their complex properties. Another contribution is the ability of MOEA/D-TPN to find a good approximation of well-diversified and well-converged solutions for MOPs with complex PFs.

In Chapter 4, two new families of scalarizing functions are proposed to overcome the difficulties faced by existing scalarizing functions, and their impact on EAs' search behavior is analyzed. Based on the proposed scalarizing functions, an efficient decomposition-based EA (eMOEA/D) is suggested to deal with hard MOPs with local attractors. Experimental studies show two main findings. First, EAs need different scalarizing functions for different search stages, and the suggested scalarizing function families perform generally better than the well-known weight chebycheff function. Second, the proposed eMOEA/D approach solves hard MOPs better in comparison with nine state-of-the-art algorithms. The main contribution is the proposal of new scalarizing functions that control EAs' search behavior nicely. The second contribution is the deep analysis of the performance of the proposed scalarizing functions. The third contribution is the efficient version of MOEA/D that conducts solution selection and update in an efficient way.

Chapter 5 presents a diversity-first-and-convergence-second (DFCS) selection method for EAs, making an attempt to overcome potential diversity loss caused by widely-used convergence-first based selection methods. Based on DFCS, a new version of strength Pareto EA (SPEA/R) is introduced in the hope of having the ability to deal with both MOPs and many-objective problems. As a starting point, SPEA/R is compared with top-performing algorithms on difficult MOPs, showing it is very capable of maintaining diversity during the search and producing high-quality solutions. Having exhibited this encouraging performance, SPEA/R is then examined on many-objective optimization. Experimental studies show that SPEA/R is very promising for solving many-objective problems with up to 12 objectives. The contributions of this work are manifold. First, a novel selection method is proposed for improving EAs' ability to balance diversity and convergence. Second, a new weight vector generation method is introduced to produce uniformly-distributed weight vectors. Third, a new EA framework is developed for handling both MOPs and many-objective problems. Fourth, experimental studies have revealed an interesting finding that diversity may play a more important role than convergence in some many-objective optimization problems.

In Chapter 6, new test problems are presented to enrich test environments in the field of dynamic multiobjective optimization. Several new performance metrics are proposed to help the performance assessment of EAs in dynamic environments. Extensive empirical studies are conducted to examine the usefulness of the proposed test suite. Moreover, algorithm comparisons are made to study the dynamic environments and help understand the impact of environmental changes on EAs' performance. This work has led to three key contributions. The main contribution is the proposal of new test problems containing diverse dynamic characteristics. The second contribution is the introduction of new performance metrics that are suitable for performance assessment and algorithm comparisons in dynamic multiobjective environments. The last contribution is the extensive algorithm

comparisons which provide great insights in the impact of dynamic environments on EAs' performance.

Finally, in Chapter 7, a new optimizer (SGEA) is introduced for handling dynamic MOPs. The proposed SGEA approach is a combination of generational and steady-state EAs, and therefore inherits the advantages of both kinds of EAs, i.e., good diversity maintenance and fast convergence speed. Furthermore, SGEA introduces a new change detection mechanism and a guided change reaction mechanism, which enable SGEA to detect changes timely and effectively and respond to them efficiently. SGEA is tested with several popular EAs existing in the field of dynamic multiobjective optimization on various MOPs with different dynamic properties. This empirical study demonstrates the usefulness and potential for dealing with dynamic environments. Three important contributions have been made regarding this work. The first contribution lies in an initial attempt to combine generational and steady-state EAs for dynamic environments. The second contribution is the new steady-state change detection method and the guided change response strategy, which represent a great advance in the design of dynamics handling mechanisms. The third contribution is the promise of SGEA that shows very nice ability to track environmental changes and provide high-quality solutions for each static period of environmental changes.

8.2 Future Work

Concerning the proposed methodologies, there are several possible aspects for future work that have arisen from this research. These aspects can be summarized as follows:

Applicability of Decomposition-based EAs As a popular metaheuristic, EAs based on decomposition have been widely applied to various MOPs, varying from simple static problems [188] to dynamic ones [11]. Despite wide application with various degrees of success, decomposition-based EAs have their own concerns, which limit their applicability to some types of problems. A particular type of problems where decomposition-based EAs seem unpromising is the problem with disconnected PF segments. The poor performance is because decomposition-based EAs are invented under the assumption that the PF is continuous. If there are disconnected PF segments, some weight vectors or subproblems will be wasted because their associated solutions are dominated. To widen the applicability of decomposition-based EAs, weight vectors or subproblems should be constructed to avoid producing undesirable dominated solutions. Another method can be the use of detection methods which detect regions between disconnected PF segments and avoid searching these regions. Besides, the proposed MOEA/D-TPN can be extended to many-objective optimization and dynamic multi-objective optimization. We will also consider to

develop a self-adaptive version of MOEA/D-TPN that is less sensitive to parameter settings and even free from key parameters. Another interesting research line will be to design a smart version of MOEA/D-TPN, where the solutions on the PF are smart distributed [70, 77, 129]. In other words, we will emphasize the regions of the PF that entail significant trade-off, and de-emphasizes the regions corresponding to little trade-off. Besides, the issue on how to set an arbitrary population size of MOEA/D in many-objective cases needs to be addressed in the future.

Test Suites Test suites are very important for theoretical study, as they not only ease the theoretical analysis of problems, but also help evaluate EAs and in turn improve their performance. In static multiobjective optimization, including both multiobjective and many-objective cases, ZDT [200], LZ [120], DTLZ [48], and WFG [82] are commonly used test suites. Despite undisputed importance for performance assessment, there test suites have underlying biases. A particular feature is that all these problems can be mathematically formulated by two separate functions, the position-related function and the distance-related function. However, it remains unknown how popular this kind of problem is in real-life applications. Another feature is that the PF geometry of these test problems ideally covers the whole objective space if the PF is continuous. This feature favours some algorithms, particularly decomposition-based algorithms as their weight vectors can be easily designed to cover the whole objective space. If the PF does not contain any solution on the axis, some existing EAs may face great difficulties to perform well. On the other hand, in the field of dynamic multiobjective optimization, the popular test suites like FDA [55] and dMOP [55] are far from being able to cover representative dynamic features that occur in real-world applications. It is expected to make a deep survey on the practicability of existing test suites, thereby providing a useful guideline on how to make diverse test environments.

Variation Operators Genetic variation is a necessity of the process of evolution. It has been shown in several chapters of this thesis that variation operators have a great impact on EAs' performance. Some variation operators like simulated binary crossover (SBX) [42] are good for diversity whereas others like differential evolution (DE) [138] is beneficial for convergence. SBX and DE are sufficient for solving simple MOPs while estimation-based recombination operators are required to solve difficult MOPs with variable linkages [192]. This means different variation operators are suitable for different MOPs and for different search stages. In order to maximize EAs' performance, a suitable variation operator or a method of automating the selection of variation operators is needed. In many-objective optimization. It often happens that individuals tend to be far from each other, and recombining two distant individuals are not likely to generate good offspring. As a result, this

slows down the search process and therefore affects the final performance of EAs. In this situation, neighbouring mating or restricted mating may be a good choice and but needs to be deeply investigated. In dynamic multiobjective optimization, it is desirable for EAs to have a fast convergence speed so as to be able to track environmental changes. Thus, fast-converging or dynamics-based variation operators may be useful in this situation.

Change Detection in DMO Change detection is an important step in EDMO if changes in the environment are made unknown to algorithms. Generally, change detection can be conducted by either re-evaluating a portion of existing solutions [44, 61, 182, 196, 197] or assessing some statistical information of some selected population members [55, 125, 140]. Since both methods choose a small proportion of population members as detectors, detection may fail if changes occur on non-detectors. However, most environmental changes in existing dynamic test problems are easily detectable and impose no challenges to these methods. In other words, the effectiveness of these change detection techniques is not thoroughly examined by existing studies in the literature. Therefore, change detection techniques are far from being well understood in terms of their ability to detect possible environmental changes.

An interesting research direction is to make an extensive investigation into the ability of different change detection methods. To do this, first, dynamic test problems that have partially or wholly hard-to-detect changes should be constructed. Then, different change detection methods are tested on the designed test problems, and observations need to be deeply analyzed to well understand their detection ability. On the other hand, it is worthy of attention to the efficiency of each change detection method because different methods may need different number of detectors to be able to detect changes. In some cases, it may require the whole population members to be detectors in order to detect changes, which is computationally expensive.

Linking Theory to Practice It has been increasingly recognized that there is a lack of clear links between theoretical research and real-world applications. This issue is related to how realistic the current academic research is and how to use the obtained research findings to solve similar real-world problems. To answer these questions, a deep investigation should be made to analyze similarities and differences between current academic benchmark problems and real-world problems. Based on the observed similarities and differences, benchmark test suites can be improved or widened by introducing some uncaptured characteristics of real-world problems. This way, the link between theory and practice can be built up and academic research can move forward in a practical way.

All presented research works throughout this thesis are concentrated mainly on a number of benchmark test suites. These test suites, however, are largely used for academic purposes and hardly represent exact characteristics of real-life problems. It remains unknown that how much the presented research is close to real-life applications. In the future, it is worth extending the current research work to real-world problems, such as vehicle routing problems [100] and railway junction rescheduling problems [52].

We can conclude that the field of EMO has still many interesting questions to offer for future research. Especially the theoretical investigation of EAs' search behavior for various multiobjective optimization topics is still in its infancies and, from the author's point of view, will yield fruitful results in the future.

Bibliography

[1] H. Abdi, "Bonferroni and Šidák corrections for multiple comparison," in *Encyclopedia of Measurement and Statistics*, N. J. Salkind, Eds. Thousand Oaks, CA: Sage, 2007.

[2] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, 2011.

[3] V. Aragón, S. Esquivel, and C. A. Coello Coello, "Evolutionary multiobjective optimization in non-stationary environments," *J. Comput. Sci. Technol.*, vol. 5, no. 3, pp. 133–143, 2005.

[4] M. Aruldoss, T. M. Lakshmi, and V. P. Venkatesan, "A survey on multi criteria decision making methods and its applications," *Am. J. Inf. Syst.*, vol. 1, no. 1, pp. 31–43, 2013.

[5] C. R. B. Azevedo and A. F. R. Araujo, "Generalized immigration schemes for dynamic evolutionary multiobjective optimization," in *Proc. 2011 IEEE Congr. Evol. Comput. (CEC)*, 2011, pp. 2033–2040.

[6] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[7] L. S. Batista, F. Campelo, F. G. Guimaraes, and J. A. Ramirez, "Pareto cone ε-dominance: Improving convergence and diversityin multiobjective evolutionary algorithms," in *Proc. 6th Int. Conf. Evol. Multi-criterion Optimiz.*, LNCS, vol. 6576, 2011, pp. 76–90.

[8] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1652–1669, 2007.

[9] S. Biswas, D. Bose, S. Das, and S. Kundu, "Decomposition-based evolutionary multi-objective optimization approach to the design of concentric circular antenna arrays," *Prog. Electromagn. Res. B*, vol. 52, pp.185–205, 2013.

[10] S. Biswas and S. Das, "Utilizing time-linkage property in DOPs: An information sharing based artificial bee colony algorithm for tracking multiple optima in uncertain environments," *Soft Computing*, vol. 18, no. 6, pp. 1199–1212, 2014.

[11] S. Biswas, S. Das, P. N. Suganthan, and C. A. C. Coello, "Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions," in *Proc. 2014 IEEE Congr. Evol. Comput.*, 2014, pp. 3192–3199.

[12] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, 2014.

[13] T. Blackwell and J. Branke, "Multi-swarm optimization in dynamic environments," in *EvoWorkshops 2004: Appl. Evol. Comput.*, 2004, LNCS, vol. 3005, pp. 489–500.

[14] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. 1999 IEEE Congr. Evol. Comput. (CEC)*, vol. 3, 1999, pp. 1875–1882.

[15] D. Brockhoff, *Many-objective Optimization and Hypervolume Based Search*, Ph.D Thesis, Universität Dortmund, Germany, 2009.

[16] D. Brockhoff and E. Zitzler, "Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization," in *Proc. Parallel Problem Solving from Nature (PPSN IX)*, 2006, pp. 533–542.

[17] D. Brockhoff, T. Wagner, and H. Trautmann, "On The Properties of The R2 Indicator," in *Proc. 14th Ann. Conf. Genetic and Evolutionary Algorithm (GECCO'12)*, 2012, pp. 465–472.

[18] D. Brockhoff, T. Wagner, and H. Trautmann, "R2 Indicator-based Multiobjective Search," *Evol. Comput.*, vol. 23, no. 3, pp. 369–395.

[19] D. Brockhoff and E. Zitzler, "Dimensionality reduction in multiobjective optimization: The minimum objective subset problem," in *Oper. Res. Proc. 2006*, vol. 2006, 2007, pp. 423–429.

[20] L. T. Bui, M. Zbignew, P. Eddy, and B. A. Manuel, "Adaptation in dynamic environments: A case study in mission planning," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 190–209, 2012.

[21] D. Cai and Y. Wang, "A new multiobjective evolutionary algorithm based on decomposition of the objective space for multiobjective optimization," *J. Appl. Math.*, vol. 2014, pp. 1–9, 2014.

[22] M. Cámara, *Parallel Processing for Dynamic Multi-objective Optimization*, Ph.D. Thesis, Universidad de Granada, Dept. of Computer Architecture and Computer Technology, Universidad de Granada, Spain, April 2010.

[23] M. Cámara, J. Ortega, and F. de Toro, "Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms," in *Advances in Multi-Objective Nature Inspired Computing*, C. A. Coello Coello, C. Dhaenens and L. Jourdan, Eds. Berlin: Springer, 2010, vol. 272, pp. 63–86.

[24] M. Cámara, J. Ortega, and F. de Toro, "A single front genetic algorithm for parallel multi-objective optimization in dynamic environments," *Neurocomputing*, vol. 72, nos. 16–18, pp. 3570–3579, 2009.

[25] R. B. Carlos, A. Azevedo, and F. R. Araújo Aluizio, "Non-dominance landscapes for guiding diversity generation in dynamic environments," in *Proc. Brazilian Conf. Intelligent System (BRACIS)*, 2012.

[26] C. Chen and L. Y. Tseng, "An improved version of the multiple trajectory search for real value multiobjective optimization problems," *Eng. Optim.*, vol. 46, no. 10, pp. 1430–1445, 2014.

[27] B. Chen, W. Zeng, Y. Lin, and D. Zhang, "A new local search-based multiobjective optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 50–73, 2015.

[28] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, 2016, in press.

[29] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multi- and many-objective optimization, " *IEEE Trans. Cybern.*, 2016, in press.

[30] Y. Cheung, F. Gu, and H. Liu, "Objective extraction for many-objective optimization problems: Algorithm and test problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 755–772, 2015.

[31] C. K. Chow and S. Y. Yuen, "A multiobjective evolutionary algorithm that diversifies population by its density," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 149–172, 2012.

[32] H. Cobb, "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent non-stationary environments," Naval Research Laboratory, Tech. Rep. AIC-90-001, 1990.

[33] C. A. Coello Coello, C. Dhaenens, and L. Jourdan, *Advances in Multi-Objective Nature Inspired Computing*, ser. Studies in Computational Intelligence. Berlin: Springer, 2010, vol. 272.

[34] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorhms for Solving Multi-Objective Problems*. Boston, MA: Kluwer Academic Publishers, 2002.

[35] D. W. Corne and J. D. Knowles, "The Pareto-envelope based selection algorithm for multiobjective optimization," in *Proc. Parallel Problem Solving from Nature (PPSN VI)*, 2000, pp. 839–848.

[36] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. Genetic Evol. Comput. Conf. (GECCO'01)*, 2001, pp. 283–290.

[37] C. Cruz, J. González, and D. Pelta, "Optimization in dynamic environments: a survey on problems, methods and measures," *Soft Computing*, vol. 15, no. 7, pp. 1427–1448, 2011.

[38] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, Nature (5th ed.), London: John Murray, 1869.

[39] I. Das and J. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optimiz.*, vol. 8, no. 3, pp. 631–657, 1998.

[40] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 966–978, 2013.

[41] K. Deb, S. Agrawwal, A. Pratap, and T. Meyarivan, " A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[42] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[43] K. Deb and H. Jain, "An evolutionary many-Objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[44] K. Deb and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. 4th Int. Conf. on Evol. Multi-Criterion Optim.*, LNCS, vol. 4403, 2007, pp. 803–817.

[45] K. Deb and A. Pratap, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[46] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley and Sons, 2001.

[47] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, Oct. 2010.

[48] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scable test problems for evolutionary multi-objective optimization," in *Evolutionary Multiobjective Optimization*. London: Springer, 2005, pp. 105–145.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zzler, "Scable test problems for evolutionary multi-objective optimization," Kanpur Genetic Algorhms Lab (KanGAL), Indian Inst. Technol., KanGAL Rep. 2001001, 2001.

[50] B. Derbel, D. Brockhoff, A. Liefooghe, and S. Verel, "On the impact of multiobjective scalarizing functions," in *Proc. 8th Int. Conf. Parallel Problem Solving from Nature (PPSN XIII)*, LNCS, vol. 8672, 2014, pp. 548–558.

[51] F. di Pierro, S.T. Khu, and D. A. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, 2007.

[52] J. Eaton, S. Yang, and M. Mavrovouniotis, "Ant colony optimization with immigrants schemes for the dynamic railway junction rescheduling problem with multiple delays," *Soft Computing*, vol. 20, no. 8, pp. 2951–2966, 2015.

[53] E. U. Ergul and I. Eminoglu, "DOPGA: A new fitness assignment scheme for multiobjective evolutionary algorithms," *Int. J. Syst. Sci.*, vol. 45, no. 3, pp. 407–426, 2014.

[54] M. Farina and P. Amato, "A fuzzy definition of optimality for many-criteria problems," *IEEE Trans. Syst., Man and Cybern. Part A: Syst. and Humans*, vol. 34, no. 3, pp. 315–326, 2004.

[55] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, 2004.

[56] C. M. Fonseca and P.J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. 5th Int. Conf. Genetic Algorithms*, 1993, pp. 416–423.

[57] H. Fu, B. Sendhoff, K. Tang, and X. Yao, "Finding robust solutions to dynamic optimization problems," in *Proc. 16th Europ. Conf. on Appl. Evol. Comput.*, LNCS, vol. 7835, 2013, pp. 616–625.

[58] S. B. Gee, K. C. Tan, and H. A. Abbass, "A benchmark test suite for dynamic evolutionary multiobjective optimization", *IEEE Trans. Cybern.*, 2016, in press.

[59] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inform. Sci.*, vol. 282, pp. 363–387, 2014.

[60] C. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, 2007.

[61] C. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, 2009.

[62] D. E. Goldberg and J. Richardson."Genetic algorhms wh sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genet. Algorhms*, Hillsdale, NJ, USA, 1987, pp. 41–49.

[63] R. H. Gómez and C. A. Coello Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," *Proc. 2015 Ann. Conf. Genetic and Evolutionary Algorithm (GECCO'15)*, 2015, pp. 679–686.

[64] G. Greenwood, X. Hu, and J. D'Ambrosio, "Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings," in *Foundations of Genetic Algorithms*, vol. 4, 1997, pp. 437–455

[65] M. Greff and A. P. Engelbrecht, "Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation," in *Proc. 2008 IEEE Congr. Evol. Comput. (CEC)*, 2008, pp. 2917–2924.

[66] F. Gu and H. Liu, "A novel weight design in multi-objective evolutionary algorhm," in *Proc. Int. Conf. Comput. Intell. and Secur.*, Naning, China, 2010, pp. 137–141.

[67] S. Guan, Q. Chen, and W. Mo, "Evolving dynamic multi-objective optimization problems with objective replacement," *Artif. Intell. Review*, vol. 23, no. 3, pp. 267–293, 2005.

[68] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective optimization," *Evol. Comput.*, vol. 20, no. 3, pp. 423–453, 2012.

[69] U. Halder, S. Das, and D. Maity, "A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 881–897, 2013.

[70] B. J. Hancock and C. B. Mattson, "The smart normal constraint method for directly generating a smart pareto set," *Struct. Multidiscipl. Optim.*, vol. 48, no. 4, pp. 763–775, 2013.

[71] I. Hatzakis and D. Wallace, "Dynamic multiobjective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 8th Ann. Conf. Genetic Evol. Comput. (GECCO)*, 2006, pp. 1201–1208.

[72] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, 2014.

[73] M. Helbig and A. P. Engelbrecht, "Benchmarks for dynamic multi-objective optimisation," in *Proc. 2013 IEEE Symp. on Comput. Intell. in Dynamic and Uncertain Environments*, 2013, pp. 84–91.

[74] M. Helbig and A. P. Engelbrecht, "Performance measures for dynamic multi-objective optimisation algorithms," *Inform. Sci.*, vol. 250, pp. 61–81, 2013.

[75] M. Helbig and A. P. Engelbrecht, "Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems," *Swarm Evol. Comput.*, vol. 14, pp. 31–47, 2014.

[76] M. Helbig and A. P. Engelbrecht, "Benchmarks for dynamic multi-objective optimisation algorithms," *ACM Comput. Surv.*, vol. 46, no. 3, Article No. 37, 2014.

[77] J. M. Herrero, G. Reynoso-Meza, M. Martnez, X. Blasco, and J. Sanchis, "A smart-distributed pareto front using ev-MOGA evolutionary algorhm," *Int. J. Artif. Intell. Tools*. vol. 23, no. 2, 2014, Art. ID 1450002.

[78] F. J. Hickernell, "A generalized discrepancy and quadrature error bound," *Math. Comput.*, vol. 67, no. 221, pp. 299–322, 1998.

[79] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. New York: Wiley-Interscience, 1999.

[80] J. Horn, N. Nafpliotis, and D. E. Goldberg, "Multiobjective optimization using the niched Pareto genetic algorithm," University of Illinois, Tech. Rep. 930005, 1994.

[81] L. Huang, I. Suh, and A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Inform. Sci.*, vol. 181, no. 11, pp. 2370–2391, 2011.

[82] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 2, pp. 477–506, 2006.

[83] E. J. Hughes, "Multiple single objective Pareto sampling," in Proc. IEEE Congr. Evol. Comput., Canberra, ACT, Australia, 2003, pp. 2678–2684.

[84] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *Proc. 11th Int. Conf. Parallel Problem Solving from Nature (PPSN X)*, 2010, pp. 91–100.

[85] H. Ishibuchi, Y. Sakane,N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Evolut. Multi-Criterion Optimiz.*, Springer, 2009, pp. 438–452.

[86] H. Ishibuchi, Y. Sakane,N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. 12th Ann. Conf. Genetic Evol. Comput. (GECCO 2010)*, 2010, pp. 519–526.

[87] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. 2008 IEEE Congr. Evol. Comput.*, 2008, pp. 2424–2431.

[88] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Iterative approach to indicator-based multiobjective optimization," in *Proc. 2007 IEEE Congr. Evolut. Comput.*, 2007, pp. 3697–3704.

[89] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing function," in *Proc. 12th Ann. Conf. Genetic Evol. Comput. (GECCO 2010)*, 2010, pp. 527–534.

[90] S. Jiang and S. Yang, "An improved multi-objective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, 2016.

[91] S. Jiang and S. Yang, "A benchmark generator for dynamic multi-objective optimization problems," in *Proc. 2014 UK Workshop Comput. Intell. (UKCI)*, 2014, pp. 1–8.

[92] S. Jiang and S. Yang, "Evolutionary dynamic multi-objective optimization: Benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, 2016, in press.

[93] S. Jiang and S. Yang, "A framework of scalable dynamic test problems for dynamic multi-objective optimization," in *Proc. 2014 IEEE Symp. Comput. Intell. Dynamic and Uncertain Environments (CIDUE)*, 2014, pp. 32–39.

[94] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization," *IEEE Trans. Evol. Comput.*, 2016, in press.

[95] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based evolutionary algorithms," submitted to *IEEE Trans. Evol. Comput.*, 2016.

[96] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Appl. Evolut. Comput.*, LNCS, vol. 3005, 2004, pp. 525–536.

[97] Y. Jin, K. Tang, X.Yu, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Computing*, vol. 5, no. 1, pp. 3–18, 2013.

[98] F. Jones, *Lebesgue Integration on Euclidean Space*. London: Jones and Bartlett Learning, 2001.

[99] R. Joshi and B. Deshpande, "Empirical and analytical study of many-objective optimization problems: Analysing distribution of nondominated solutions and population size for scalability of randomized heuristics," *Memetic Computing*, vol. 6, no. 2, pp. 133–145, 2014.

[100] N. Jozefowiez, F. Semet, and E. G. Talbi, "Multi-objective vehicle routing problems," *Eur. J. Oper. Res.*, vol. 189, no. 2, pp. 293–309, 2008.

[101] J. R. Kasprzyk, P. M. Reed, B. R. Kirsch, and G. W. Characklis, "Managing population and drought risks using many-objective water portfolio planning under uncertainty," *Water Resour. Res.*, vol. 45, no. 12, Dec. 2009.

[102] L. Ke, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[103] J. D. Knowles and D. W. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation," in *Proc. 1999 IEEE Congr. Evol. Comput.*, 1999, pp. 98–105.

[104] W. Kong, T. Chai, S. Yang, and J. Ding, "A hybrid evolutionary multiobjective optimization strategy for the dynamic power supply problem in magnesia grain manufacturing," *Appl. Soft Comput.*, vol. 13, pp. 2960–2969, 2013.

[105] W. Koo, C. Goh, and K. C. Tan, "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment," *Memetic Computing*, vol. 2, no. 2, pp. 87–110, 2010.

[106] N. Kowatari, A. Oyama, H. E. Aguirre, and K. Tanaka, "A study on large population MOEA using adaptive ε-box dominance and neighborhood recombination for many-objective optimization," in *Learning and Intelligent Optimization*, LNCS, vol. 7219, 2012, pp. 86–100.

[107] S. Kundu, S. Biswas, S. Das, and P. N Suganthan, "Crowding-based local differential evolution with speciation-based memory archive for dynamic multimodal optimization," in *Proc. 15th Ann. Conf. Genetic Evol. Comput. (GECCO)*, 2013, pp.33–40.

[108] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Jan. 2002.

[109] M. Laumanns, E. Zitzler, and L. Thiele, "A unified model for multi-objective evolutionary algorithms with elitism," in *Proc. 2000 IEEE Congr. Evol. Comput. (CEC)*, 2000, pp. 46–53.

[110] M. S. Lechuga, *Multi-Objective Optimisation Using Sharing in Swarm Optimisation Algorithms*, Ph.D. dissertation, University of Birmingham, 2009.

[111] J. Lepagnot and A. Nakib, "A new multiagent algorithm for dynamic continuous optimization," *Int. J. Appl. Metaheuristic Comput.*, vol. 1, no. 1, pp. 16–38, 2010.

[112] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2014.

[113] M. Li, S. Yang, and X. Liu, "Diversity comparison of Pareto front approximations in many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2568–2584, 2014.

[114] M. Li, S. Yang, and X. Liu, "A performance comparison indicator for Pareto front approximations in many-objective optimization," in *Proc. 2015 Ann. Conf. Genetic and Evolut. Comput. (GECCO'15)*, 2015, pp. 703–710.

[115] M. Li, S. Yang, and X. Liu, "A test problem for visual investigation of high-dimensional multi-objective search" in *Proc. 2014 IEEE Congress Evolut. Comput.*, 2014, pp. 2140–2147.

[116] M. Li, S. Yang, X. Liu, "Evolutionary algorithms with segment-based search for multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1295–1313, Aug. 2014.

[117] M. Li, S. Yang, X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[118] C. Li, S. Yang, and D. Pelta, "Benchmark generator for CEC'2012 competition on evolutionary computation for dynamic optimization problems," School of Computer Science, China University of Geosciences, Wuhan, China, Tech. Rep.,2011.

[119] C. Li, S. Yang, and M. Yang, "An adaptive multi-swarm optimizer for dynamic optimization problems," *Evolut. Comput.*, vol. 22, no. 4, pp. 559–594, 2014.

[120] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.

[121] K. Li, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[122] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, 2014.

[123] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in *Proc. 2009 IEEE Congr. Evol. Comput.*, 2009, pp. 361–384.

[124] R. Liu, Y. Chen, W. Ma, C. Mu, and L. Jiao, "A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model," *Soft Computing*, vol. 18, no. 10, pp. 1913–1929, 2013.

[125] R. Liu, J. Fan, and L. Jiao, "Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm," *Appl. Intell.*, vol. 43, no. 1, pp. 192–207, 2015.

[126] H. Liu, F. Gu, and Y. Cheung, "T-MOEA/D: MOEA/D wh objective transform in multiobjective problems," in *Proc. Int. Conf. Inform. Sci. and Manage. Eng.*, Xi'an, China, 2010, pp. 282–285.

[127] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput*., vol. 18, no. 3, pp. 450–455, 2014.

[128] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 325–343, 2003.

[129] C. A. Mattson, A. A. Mullur, and A. Messac, "Smart Pareto filter: Obtaining a minimal representation of multiobjective design space," *Eng. Optim.* vol. 36, no. 6, pp. 721–740, 2004.

[130] J. Mehnen, G. Rudolph, and T. Wagner, "Evolutionary optimization of dynamic multiobjective functions," in *Proc. 2nd Italian Workshop on Evol. Comput.*, 2006.

[131] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. Evolutionary Multi-Criterion Optimization*, vol. 1993, LNCS, 2001, pp. 82–95.

[132] A. Muruganantham, K. C. Tan, and F. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, 2016, in press.

[133] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 6, pp. 1–24, Oct. 2012.

[134] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 193–208, 2014.

[135] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multiobjective optimization," in *Parallel Problem Solving from Nature (PPSN VIII)*, ser. LNCS, vol. 3242, 2004, pp. 792–802.

[136] Z. Peng, J. Zheng, J. Zou, and M. Liu, "Novel prediction and memory strategies for dynamic multiobjective optimization," *Soft Computing*, vol. 19, no. 9, pp. 2633–2653, 2014.

[137] C. Poloni, "Hybrid GA for multiobjective aerodynamic shape optimization," in *Genetic Algorhms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York, NY, USA: Wiley, 1997, pp. 397–414.

[138] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Ser. Natural Computing, Secaucus, NJ, USA: Springer, 2005.

[139] T. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[140] H. Richter, "Detecting change in dynamic fitness landscapes," in *Proc. 2014 IEEE Congr. Evol. Comput. (CEC)*, 2009, pp. 1613–1620.

[141] C. A. Rodriguez Villalobos and C. A. Coello Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proc. 14th Ann. Conf. Gentic Evolut. Comput. (GECCO)*, 2012, pp. 505–512.

[142] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, "Evolutionary path control strategy for solving many-objective optimization problem," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 702–715, 2015.

[143] G. Rudolph, O. Schütze, C. Grimme, and H. Trautmann, "A multiobjective evolutionary algorithm guided by averaged Hausdorff distance to aspiration sets," in *EVOLVE–A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, 2014, pp. 261–273.

[144] R. Saborido, A. B. Ruiz, and M. Luque, "Gobal WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, in press, 2016.

[145] S. Salomon, G. Avigad, P. J. Fleming, and R. C. Purshouse, "Active robust optimization: Enhancing robustness to uncertain environments," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2221–2231, 2014.

[146] S. Salomon, R. C. Purshouse, G. Avigad, and P. J. Fleming, "An evolutionary approach to active robust multiobjective optimisation," in *Evolutionary Multi-Criterion Optimization*, vol. 9019, Lecture Notes in Computer Science, 2015, pp. 141–155.

[147] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs," *J. Heuristics*, vol. 21, no. 6, pp. 819–849, 2015.

[148] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, no. 17, no. 1, pp. 77–99, Feb. 2013.

[149] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. on Genetic Algorithms*, 1985, pp. 93–100.

[150] J. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, M.S. thesis, Massachusetts of Technology, 1995.

[151] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, no. 16, no. 4, pp. 504–522, 2012.

[152] R. Shang, L. Jiao, Y. Ren, L. Li, and L. Wang, "Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization," *Soft Computing*, vol. 18, no. 4, pp.743–756, 2014.

[153] J. K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.

[154] H. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, 2011.

[155] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 2009.

[156] R. Takahashi, E. Carrano, and E. Wanner, "On a stochastic differential equation approach for multiobjective optimization up to Pareto-criticality," in *Proc. 6th Int. Conf. Evol. Multi-criterion Optimiz. (EMO 2011)*, LNCS, vol. 6576, 2011, pp. 61–75.

[157] T. Tanino, M. Tanaka, and C. Hojo, "An interactive multicriteria decision making method by using a genetic algorithm," in *Proc. 2nd Int. Conf. Systems Science and Systems Engineering*, 1993, pp. 381–386.

[158] T. Tanino, T. Tanaka, and M. Inuiguchi, *Multi-objective Programming and Goal Programming: Theory and Applications*, Lecture Notes in Economics and Mathematical Systems, vol. 618, Berlin: Springer-Verlag, 2009.

[159] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D+uniform design: A new version of MOEA/D for optimization problems wh many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.

[160] E. Tantar, A. Tantar, and P. Bouvry, "On dynamic multi-objective optimization classification and performance measures," in *Proc. 2011 IEEE Congr. Evol. Comput. (CEC 2011)*, 2011, pp. 2759–2766.

[161] R. Tinós and S. Yang, "A self-organizing random immigrants genetic algorithm for dynamic optimization problems," *Genetic Programming and Evolvable Machines*, vol. 8, no. 3, pp. 255–286, 2007.

[162] S. Tiwari, G. Fadel, and K. Deb, "AMGA2: Improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization," *Eng. Optim.*, vol. 43, no. 4, pp. 377–401, 2011.

[163] A. Trivedi, D. Srinivasan, S. Biswas, and T. Reindl, "Hybridizing genetic algorithm with different evolution for solving the unit commitment scheduling problem," *Swarm Evol. Comput.*, vol. 23, pp.50-64, 2015.

[164] L. Y. Tseng and C. Chen, "Multiple trajectory search for unconstrained/constrained multi-objective optimization," in *Proc. 2014 IEEE Congr. Evol. Comput. (CEC)*, 2009, pp. 1951–1958.

[165] R. K. Ursem, T. Krink, and B. Filipic, "A numerical simulator of a crop-producing greenhouse," EVALife, Department of Computer Science, University of Aarhus, Denmark, Tech. Rep. EVALife 2002-01, 2002.

[166] F. Vavak and T. C. Fogarty, "Comparison of steady-state and generational genetic algorithms for use in nonstationary environments," in *Proc. 1996 IEEE Int. Conf. Evol. Comput.*, 1996, pp. 192–195.

[167] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-criterion Optimiz.*, 2007, pp. 742–756.

[168] J. Wahle, O. Annen, Ch. Schuster, L. Neubert, and M. Schreckenberg, "A dynamic rout guidance system based on real traffic data," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 302–308, 2001.

[169] H. Wang, L. Jiao, and X. Yao,"Two arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, 2015.

[170] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybern.*, 2016, in press.

[171] Y. Wang and B. Li, "Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization," *Memetic Computing*, vol. 2, no. pp. 3–24, 2010.

[172] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[173] H. Wang, D. Wang, and S. Yang, "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems," *Soft Computing*, vol. 13, no. 8-9, pp. 763–780, 2009.

[174] R. Wang, Q. Zhang, T. Zhang, "Decomposition based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, 2016, in Press.

[175] L. Wang, Q. Zhang, A. Zhou, M. Gong, and, L. Jiao, "Constrained subproblems in decomposition based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, vol. 3, pp. 475–480, 2015.

[176] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and, L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 474–486, 2016.

[177] K. Weicker, "Performance measures for dynamic environments," in *Parallel Problem Solving from Nature (PPSN VII)*, LNCS, vol. 2439, 2002, pp. 64–73.

[178] R. L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, 2012.

[179] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[180] L. Wismans, E. van Berkum, and M. Bliemer, "Handling multiple objectives in optimization of externalities as objectives for dynamic traffic management," *Eur. J. Transport and Infrastructure Res.*, vol. 14, no. 2, pp. 159–177, 2014.

[181] P. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Trans. Syst., Man, and Cybern., Part B: Cybern.*, vol. 41, no. 3, pp. 621–634, 2011.

[182] Y. Wu, Y. Jin, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Computing*, vol. 19, no. 11, pp. 3221–3235, 2014.

[183] S. Yang and X. Yao, *Evolutionary Computation for Dynamic Optimization Problems*, ser. Studies in Computational Intelligence, vol. 490, Berlin: Springer, 2013.

[184] S. Yang, S. Jiang, and Y. Jiang, "Improving the multiobjective evolutionary algorithm based on decomposition," *Soft Computing*, in press, 2016.

[185] Y. Yuan, H. Xu, B. Wang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, 2015.

[186] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, 2015.

[187] Z. Zhang, "Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control," *Applied Soft Computing*, vol. 8, no. 2, pp. 959–971, 2008.

[188] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[189] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D wh NBI-style Tchebycheff approach for portfolio management," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, 2010, pp. 1–8.

[190] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," School of CS and EE, University of Essex, Colchester, UK, Work. Rep. CES-491, 2009.

[191] Z. Zhang and S. Qian, "Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems," *Soft Computing*, vol. 15, no. 7, pp. 1333–1349, 2011.

[192] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, 2008.

[193] Q. Zhang, A. Zhou, S. Z. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," University of Essex, and Nanyang technological University, Tech. Rep. CES-887, 2008.

[194] S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, 2012.

[195] B. Zheng, "A new dynamic multi-objective optimization evolutionary algorithm," in *Proc. Third Int. Conf. Natural Comput.*, vol. 5, 2007, pp. 565–570.

[196] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, 2014.

[197] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim.*, 2007, pp. 832–846.

[198] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, 2016.

[199] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of Pareto-optimality for many-objective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 299–315, 2015.

[200] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[201] E. Zitzler, S. Kunzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Problem Solving from Nature (PPSN VIII)*, LNCS, vol. 3242, 2004, pp. 832–842.

[202] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. the EUROGEN Evol. Methods for Design, Optim. and Control with Appl. to Industry Problems*, 2002, pp. 95–100.

[203] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[204] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

Appendix A

MOP Test Problems

Table A.1 presents the formulation of the MOP test suite, including two new tri-objective problems (i.e., MOP8 and MOP9). The number of decision variables is $n = 10$, and the search space of these problems are $[0,1]^n$.

Table A.1 MOP Test Suite

Instance	Description	Remarks
MOP1	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$ PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	Two extremal attractors; Convex PF.
MOP2	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-x_1^2)$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$ PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-f_1^2, f_1 \in [0,1]\}$	Two extremal attractors; Concave PF.
MOP3	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)$ $f_2(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$ PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = \sqrt{1-f_1^2}, f_1 \in [0,1]\}$	Two extremal attractors; Concave PF.

Table A.1: (Continued)

MOP4	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1}\cos^2(2\pi x_1))$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$ PS: $\{x \in R^{(n-1)} \mid x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \mid f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	Two extremal attractors; Disconnected PF.
MOP5	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\cos(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$ PS: $\{x \in R^{(n-1)} \mid x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \mid f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	One intermediate attractor; Convex PF.
MOP6	$f_1(x) = (1+g(x))x_1x_2$ $f_2(x) = (1+g(x))x_1(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$ PS: $\{x \in R^{(n-2)} \mid x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \mid f_1+f_2+f_3 = 1, f_i \in [0,1], i = 1,2,3\}$	Many boundary attractors; Linear PF.
MOP7	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$ PS: $\{x \in R^{(n-2)} \mid x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \mid f_1^2+f_2^2+f_3^2 = 1, f_i \in [0,1], i = 1,2,3\}$	Many boundary attractors; Concave PF.
MOP8	$f_1(x) = (1+g(x))x_1x_2$ $f_2(x) = (1+g(x))x_1(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x)=5(\lvert\cos(\pi x_1)\rvert+\lvert\cos(\pi x_2)\rvert)\sum_{i=3}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$ PS: $\{x \in R^{(n-2)} \mid x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \mid f_1+f_2+f_3 = 1, f_i \in [0,1], i = 1,2,3\}$	One intermediate attractor; Linear PF.
MOP9	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 2\sin(0.5\pi(x_1+x_2))\sum_{i=3}^{n}(\lvert t_i\rvert^{0.6} - 0.9t_i^2)$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$ PS: $\{x \in R^{(n-2)} \mid x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \mid f_1^2+f_2^2+f_3^2 = 1, f_i \in [0,1], i = 1,2,3\}$	Two extremal attractors Concave PF.

Appendix B

List of Publications

1. S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017. (IF: 5.908)

2. S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2017, in press. (IF: 5.908)

3. S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, accepted, 2017. (IF: 5.908)

4. S. Jiang and S. Yang, "Evolutionary dynamic multi-objective optimization: benchmarks and algorithm comparisons," *IEEE Transactions on Cybernetics*, 5. vol. 47, no. 1, pp. 198–211, 2016. (IF: 4.943)

5. S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts" *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016. (IF: 4.943)

6. S. Yang, S. Jiang, and Y. Jiang, "Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes," *Soft Computing*, 2015, in press. (IF: 1.63)

7. W. Qian, H. Hou and S. Jiang, "New self-adaptive cuckoo search algorithm," *Computer Science (China)*, vol. 41, no. 7, pp. 279–282, Jun. 2014.

8. S. Jiang and S. Yang, "Convergence versus diversity in multiobjective optimization," in *14th International Conference on Parallel Problem Solving from Nature (PPSN XIV)*, 2016, in press.

9. S. Jiang and S. Yang, “Adaptive penalty scheme for multiobjective evolutionary algorithm based on decomposition,” in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, 2016.

10. S. Jiang, and S. Yang, “On the use of Hypervolume for diversity measurement of Pareto front approximations,” in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence in Multi-Criteria Decision Making (CIMCDM)*, 2016.

11. S. Jiang and S. Yang, “A fast strength Pareto evolutionary algorithm incorporating predefined preference information,” in *Proceedings of the 15th UK Workshop on Computational Intelligence (UKCI)*, 2015.

12. S. Jiang and S. Yang, “Approximating multiobjective optimization problems with complex Pareto fronts,” in *Proceedings of the 15th UK Workshop on Computational Intelligence (UKCI)*, 2015.

13. S. Jiang and S. Yang, “A framework of scalable dynamic test problems for dynamic multi-objective optimization,” in *Proceedings of the 2014 IEEE Symposium Series on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, 2014, pp. 32-39

14. S. Jiang and S. Yang, “A benchmark generator for dynamic multi-objective optimization problems,” in *Proceedings of the 14th UK Workshop on Computational Intelligence (UKCI)*, 2014, pp. 1-8.

15. S. Jiang and S. Yang, “An improved quantum-behaved particle swarm optimization based on linear interpolation,” in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 769-775.